Chapter 1
Model-based reinforcement learning for approximate optimal regulation

Rushikesh Kamalapurkar, Patrick Walters, and Warren E. Dixon

Abstract Reinforcement learning (RL)-based online approximate optimal control methods applied to deterministic systems typically require a restrictive persistence of excitation (PE) condition for convergence. This chapter develops a model-based RL algorithm to solve approximate optimal regulation problems online under a PE-like rank condition. The development is based on the observation that, given a model of the system, model-based RL can be implemented by evaluating the Bellman error at any number of desired points in the state space. Uniformly ultimately bounded regulation of the system states to a neighborhood of the origin, and convergence of the developed policy to a neighborhood of the optimal policy are established using a Lyapunov-based analysis, and simulations are presented to demonstrate the performance of the developed controller.

Key words: model-based reinforcement learning; concurrent learning; simulated experience; data-based control; adaptive control; system identification

1.1 Introduction

Reinforcement learning (RL) enables a cognitive agent to learn desirable behavior from interactions with its environment. In control theory, the desirable behavior is typically quantified using a cost function, and the control problem is formulated as the desire to find the optimal policy that minimizes a cumulative cost. In recent years, various RL-based techniques have been developed to approximately solve optimal control problems for continuous-time and discrete-time deterministic systems [1–34]. The approximate solution is

Rushikesh Kamalapurkar, Patrick Walters, and Warren Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, USA. Email: {rkamalapurkar, walters8, wdixon}@ufl.edu.

facilitated via value function approximation, where the optimal policy is computed based on an estimate of the value function.

Methods that seek online solutions to optimal control problems are comparable to adaptive control (cf., [3,8,12,14,35,36] and the references therein). In adaptive control, the estimates for the uncertain parameters in the plant model are updated using the current tracking error as a performance metric; whereas, in online RL-based techniques, estimates for the uncertain parameters in the value function are updated using the Bellman error (BE) as a performance metric. Typically, to establish regulation or tracking, adaptive control methods do not require the adaptive estimates to convergence to the true values. However, convergence of the RL-based controller to a neighborhood of the optimal controller requires convergence of the parameter estimates to a neighborhood of their ideal values.

Parameter convergence has been a focus of research in adaptive control for several decades. It is common knowledge that least squares and gradient descent-based update laws generally require persistence of excitation (PE) in the system state for convergence of the parameter estimates. Modification schemes such as projection algorithms, $\sigma-$modification, and $e-$modification are used to guarantee boundedness of parameter estimates and overall system stability; however, these modifications do not guarantee parameter convergence unless the PE condition, which is generally impossible to verify online, is satisfied [37–40].

In RL-based approximate online optimal control, the Hamilton-Jacobi-Bellman (HJB) equation along with an estimate of the state derivative (cf. [7,12]), or an integral form of the HJB equation (cf. [41]) is utilized to approximately evaluate the BE at each visited state along the system trajectory. The BE provides an indirect measure of the quality of the current estimate of the value function at each visited state along the system trajectory. Hence, the unknown value function parameters are updated based on evaluation of the BE along the system trajectory. Such weight update strategies create two challenges for analyzing convergence. The system states need to satisfy PE, and the system trajectory needs to visit enough points in the state space to generate a good approximation to the value function over the entire operating domain: i.e., exploration versus exploitation. These challenges are typically addressed in related literature (cf. [5, 8, 12, 19, 25–27, 42–44]) by adding an exploration signal to the control input to ensure sufficient exploration in the desired region of the state space. However, no analytical methods exist to compute the appropriate exploration signal when the system dynamics are nonlinear.

The aforementioned challenges arise from the restriction that the BE can only be evaluated along the system trajectories. In particular, the integral BE is only meaningful as a measure of quality of the value function if evaluated along the system trajectories, and state derivative estimators can only generate estimates of the state derivative along the system trajectories using numerical smoothing. Recently, [26] demonstrated that experience replay can

be used to improve data efficiency in online approximate optimal control by reuse of recorded data. However, since the data needs to be recorded along the system trajectory, the system trajectory under the designed approximate optimal controller needs to provide enough excitation for learning. In general, such excitation is not available; hence, the simulation results in [26] are generated using an added probing signal.

In this chapter, a different approach is used to improve data efficiency by observing that if the system dynamics are known, the state derivative, and hence, the BE can be evaluated at any desired point in the state space. Unknown parameters in the value function can therefore be adjusted based on least square minimization of the BE evaluated at any number of desired points in the state space. For example, in an infinite horizon regulation problem, the BE can be computed at sampled points uniformly distributed in a neighborhood around the origin of the state space. The results of this chapter indicate that convergence of the unknown parameters in the value function is guaranteed provided the selected points satisfy a rank condition. Since the BE can be evaluated at any desired point in the state space, sufficient exploration can be achieved by appropriately selecting the points in a desired neighborhood.

If each new evaluation of the BE along the system trajectory is interpreted as gaining experience via exploration, the use of a model to evaluate the BE at an unexplored point in the state space can be interpreted as a simulation of experience. Learning based on simulation of experience has been investigated in results such as [45–50] for stochastic model-based RL; however, these results solve the optimal control problem off-line in the sense that repeated learning trials need to be performed before the algorithm learns the controller, and system stability during the learning phase is not analyzed.

In this chapter a novel implementation of simulation of experience is presented for deterministic nonlinear systems using BE extrapolation. A detailed stability analysis is presented to establish online approximate learning of the optimal controller while maintaining system stability during the learning phase. The stability analysis shows that provided an exact model of the system is available, simulation of experience based on the model implemented via BE extrapolation can be utilized to approximately solve an infinite horizon optimal regulation problem online.

Exact model knowledge is assumed in this chapter for ease of exposition. Using techniques similar to results such as [15, 20, 22, 51], the developed method can be easily extended to establish set-point regulation and convergence to optimality when a system identifier is employed instead of an exact model of the system. Using techniques similar to results such as [22, 30], the developed method can also be extended to optimally track of a class of desired trajectories.

Simulation results are provided that demonstrate the efficacy of the system identification-based extension of the developed method for uncertain inherently unstable control affine nonlinear systems without the addition of a

probing signal. The performance of the developed controller is demonstrated through approximate solution of an optimal control problem that has a known analytical solution. Based on the known solution, an exact polynomial basis is used for value function approximation. Another simulation demonstrates the applicability of the developed technique in the case where the analytical solution, and hence, the basis for value function approximation is unknown. In this case, since the optimal solution is unknown, the optimal trajectories obtained using the developed technique are compared with optimal trajectories obtained through numerical optimal control techniques.

The performance of the developed controller is demonstrated via experiments conducted on an autonomous underwater vehicle (AUV) at a spring head. The developed approximate optimal controller is used to regulate three degrees-of-freedom of the AUV, i.e., surge, sway, and yaw, to a set-point. All the computations required to implement the controller are performed onboard using an embedded processor. The experimental results demonstrate the capability of the developed method to concurrently identify, optimize, and control a real-world nonlinear autonomous system.

1.2 Problem Formulation

Consider a control affine nonlinear dynamical system of the form

$$\dot{x}(t) = f(x(t)) + g(x(t))\,u(t), \tag{1.1}$$

$t \in \mathbb{R}_{\geq t_0}$,[1] where t_0 denotes the initial time, $x : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ denotes the system state $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ denote the drift dynamics and the control effectiveness, respectively, and $u : \mathbb{R}_{\geq t_0} \to \mathbb{R}^m$ denotes the control input. The functions f and g are assumed to be locally Lipschitz continuous. Furthermore, $f(\mathbf{0}_{n\times 1}) = \mathbf{0}_{n\times 1}$ and $\nabla f : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ are continuous.[2][3] In the following, the notation $\phi^u(t; t_0, x^o)$ denotes the trajectory of the system in (1.1) under the control signal u with the initial condition[4] $x^o \in \mathbb{R}^n$ and initial time $t_0 \in \mathbb{R}_{\geq 0}$.

The control objective is to solve the infinite-horizon optimal regulation problem online, i.e., to design a control signal u online to minimize the cost functional

[1] The notation $\mathbb{R}_{\geq a}$ denotes the interval $[a, \infty)$, and the notation $\mathbb{R}_{>a}$ denotes the interval (a, ∞).

[2] The notation $\nabla f(x, y, \cdots)$ denotes the partial derivative of f with respect to the first argument.

[3] The notations $\mathbf{0}_{n\times m}$ and I_n denote an $n \times m$ matrix of zeros and an $n \times n$ identity matrix, respectively.

[4] The notation $(\cdot)^o$ is used to denote an arbitrary variable.

$$J(x,u) \triangleq \int_{t_0}^{\infty} r\left(x(\tau), u(\tau)\right) d\tau, \tag{1.2}$$

under the dynamic constraint in (1.1) while regulating the system state to the origin. In (1.2), $r : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ denotes the instantaneous cost defined as

$$r\left(x^o, u^o\right) \triangleq Q\left(x^o\right) + u^{oT} R u^o, \tag{1.3}$$

for all $x^o \in \mathbb{R}^n$ and $u^o \in \mathbb{R}^m$, where $Q : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a positive definite function, and $R \in \mathbb{R}^{m \times m}$ is a constant positive definite symmetric matrix.

Assuming an optimal controller exists, a closed-form solution to the optimal control problem is formulated in terms of the optimal value function $V^* : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ defined as

$$V^*\left(x^o\right) \triangleq \min_{u(\tau) \in U | \tau \in \mathbb{R}_{\geq t}} \int_{t}^{\infty} r\left(\phi^u\left(\tau; t, x^o\right), u(\tau)\right) d\tau. \tag{1.4}$$

Assuming that the optimal value function is continuously differentiable, it can be obtained by solving the corresponding HJB equation [52]

$$\nabla V^*\left(x^o\right)\left(f\left(x^o\right) + g\left(x^o\right) u^*\left(x^o\right)\right) + Q\left(x^o\right) + u^{*T}\left(x^o\right) R u^*\left(x^o\right) = 0, \tag{1.5}$$

for all $x^o \in \mathbb{R}^n$, with the boundary condition $V^*(0) = 0$. The optimal control law can be determined using the optimal value function as $u^*\left(x^o\right) = -\frac{1}{2} R^{-1} g^T\left(x^o\right)\left(\nabla V^*\left(x^o\right)\right)^T$ [52].

An analytical solution of the HJB equation is generally infeasible; hence, an approximate solution is sought. An approximate solution of the HJB equation is facilitated by replacing V^* and u^* in (1.5) by their respective subsequently defined parametric estimates $\hat{V}\left(x^o, \hat{W}_c^o\right)$ and $\hat{u}\left(x^o, \hat{W}_a^o\right)$ to compute the BE $\delta : \mathbb{R}^{n+2L} \to \mathbb{R}$ as

$$\delta\left(x^o, \hat{W}_c^o, \hat{W}_a^o\right) = \nabla \hat{V}\left(x^o, \hat{W}_c^o\right)\left(f\left(x^o\right) + g\left(x^o\right)\hat{u}\left(x^o, \hat{W}_a^o\right)\right) + x^{oT} Q x^o + \hat{u}^T\left(x^o, \hat{W}_a^o\right) R \hat{u}\left(x^o, \hat{W}_a^o\right), \tag{1.6}$$

where $\hat{W}_c^o \in \mathbb{R}^L$ and $\hat{W}_a^o \in \mathbb{R}^L$ denote the estimates of the unknown parameters in the approximation of the value function and the policy, respectively. The control objective is achieved by simultaneously adjusting the estimates $\hat{W}_c^o$ and $\hat{W}_a^o$ to minimize the BE.

Since the BE depends on the drift dynamics, f is assumed to be known. The focus of this chapter is a novel implementation of simulation of experience for online approximate optimal control of deterministic nonlinear systems. If a system model is available, then the approximate optimal control technique can be implemented using the model. However, if an exact model of the

system is unavailable, then parametric system identification can be employed to generate an estimate of the system model. A possible approach is to use parameters that are estimated offline in a separate experiment. A more useful approach is to use the offline estimate as an initial guess, and employ an adaptive system identification technique capable of refining the initial guess based on input-output data. The proposed technique can be easily extended to incorporate an online adaptive system identifier (cf. [15, 20, 22, 51]).

1.3 Approximate Optimal Control

1.3.1 Value function approximation

Approximations to the optimal value function V^* and the optimal policy u^* are designed based on neural network (NN)-based representations. Given any compact set $\chi \subset \mathbb{R}^n$ and positive constants $\bar{\epsilon}, \bar{\epsilon}' \in \mathbb{R}$, the universal approximation property of NNs can be exploited to represent the optimal value function V^* as $V^*\left(x^o\right) = W^T \sigma\left(x^o\right) + \epsilon\left(x^o\right)$, for all $x^o \in \chi$, where $W \in \mathbb{R}^L$ is the ideal weight matrix, which is bounded above by a known positive constant $\bar{W}$ in the sense that $\|W\| \leq \bar{W}$, $\sigma : \mathbb{R}^n \to \mathbb{R}^L$ is a continuously differentiable nonlinear activation function such that $\sigma(0) = 0$ and $\sigma'(0) = 0$, $L \in \mathbb{N}$ is the number of neurons, and $\epsilon : \mathbb{R}^n \to \mathbb{R}$ is the function reconstruction error such that $\sup_{x^o \in \chi} \left|\epsilon\left(x^o\right)\right| \leq \bar{\epsilon}$ and $\sup_{x^o \in \chi} \left|\nabla \epsilon\left(x^o\right)\right| \leq \bar{\epsilon}'$.

Based on the NN representation of the value function a NN-based representation of the optimal controller is derived as $u^*\left(x^o\right) = -\frac{1}{2} R^{-1} g^T\left(x^o\right)\left(\nabla \sigma^T\left(x^o\right) W + \nabla \epsilon^T\left(x^o\right)\right)$. The NN-based approximations $\hat{V} : \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}$ and $\hat{u} : \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}^m$ are defined as

$$\hat{V}\left(x^o, \hat{W}_c^o\right) \triangleq \hat{W}_c^{oT} \sigma\left(x^o\right),$$
$$\hat{u}\left(x^o, \hat{W}_a^o\right) \triangleq -\frac{1}{2} R^{-1} g^T\left(x^o\right) \nabla \sigma^T\left(x^o\right) \hat{W}_a^o, \tag{1.7}$$

where $\hat{W}_c^o \in \mathbb{R}^L$ and $\hat{W}_a^o \in \mathbb{R}^L$ denote the estimates of the ideal weights W. The use of two sets of weights to estimate the same set of ideal weights is motivated by the stability analysis and the fact that it enables a formulation of the BE that is linear in the value function weight estimates $\hat{W}_c^o$, enabling a least squares-based adaptive update law.

1.3.2 Simulation of experience via BE extrapolation

In traditional RL-based algorithms, the value function estimate and the policy estimate are updated based on observed data. The use of observed data to learn the value function naturally leads to a sufficient exploration condition, which demands sufficient richness in the observed data. In stochastic systems, this is achieved using a randomized stationary policy (cf. [7,53,54]), whereas in deterministic systems, a probing noise is added to the derived control law (cf. [8,12,55–57]). The technique developed in this result implements simulation of experience in a model-based RL scheme by using the system model to extrapolate the approximate BE to unexplored areas of the state space.

In the following, $\delta_t : \mathbb{R}_{\geq t_0} \to \mathbb{R}$ denotes the BE in (1.6) evaluated along the trajectories of (1.1), (1.8), and (1.10) as $\delta_t(t) \triangleq \delta\left(x(t), \hat{W}_c(t), \hat{W}_a(t)\right)$ and $\delta_{ti} : \mathbb{R}_{\geq t_0} \to \mathbb{R}$ denotes BE extrapolated along the trajectories of (1.8), (1.10), and a predefined set of trajectories $\{x_i : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n \mid i = 1, \cdots, N\}$ as $\delta_{ti} \triangleq \delta\left(x_i(t), \hat{W}_c(t), \hat{W}_a(t)\right)$. A least-squares update law for the value function weights is designed based on the subsequent stability analysis as

$$\dot{\hat{W}}_c = -k_{c1}\Gamma(t)\frac{\omega(t)}{\rho(t)}\delta_t(t) - \frac{k_{c2}}{N}\Gamma(t)\sum_{i=1}^{N}\frac{\omega_i(t)}{\rho_i(t)}\delta_{ti}(t), \tag{1.8}$$

$$\dot{\Gamma}(t) = \beta\Gamma(t) - k_{c1}\Gamma(t)\frac{\omega(t)\omega^T(t)}{\rho^2(t)}\Gamma(t) - \frac{k_{c2}}{N}\Gamma(t)\sum_{i=1}^{N}\frac{\omega_i(t)\omega_i^T(t)}{\rho_i^2(t)}\Gamma(t), \tag{1.9}$$

$\Gamma(t_0) = \Gamma_0$, where $\Gamma : \mathbb{R}_{\geq t_0} \to \mathbb{R}^{L\times L}$ is a time-varying least-squares gain matrix, $\omega(t) \triangleq \sigma'(x(t))\left(Y(x(t))\hat{\theta}(t) + g(x(t))\hat{u}\left(x(t), \hat{W}_a(t)\right)\right)$, $\omega_i(t) \triangleq \sigma'(x_i(t))\left(Y(x_i(t))\hat{\theta}(t) + g(x_i(t))\hat{u}\left(x_i(t), \hat{W}_a(t)\right)\right)$, $\rho(t) \triangleq 1 + \gamma_1\omega^T(t)\omega(t)$, $\rho_i(t) \triangleq 1 + \gamma_1\omega_i^T(t)\omega_i(t)$, where $\gamma_1 \in \mathbb{R}$ is a constant positive normalization gain, $\mathbf{1}_{\{.\}}$ denotes the indicator function, $\overline{\Gamma} > 0 \in \mathbb{R}$ is a saturation constant, $\beta > 0 \in \mathbb{R}$ is a constant forgetting factor, and $k_{c1}, k_{c2} > 0 \in \mathbb{R}$ are constant adaptation gains.

The policy weights are updated to follow the value function weights as[5]

$$\dot{\hat{W}}_a(t) = -k_{a1}\left(\hat{W}_a(t) - \hat{W}_c(t)\right) - k_{a2}\hat{W}_a(t) + \frac{k_{c1}G_\sigma^T(t)\hat{W}_a(t)\omega^T(t)}{4\rho(t)}\hat{W}_c(t)$$

[5] Using the fact that the ideal weights are bounded, a projection-based (cf. [58]) update law $\dot{\hat{W}}_a = proj\left\{-k_{a1}\left(\hat{W}_a - \hat{W}_c\right)\right\}$ can be utilized to update the policy weights. Since the policy weights are bounded a priori by the projection algorithm, a less complex stability analysis can be used to establish the result in Theorem 1.

$$+\sum_{i=1}^{N}\frac{k_{c2}G_{\sigma i}^{T}\left(t\right)\hat{W}_{a}\left(t\right)\omega_{i}^{T}\left(t\right)}{4N\rho_{i}\left(t\right)}\hat{W}_{c}\left(t\right), \quad (1.10)$$

where k_{a1}, $k_{a2} \in \mathbb{R}$ are positive constant adaptation gains, $G_{\sigma}\left(t\right) \triangleq \sigma'\left(x\left(t\right)\right)g\left(x\left(t\right)\right)R^{-1}g^{T}\left(x\left(t\right)\right)\sigma'^{T}\left(x\left(t\right)\right)$, $G_{\sigma i}\left(t\right) \triangleq \nabla\sigma_{i}\left(t\right)g_{i}\left(t\right)R^{-1}g_{i}^{T}\left(t\right)\nabla\sigma_{i}^{T}\left(t\right) \in \mathbb{R}^{L\times L}$, where $g_{i}\left(t\right) = g\left(x_{i}\left(t\right)\right)$ and $\nabla\sigma_{i}\left(t\right) = \nabla\sigma\left(x_{i}\left(t\right)\right)$. Using the weight estimates $\hat{W}_{a}$, the controller for the system in (1.1) is designed as

$$u\left(t\right) = \hat{u}\left(x\left(t\right), \hat{W}_{a}\left(t\right)\right). \quad (1.11)$$

The following rank condition facilitates the subsequent stability analysis.

Assumption 1.1. There exists a finite set of trajectories $\left\{x_{i} : \mathbb{R}_{\geq t_{0}} \to \mathbb{R}^{n} \mid i = 1, \cdots, N\right\}$ and a constant $T \in \mathbb{R}_{>0}$ such that

$$\underline{c}_{1}I_{L} \leq \int_{t}^{t+T}\left(\frac{\omega\left(\tau\right)\omega^{T}\left(\tau\right)}{\rho^{2}\left(\tau\right)}\right)d\tau, \ \forall t \in \mathbb{R}_{\geq t_{0}}, \quad (1.12)$$

$$\underline{c}_{2}I_{L} \leq \inf_{t\in\mathbb{R}_{\geq t_{0}}}\left(\frac{1}{N}\sum_{i=1}^{N}\frac{\omega_{i}\left(t\right)\omega_{i}^{T}\left(t\right)}{\rho_{i}^{2}\left(t\right)}\right), \quad (1.13)$$

$$\underline{c}_{3}I_{L} \leq \frac{1}{N}\int_{t}^{t+T}\left(\sum_{i=1}^{N}\frac{\omega_{i}\left(\tau\right)\omega_{i}^{T}\left(\tau\right)}{\rho_{i}^{2}\left(\tau\right)}\right)d\tau, \ \forall t \in \mathbb{R}_{\geq t_{0}}, \quad (1.14)$$

where, at least one of the nonnegative constants $\underline{c}_{1}, \underline{c}_{2}$, and $\underline{c}_{3}$ is strictly positive.

The rank conditions in (1.12) - (1.14) depend on the estimates $\hat{W}_{a}$; hence, in general, they are impossible to guarantee a priori. However, unlike traditional adaptive dynamic programming literature that assumes ω is PE, Assumption 1.1 only requires either the regressor ω or the regressor ω_{i} to be persistently exciting. The regressor ω is completely determined by the system state x, and the weights $\hat{W}_{a}$. Hence, excitation in ω vanishes as the system states and the weights converge. Hence, in general, it is unlikely that $\underline{c}_{1} > 0$. However, the regressor ω_{i} depends on x_{i}, which can be designed independent of the system state x. Hence, $\underline{c}_{3}$ can be made strictly positive if the signal x_{i} contains enough frequencies, and $\underline{c}_{2}$ can be made strictly positive by selecting a sufficient number of extrapolation functions, i.e., $N \gg L$.

The update law in (1.8) is fundamentally different from the CL adaptive update in results such as [59, 60], in the sense that the trajectories $\{x_{i}\}$ are selected a priori based on prior information about the desired behavior of the system. Given the system dynamics, or an estimate of the system dynamics, the approximate BE can be extrapolated to any desired point in the state space, whereas in adaptive control, the prediction error is used as a

metric which can only be evaluated at observed data points along the state trajectory.

1.4 Stability analysis

1.4.1 Boundedness of the least-squares gain under persistent excitation

Intuitively, the selection of time-varying trajectories for BE extrapolation results in virtual excitation. That is, instead of using input-output data from a persistently excited system, the dynamic model is used to simulate persistent excitation to facilitate parameter convergence. In the following upper and lower bounds on the eigenvalues of the least-squares learning gain matrix Γ are established. Bounds on the eigenvalues of Γ are traditionally established under PE. The following lemma extends the traditional PE-based result to incorporate the generalized excitation conditions in Assumption 1.1.

Lemma 1. *Provided Assumption 1.1 holds and* $\lambda_{\min}\left\{\Gamma_0^{-1}\right\} > 0$*, the update law in (1.9) ensures that the least squares gain matrix satisfies*

$$\underline{\Gamma} I_L \leq \Gamma(t) \leq \overline{\Gamma} I_L, \tag{1.15}$$

$\forall t \in \mathbb{R}_{\geq 0}$*, where* $\overline{\Gamma} = \frac{1}{\min\left\{k_{c1}\underline{c}_1 + k_{c2}\max\left\{\underline{c}_2 T, \underline{c}_3\right\}, \lambda_{\min}\left\{\Gamma_0^{-1}\right\}\right\} e^{-\beta T}}$ *and* $\underline{\Gamma} = \frac{1}{\lambda_{\max}\left\{\Gamma_0^{-1}\right\} + \frac{(k_{c1}+k_{c2})}{\beta\gamma_1}}$*. Furthermore,* $\overline{\Gamma} > 0$*.*

Proof. The proof closely follows the proof of [40, Corollary 4.3.2]. The update law in (1.9) implies that $\frac{d}{dt}\Gamma^{-1}(t) = -\beta\Gamma^{-1}(t) + k_{c1}\frac{\omega(t)\omega^T(t)}{\rho^2(t)} + \frac{k_{c2}}{N}\sum_{i=1}^{N}\frac{\omega_i(t)\omega_i^T(t)}{\rho_i^2(t)}$. Hence,

$$\Gamma^{-1}(t) = e^{-\beta t}\Gamma_0^{-1} + k_{c1}\int_0^t e^{-\beta(t-\tau)}\frac{\omega(\tau)\omega^T(\tau)}{\rho^2(\tau)}d\tau + \frac{k_{c2}}{N}\int_0^t e^{-\beta(t-\tau)}\sum_{i=1}^{N}\frac{\omega_i(\tau)\omega_i^T(\tau)}{\rho_i^2(\tau)}d\tau.$$

To facilitate the proof, let $t < T$. Then,

$$\Gamma^{-1}(t) \geq e^{-\beta t}\Gamma_0^{-1} \geq e^{-\beta T}\Gamma_0^{-1} \geq \lambda_{\min}\left\{\Gamma_0^{-1}\right\} e^{-\beta T} I_L.$$

If $t \geq T$, then since the integrands are positive, Γ^{-1} can be bounded as

$$\Gamma^{-1}(t) \geq k_{c1} \int_{t-T}^{t} e^{-\beta(t-\tau)} \frac{\omega(\tau)\omega^T(\tau)}{\rho^2(\tau)} d\tau + \frac{k_{c2}}{N} \int_{t-T}^{t} e^{-\beta(t-\tau)} \sum_{i=1}^{N} \frac{\omega_i(\tau)\omega_i^T(\tau)}{\rho_i^2(\tau)} d\tau.$$

Hence,

$$\Gamma^{-1}(t) \geq k_{c1} e^{-\beta T} \int_{t-T}^{t} \frac{\omega(\tau)\omega^T(\tau)}{\rho^2(\tau)} d\tau + \frac{k_{c2}}{N} e^{-\beta T} \int_{t-T}^{t} \sum_{i=1}^{N} \frac{\omega_i(\tau)\omega_i^T(\tau)}{\rho_i^2(\tau)} d\tau.$$

Using Assumption 1.1,

$$\frac{1}{N} \int_{t-T}^{t} \sum_{i=1}^{N} \frac{\omega_i(\tau)\omega_i^T(\tau)}{\rho_i^2(\tau)} d\tau \geq \max\{\underline{c}_2 T, \underline{c}_3\} I_L, \int_{t-T}^{t} \frac{\omega(\tau)\omega^T(\tau)}{\rho^2(\tau)} d\tau \geq \underline{c}_1 I_L.$$

Hence a lower bound for Γ^{-1} is obtained as,

$$\Gamma^{-1}(t) \geq \min\left\{k_{c1}\underline{c}_1 + k_{c2}\max\{\underline{c}_2 T, \underline{c}_3\}, \quad \lambda_{\min}\left\{\Gamma_0^{-1}\right\}\right\} e^{-\beta T} I_L. \quad (1.16)$$

Provided Assumption 1.1 holds, the lower bound in (1.16) is strictly positive. Furthermore, using the facts that $\frac{\omega(t)\omega^T(t)}{\rho^2(t)} \leq \frac{1}{\gamma_1}$ and $\frac{\omega_i(t)\omega_i^T(t)}{\rho_i^2(t)} \leq \frac{1}{\gamma_1}$ for all $t \in \mathbb{R}_{\geq t_0}$,

$$\begin{aligned}\Gamma^{-1}(t) &\leq \int_{0}^{t} e^{-\beta(t-\tau)} \left(k_{c1}\frac{1}{\gamma_1} + \frac{k_{c2}}{N}\sum_{i=1}^{N}\frac{1}{\gamma_1}\right) I_L d\tau + e^{-\beta t}\Gamma_0^{-1} \\ &\leq \left(\lambda_{\max}\left\{\Gamma_0^{-1}\right\} + \frac{(k_{c1}+k_{c2})}{\beta\gamma_1}\right) I_L.\end{aligned}$$

Since the inverse of the lower and upper bounds on Γ^{-1} are the upper and lower bounds on Γ, respectively, the proof is complete.

1.4.2 Regulation and convergence to optimality

For notational brevity, the dependence of all the functions on the system states and time is suppressed hereafter unless required for clarity of exposition. To facilitate the subsequent stability analysis, the approximate BE is expressed in terms of the weight estimation errors $\tilde{W}_c \triangleq W - \hat{W}_c$ and $\tilde{W}_a \triangleq W - \hat{W}_a$. Subtracting (1.5) from (1.6), an unmeasurable form of the instantaneous BE can be expressed as

$$\delta_t = -\omega^T \tilde{W}_c + \frac{1}{4}\tilde{W}_a^T G_\sigma \tilde{W}_a + \Delta \quad (1.17)$$

where $G \triangleq gR^{-1}g^T \in \mathbb{R}^{n\times n}$, $\Delta \triangleq \frac{1}{2}W^T\nabla\sigma G\nabla\epsilon^T + \frac{1}{4}G_\epsilon - \nabla\epsilon f \in \mathbb{R}$, $G_\epsilon \triangleq \nabla\epsilon G\nabla\epsilon^T \in \mathbb{R}$, and G_σ was introduced in (1.10). Similarly, the approximate BE evaluated along the selected trajectories $\{x_i \mid i = 1, \cdots, N\}$ can be expressed as

$$\delta_{ti} = -\omega_i^T\tilde{W}_c + \frac{1}{4}\tilde{W}_a^T G_{\sigma i}\tilde{W}_a + \Delta_i, \tag{1.18}$$

where $\nabla\epsilon_i = \nabla\epsilon(x_i)$, $f_i = f(x_i)$, $G_i \triangleq g_i R^{-1} g_i^T \in \mathbb{R}^{n\times n}$, $\Delta_i \triangleq \frac{1}{2}W^T\nabla\sigma_i G_i\nabla\epsilon_i^T + \frac{1}{4}G_{\epsilon i} - \nabla\epsilon_i f_i \in \mathbb{R}$ is a constant, $G_{\epsilon i} \triangleq \nabla\epsilon_i G_i\nabla\epsilon_i^T \in \mathbb{R}$, and $G_{\sigma i}$ was introduced in (1.10).

Let $B_\zeta \subset \mathbb{R}^{n+2L}$ denote a closed ball with radius ζ centered at the origin. Let $\chi \triangleq B_\zeta \cap \mathbb{R}^n$. Let the notation $\overline{\|(\cdot)\|}$ be defined as $\overline{\|h\|} \triangleq \sup_{x^o\in\chi}\|h(x^o)\|$, for some continuous function $h : \mathbb{R}^n \to \mathbb{R}^k$. To facilitate the analysis, let $\{\varpi_j \in \mathbb{R}_{>0} \mid j = 1, \cdots, 7\}$ be constants such that $\varpi_1 + \varpi_2 + \varpi_3 = 1$, and $\varpi_4 + \varpi_5 + \varpi_6 + \varpi_7 = 1$. Let $\underline{c} \in \mathbb{R}_{>0}$ be a constant defined as

$$\underline{c} \triangleq \frac{\beta}{2\overline{\Gamma}k_{c2}} + \frac{c_2}{2}, \tag{1.19}$$

and let $\iota \in \mathbb{R}$ be a positive constant defined as

$$\iota \triangleq \frac{(k_{c1}+k_{c2})^2\overline{\|\Delta\|}^2}{4k_{c2}\underline{c}\varpi_3} + \frac{1}{4(k_{a1}+k_{a2})\varpi_6}\left(\frac{\overline{W}\overline{\|G_\sigma\|}}{2} + \frac{(k_{c1}+k_{c2})\overline{W}^2\overline{\|G_\sigma\|}}{4} + \frac{\overline{\|\nabla\epsilon G^T\nabla\sigma^T\|}}{2} + k_{a2}\overline{W}\right)^2 + \frac{1}{4}\overline{\|G_\epsilon\|}. \tag{1.20}$$

To facilitate the stability analysis, let $V_L : \mathbb{R}^{n+2L}\times\mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a continuously differentiable candidate Lyapunov function defined as

$$V_L(Z^o, t) \triangleq V^*(x^o) + \frac{1}{2}\tilde{W}_c^{oT}\Gamma^{-1}\tilde{W}_c^o + \frac{1}{2}\tilde{W}_a^{oT}\tilde{W}_a^o, \tag{1.21}$$

where V^* is the optimal value function and $Z^o \triangleq \left[x^{oT},\ \tilde{W}_c^{oT},\ \tilde{W}_a^{oT}\right]^T$. Using the fact that V^* is positive definite, (1.15), Lemma 1, and Lemma 4.3 from [61] yield

$$\underline{v_l}(\|Z^o\|) \leq V_L(Z^o, t) \leq \overline{v_l}(\|Z^o\|), \tag{1.22}$$

for all $t \in \mathbb{R}_{\geq t_0}$ and for all $Z^o \in \mathbb{R}^{n+2L}$, where $\underline{v_l}, \overline{v_l} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ are class $\mathcal{K}$ functions. Let $v_l : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a class $\mathcal{K}$ function such that $v_l(\|Z^o\|) \geq \frac{Q(x^o)}{2} + \frac{k_{c2}\underline{c}\varpi_1}{2}\left\|\tilde{W}_c^o\right\|^2 + \frac{(k_{a1}+k_{a2})\varpi_4}{2}\left\|\tilde{W}_a^o\right\|^2$.

Let $Z : \mathbb{R}_{\geq t_0} \to \mathbb{R}^{n+2L}$ denote the concatenated trajectories of (1.1), (1.8), and (1.10), defined as $Z(t) \triangleq \left[x^T(t),\ \tilde{W}_c^T(t),\ \tilde{W}_a^T(t)\right]^T$. The sufficient

conditions for ultimate boundedness of Z are derived based on the subsequent stability analysis as

$$k_{c2}\underline{c}\left(k_{a1}+k_{a2}\right)\varpi_5\varpi_2 \geq \left(k_{a1}+\frac{1}{4}\left(k_{c1}+k_{c2}\right)\overline{W}\overline{\|G_\sigma\|}\right), \tag{1.23}$$

$$\left(k_{a1}+k_{a2}\right)\varpi_7 \geq \frac{1}{4}\left(k_{c1}+k_{c2}\right)\overline{W}\overline{\|G_\sigma\|}, \tag{1.24}$$

$$v_l^{-1}\left(\iota\right) < \overline{v_l}^{-1}\left(\underline{v_l}\left(\zeta\right)\right). \tag{1.25}$$

The bound on the function f and the NN function approximation errors depend on the underlying compact set; hence, ι is a function of ζ. Even though, in general, ι increases with increasing ζ, the sufficient condition in (1.25) can be satisfied provided the points for BE extrapolation are selected such that the constant $\underline{c}$, introduced in (1.19) is large enough and that the basis for value function approximation are selected such that $\overline{\|\epsilon\|}$ and $\overline{\|\nabla\epsilon\|}$ are small enough. The main result of this chapter can now be stated as follows.

Theorem 1. *Provided Assumption (1.1) holds and the sufficient gain conditions in (1.23) - (1.25) are satisfied, the controller in (1.11) along with the adaptive update laws in (1.8) - (1.10) ensure that the state x, the value function weight estimation error $\tilde{W}_c$ and the policy weight estimation error $\tilde{W}_a$ are uniformly ultimately bounded.*

Proof. The time derivative of (1.21) along the trajectories of (1.1) and (1.8) - (1.10) is given by

$$\begin{aligned}
\dot{V}_L &= \dot{V}^* - \tilde{W}_c^T\Gamma^{-1}\dot{\hat{W}}_c - \frac{1}{2}\tilde{W}_c^T\dot{\Gamma}^{-1}\tilde{W}_c - \tilde{W}_a^T\dot{\hat{W}}_a,\\
&= \nabla V^*\left(f+gu\right) - \tilde{W}_c^T\left(-k_{c1}\frac{\omega}{\rho}\delta_t - \frac{k_{c2}}{N}\sum_{i=1}^{N}\frac{\omega_i}{\rho_i}\delta_{ti}\right)\\
&- \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\left(\beta\Gamma - k_{c1}\left(\Gamma\frac{\omega\omega^T}{\rho^2}\Gamma\right) - \frac{k_{c2}}{N}\Gamma\sum_{i=1}^{N}\frac{\omega_i\omega_i^T}{\rho_i^2}\Gamma\right)\Gamma^{-1}\tilde{W}_c\\
&- \tilde{W}_a^T\left(-k_{a1}\left(\hat{W}_a-\hat{W}_c\right) - k_{a2}\hat{W}_a\right)\\
&- \tilde{W}_a^T\left(\frac{k_{c1}G_\sigma^T\hat{W}_a\omega^T}{4\rho} + \sum_{i=1}^{N}\frac{k_{c2}G_{\sigma i}^T\hat{W}_a\omega_i^T}{4N\rho_i}\right)\hat{W}_c.
\end{aligned}$$

Substituting for the approximate BEs from (1.17) and (1.18) and using the inequality $\frac{\omega\omega^T}{\rho^2} \leq \frac{\omega\omega^T}{\rho}$, the Lyapunov derivative can be bounded as

$$\dot{V}_L \leq -Q\left(x\right) - k_{c2}\tilde{W}_c^T\left(\frac{\beta\Gamma^{-1}}{2k_{c2}} + \sum_{i=1}^{N}\frac{\omega_i\omega_i^T}{2N\rho_i}\right)\tilde{W}_c - \left(k_{a1}+k_{a2}\right)\tilde{W}_a^T\tilde{W}_a$$

$$+\left(\frac{W^T G_\sigma + \nabla\epsilon G^T \nabla\sigma^T}{2} + k_{a2} W^T - \frac{k_{c1} W^T \omega W^T G_\sigma}{4\rho} - \sum_{i=1}^{N} \frac{k_{c2} W^T \omega_i W^T G_{\sigma i}}{4N\rho_i}\right)\tilde{W}_a$$
$$+ \tilde{W}_c^T \left(k_{a1} + \frac{k_{c1}\omega W^T G_\sigma}{4\rho} + \sum_{i=1}^{N} \frac{k_{c2}\omega_i W^T G_{\sigma i}}{4N\rho_i}\right)\tilde{W}_a$$
$$+ W^T \frac{k_{c1}\omega}{4\rho}\tilde{W}_a^T G_\sigma \tilde{W}_a + W^T \sum_{i=1}^{N} \frac{k_{c2}\omega_i}{4N\rho_i}\tilde{W}_a^T G_{\sigma i}\tilde{W}_a$$
$$+ \tilde{W}_c^T \left(k_{c1}\frac{\omega}{\rho}\Delta + \frac{1}{N}\sum_{i=1}^{N} \frac{k_{c2}\omega_i}{\rho_i}\Delta_i\right) + \frac{1}{4}G_\epsilon.$$

Provided the gains are selected based on the sufficient conditions in (1.23) - (1.25), the Lyapunov derivative can be upper-bounded as

$$\dot{V}_L \leq -v_l\left(\|Z\|\right), \quad \forall \|Z\| > v_l^{-1}\left(\iota\right), \tag{1.26}$$

for all $t \geq 0$ and $\forall Z \in B_\zeta$. Using (1.22), (1.25), and (1.26), Theorem 4.18 in [61] can now be invoked to conclude that Z is uniformly ultimately bounded in the sense that $\limsup_{t\to\infty} \|Z(t)\| \leq \underline{v_l}^{-1}\left(\overline{v_l}\left(v_l^{-1}(\iota)\right)\right)$. Furthermore, the concatenated state trajectories are bounded such that $\|Z(t)\| \in B_\zeta$ for all $t \in \mathbb{R}_{\geq t_0}$. Since the estimates $\hat{W}_a$ approximate the ideal weights W, the policy $\hat{u}$ approximates the optimal policy u^*.

1.5 Simulation

This section presents two simulations to demonstrate the performance and the applicability of the developed technique. First, the performance of the developed controller is demonstrated through approximate solution of an optimal control problem that has a known analytical solution. Based on the known solution, an exact polynomial basis is used for value function approximation. The second simulation demonstrates the applicability of the developed technique in the case where the analytical solution, and hence, the basis for value function approximation is unknown. In this case, since the optimal solution is unknown, the optimal trajectories obtained using the developed technique are compared with optimal trajectories obtained through offline numerical optimal control techniques.

1.5.1 Problem with a known basis

The performance of the developed controller is demonstrated by simulating a nonlinear, control affine system with a two dimensional state $x = [x_1, x_2]^T$. The system dynamics are described by (1.1), where [8]

$$f = \begin{bmatrix} x_1 & x_2 & 0 & 0 \\ 0 & 0 & x_1 & x_2 \left(1 - (cos(2x_1) + 2)^2\right) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix},$$
$$g = \begin{bmatrix} 0 \\ cos(2x_1) + 2 \end{bmatrix}. \tag{1.27}$$

In (1.27), a, b, c, $d \in \mathbb{R}$ are positive unknown parameters.[6] The parameters are selected as[7] $a = -1$, $b = 1$, $c = -0.5$, and $d = -0.5$. The control objective is to minimize the cost in (1.4), where $Q = I_2$ and $R = 1$. The optimal value function and optimal control for the system in (1.27) are given by $V^*(x) = \frac{1}{2}x_1^2 + x_2^2$, and $u^*(x) = -(cos(2x_1) + 2)x_2$ (cf. [8]).

To facilitate the ADP-based controller, the basis function $\sigma : \mathbb{R}^2 \to \mathbb{R}^3$ for value function approximation is selected as $\sigma = \left[x_1^2, x_1x_2, x_2^2\right]$. Based on the analytical solution, the ideal weights are $W = [0.5, 0, 1]^T$. The data points for the CL-based update law in (1.8) are selected to be on a 5×5 grid around the origin. The initial condition for the system state is selected as $x(0) = [-1, -1]^T$.

Figure 1.1 demonstrates that the system state is regulated to the origin, the unknown parameters in the drift dynamics are identified, and the value function and the policy weights converge to their true values. Furthermore, unlike previous results, a probing signal to ensure PE is not required. Figure 1.2 demonstrates the satisfaction of Assumptions 2 and 3.

1.5.2 Problem with an unknown basis

To demonstrate the applicability of the developed controller, a nonlinear, control affine system with a four dimensional state $x = [x_1, x_2, x_3, x_4]^T$ is simulated. The system dynamics are described by (1.1), where

[6] To relax the requirement of exact model knowledge, the simulations and the experiment employ a concurrent learning-based system identifier (cf. [15,20,22,51]). Using techniques similar to results such as [15,20,22,51], the analysis in Section (1.4) can be easily extended to establish set-point regulation and convergence to optimality when a system identifier is employed instead of an exact model of the system.

[7] The origin is an unstable equilibrium point of the unforced system $\dot{x} = f(x)$.

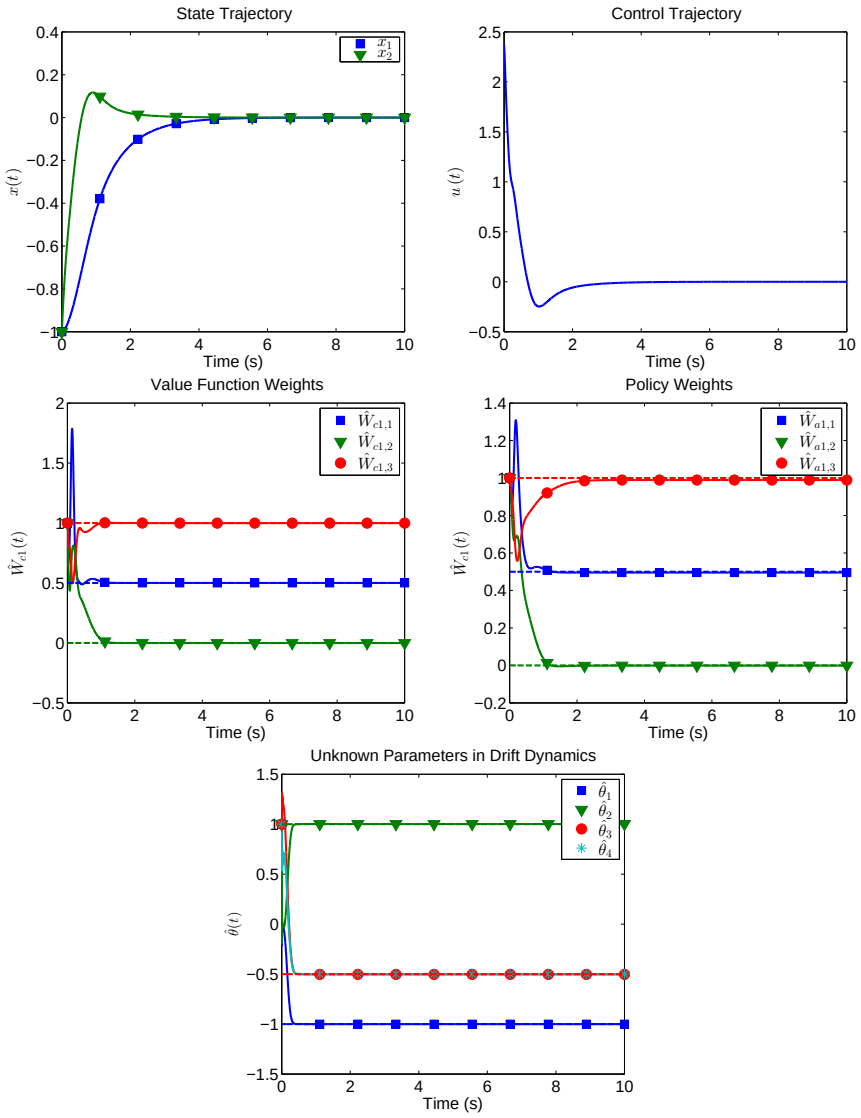

Fig. 1.1 System trajectories generated using the proposed method, and compared to the analytical solution.

$$f(x) = \begin{bmatrix} x_3 \\ x_4 \\ -M^{-1}V_m \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \end{bmatrix} + \begin{bmatrix} 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0 \\ \left[M^{-1}, \right. & M^{-1} & \left. \right] & D \end{bmatrix} \begin{bmatrix} f_{d1} \\ f_{d2} \\ f_{s1} \\ f_{s2} \end{bmatrix},$$
$$g(x) = \left[\left[0, 0\right]^T, \left[0, 0\right]^T, \left(M^{-1}\right)^T \right]^T. \tag{1.28}$$

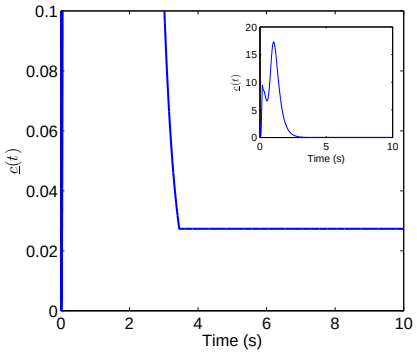

Fig. 1.2 Satisfaction of Assumption 1.1 for the simulation with known basis.

In (1.28), $x \triangleq [x_1,\, x_2,\, x_3,\, x_4]^T$, $D \triangleq \text{diag}\,[x_3,\, x_4,\, \tanh(x_3),\, \tanh(x_4)]$ and the matrices M, V_m, F_d, $F_s \in \mathbb{R}^{2\times 2}$ are defined as $M \triangleq \begin{bmatrix} p_1 + 2p_3c_2, & p_2 + p_3c_2 \\ p_2 + p_3c_2, & p_2 \end{bmatrix}$, $F_d \triangleq \begin{bmatrix} f_{d1}, & 0 \\ 0, & f_{d2} \end{bmatrix}$, $V_m \triangleq \begin{bmatrix} -p_3s_2x_4, & -p_3s_2(x_3 + x_4) \\ p_3s_2x_3, & 0 \end{bmatrix}$, and $F_s \triangleq \begin{bmatrix} f_{s1}\tanh(x_3), & 0 \\ 0, & f_{s2}\tanh(x_3) \end{bmatrix}$, where $c_2 = \cos(x_2)$, $s_2 = \sin(x_2)$, $p_1 = 3.473$, $p_2 = 0.196$, and $p_3 = 0.242$, and f_{d1}, f_{d2}, f_{s1}, $f_{s2} \in \mathbb{R}$ are positive unknown parameters.[8] The parameters are selected as $f_{d1} = 5.3$, $f_{d2} = 1.1$, $f_{s1} = 8.45$, and $f_{s2} = 2.35$. The control objective is to minimize the cost in (1.4), where $Q = \text{diag}([10,\, 10,\, 1,\, 1])$ and $R = \text{diag}([1,\, 1])$.

To facilitate the ADP-based controller, the basis function $\sigma : \mathbb{R}^4 \to \mathbb{R}^{10}$ for value function approximation is selected as $\sigma(x) = \left[x_1x_3,\, x_2x_4,\, x_3x_2,\, x_4x_1,\, x_1x_2,\, x_4x_3,\, x_1^2,\, x_2^2,\, x_3^2,\, x_4^2\right]$. The points for the CL-based update law in (1.8) are selected to be on a $3 \times 3 \times 3 \times 3$ grid around the origin, and the policy weights are updated using a projection-based update law. The initial condition for the system state is selected as $x(0) = [1,\, 1,\, 0,\, 0]^T$.

Figure 1.3 demonstrates that the system state is regulated to the origin, the unknown parameters in the drift dynamics are identified, and the value function and the policy weights converge. Figure 1.5 demonstrates the satisfaction of Assumptions 2 and 3. The value function and the policy weights converge to the following values.

$$\hat{W}_c^* = \hat{W}_a^* = [24.7,\, 1.19,\, 2.25,\, 2.67,\, 1.18,\, 0.93,\, 44.34,\, 11.31,\, 3.81\,,0.10]^T. \quad (1.29)$$

[8] See Footnote (6).

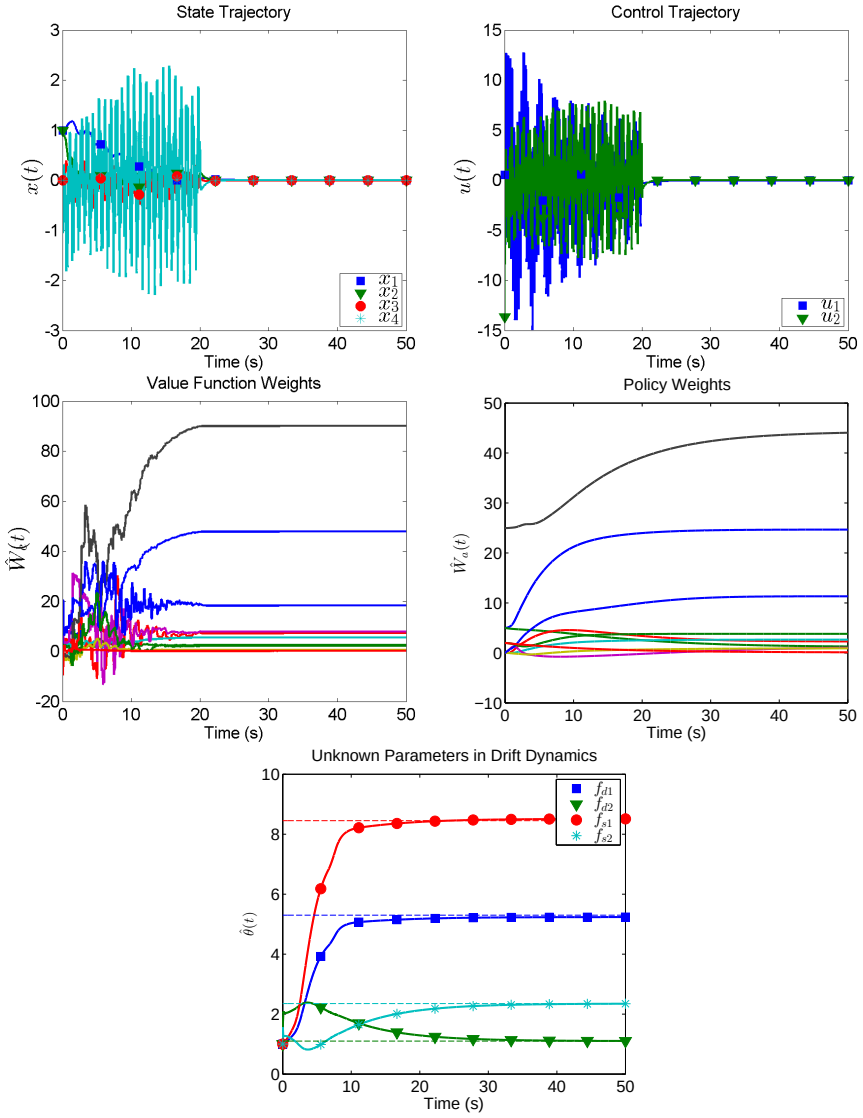

Fig. 1.3 System trajectories generated using the proposed method, where the drift parameter estimates are compared to the actual drift parameters.

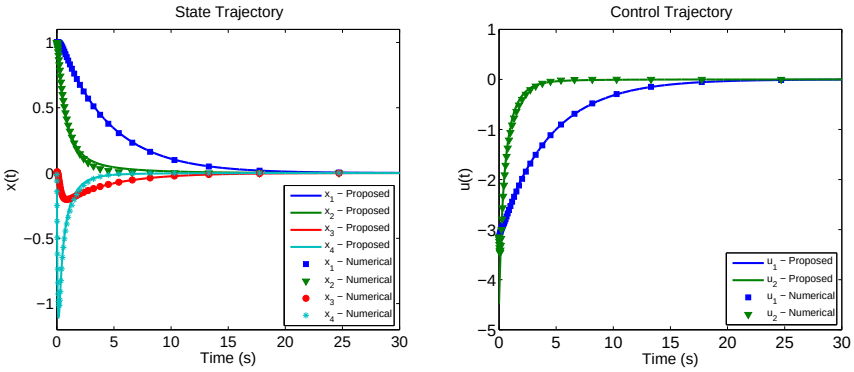

Fig. 1.4 State and control trajectories generated using feedback policy $\hat{u}^*(x)$ compared to a numerical optimal solution.

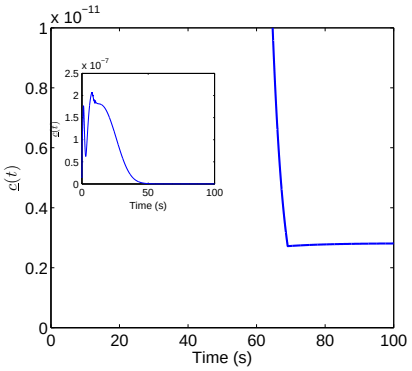

Fig. 1.5 Satisfaction of Assumption 1.1 for the simulation with unknown basis.

Since the true values of the value function weights are unknown, the weights in (1.29) can not be compared to their true values. However, a measure of proximity of the weights in (1.29) to the ideal weights W can be obtained by comparing the system trajectories resulting from applying the feedback control policy $\hat{u}^*(x) = -\frac{1}{2}R^{-1}g^T(x)\nabla\sigma^T(x)\hat{W}_a^*$ to the system, against numerically computed optimal system trajectories. In Figure 1.4, the numerical optimal solution is obtained using an infinite-horizon Gauss pseudospectral method (cf. [62]) using 45 collocation points. Figure 1.4 indicates that the weights in (1.29) generate state and control trajectories that closely match the numerically computed optimal trajectories.

1.6 Experimental Validation

The performance of the developed controller is demonstrated with experiments conducted at Ginnie Springs in High Springs, FL. Ginnie Springs is a second-magnitude spring discharging 142 million liters of freshwater daily with a spring pool measuring 27.4 m in diameter and 3.7 m deep [63]. Ginnie Springs was selected to validate the proposed controller because of its relatively high flow rate and clear waters for vehicle observation. For clarity of exposition[9] and to remain within the vehicle's depth limitations[10], the proposed method is implemented on 3 degrees-of-freedom of an AUV, i.e., surge, sway, yaw.

1.6.1 Experimental Platform

Experiments were conducted on an AUV, SubjuGator 7, developed at the University of Florida. The AUV, shown in Figure 1.6, is a small two man portable AUV with a mass of 40.8 kg. The vehicle is over-actuated with eight bidirectional thrusters.

Designed to be modular, the vehicle has multiple specialized pressure vessels that house computational capabilities, sensors, batteries, and mission specific payloads. The central pressure vessel houses the vehicle's motor controllers, network infrastructure, and core computing capability. The core computing capability services the vehicles environmental sensors (e.g. visible light cameras, scanning sonar, etc.), the vehicles high-level mission planning, and low-level command and control software. A standard small form factor computer makes up the computing capability and utilizes a 2.13 GHz server grade quad-core processor. Located near the front of the vehicle, the navigation vessel houses the vehicles basic navigation sensors. The suite of navigation sensors include an inertial measurement unit, a Doppler velocity log (DVL), a depth sensor, and a digital compass. The navigation vessel also includes an embedded 720 MHz processor for preprocessing and packaging navigation data. Along the sides of the central pressure vessel, two vessels house 44 Ah of batteries used for propulsion and electronics.

The vehicle's software runs within the Robot Operating System framework in the central pressure vessel. For the experiment, three main software nodes were used: navigation, control, and thruster mapping nodes. The navigation

[9] The number of basis functions and weights required to support a 6 DOF model greatly increases from the set required for the 3 DOF model. The increased number of parameters and complexity reduces the clarity of this proof of principal experiment.

[10] The vehicle's Doppler velocity log has a minimum height over bottom of approximately 3 m that is required to measure water velocity. A minimum depth of approximately 0.5 m is required to remove the vehicle from surface effects. With the depth of the spring nominally 3.7 m, a narrow window of about 20 cm is left operate the vehicle in heave.

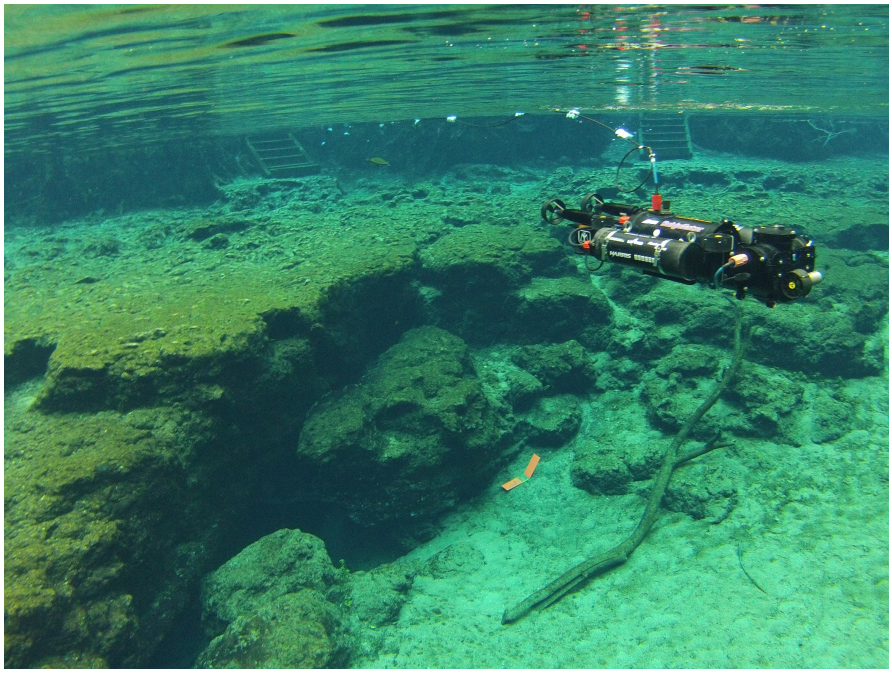

Fig. 1.6 SubjuGator 7 AUV operating at Ginnie Springs, FL.

node receives packaged navigation data from the navigation pressure vessel where an unscented Kalman filter estimates the vehicle's full state at 50Hz. The desired force and moment produced by the controller are mapped to the eight thrusters using a least-squares minimization algorithm. The controller node contains the proposed controller and system identifier.

1.6.2 Controller Implementation

Implementation of the developed controller is divided into three parts: system identification, value function iteration, and control iteration. Implementing the system identifier requires a recorded data set. The data set was collected in a swimming pool. The vehicle was commanded to track an exciting trajectory with a robust integral of the sign of the error (RISE) controller [64] while the state-action pairs were recorded. The recorded data was trimmed to a subset of 40 sampled points that were selected to maximize the minimum singular value of the history stack as in Section 6.2 of [65]. The system identifier is updated at 50 Hz.

Equations (1.6) and (1.8) form the value function iteration. Evaluating the extrapolated BE (1.6) with each control iteration is computational expensive.

Due to the limited computational resources available on-board the AUV, the update of the value function weights was calculated at 5 Hz.

For the experiments, the proposed controller was restricted to 3 degrees-of-freedom, i.e., surge, sway, and yaw. The RISE controller is used to regulate the remaining degrees-of-freedom, i.e., heave, roll, and pitch, in order to maintain the implied assumption that roll and pitch remain at zero and the depth remains constant. Implementing the proposed controller requires (1.10) and (1.11). The RISE controller in conjunction with the proposed controller runs at a rate of 50Hz.

The vehicle uses water profiling data from the DVL to measure the relative water velocity near the vehicle in addition to bottom tracking data for the state estimator. Between the state estimator, water profiling data, and recorded data, the equations used to implement the proposed controller only contain known or measurable quantities.

1.6.3 Results

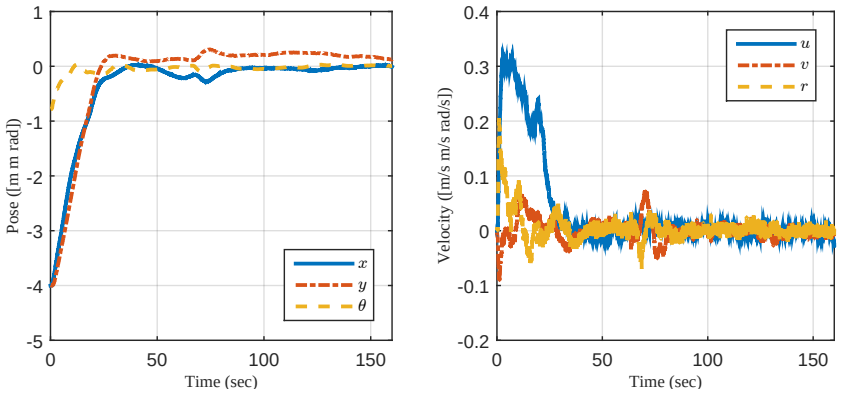

Fig. 1.7 Inertial position error $\eta \triangleq [x, y, \theta]^T$ (left) and body-fixed velocity error $\nu \triangleq [u, v, r]^T$ (right) of the AUV.

The vehicle was commanded to hold a station near the vent of Ginnie Spring. An initial condition of $x(t_0) = \left[4\,\text{m}\ 4\,\text{m}\ \frac{\pi}{4}\,\text{rad}\ 0\,\text{m/s}\ 0\,\text{m/s}\ 0\,\text{rad/s}\right]^T$ was given to demonstrate the controller's ability to regulate the state. The optimal control weighting matrices were selected to be $Q = \text{diag}([20, 50, 20, 10, 10, 10])$ and $R = I_3$. The NN weights were initialized to match the ideal values for the linearized optimal control problem, which is obtained by solving the algebraic Riccati equation with the dynamics lin-

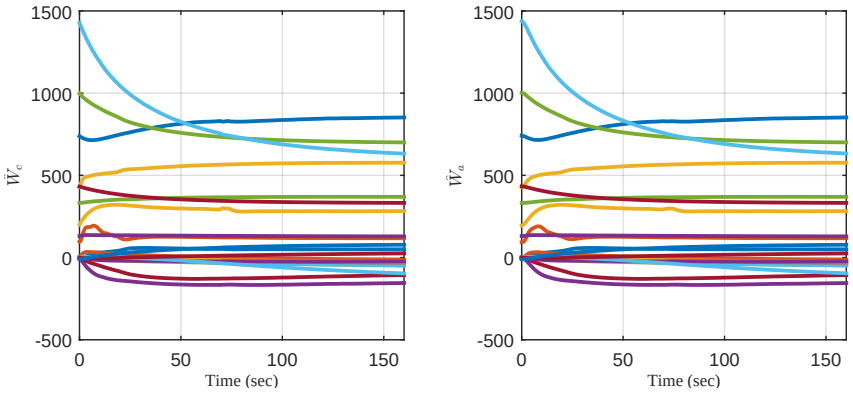

Fig. 1.8 Critic $\hat{W}_c$ (left) and actor $\hat{W}_a$ (right) neural network weight estimates online convergence.

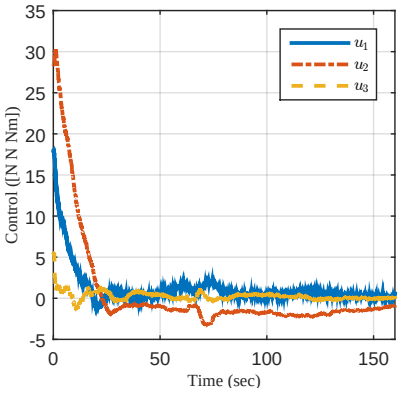

Fig. 1.9 Body-fixed optimal control effort commanded about the center of mass of the vehicle.

earized about the station. The BE was extrapolated to 2025 points in a grid about the station.

Figure 1.7 illustrates the ability of the generated policy to regulate the state. Figure 1.9 illustrates the total control effort applied to the body of the vehicle. Figure 1.9 illustrates the output of the approximate optimal policy for the residual system. Figure 1.10 illustrates the convergence of the parameters of the system identifier and Figure 1.8 illustrates convergence of the neural network weights representing the value function.

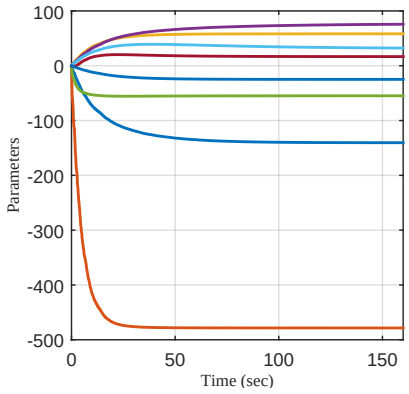

Fig. 1.10 Identified system parameters determined for the vehicle online. The parameter definitions may be found in Example 6.2 and Equation 6.100 of [66].

1.7 Conclusion

An online approximate optimal controller is developed, where the value function is approximated without PE in the system states via novel use of a model to to evaluate the BE over unexplored areas of the state-space. The PE condition along the system trajectories is replaced by an excitation condition that needs to be satisfied along virtual trajectories selected a priori. UUB regulation of the system states to a neighborhood of the origin, and convergence of the policy to a neighborhood of the optimal policy are established using a Lyapunov-based analysis. Simulations demonstrate that the developed technique generates an approximation to the optimal controller on-line, while maintaining system stability, without the use of a probing signal. Experiments demonstrate the ability to concurrently identify the uncertainties in the dynamics and generate an approximate optimal policy using the identified model. The vehicle follows the generated policy to achieve its station keeping objective in the presence of external disturbances using industry standard navigation and environmental sensors.

References

1. K. Doya, "Reinforcement learning in continuous time and space," *Neural Comput.* **12**(1), pp. 219–245, 2000.
2. R. Padhi, S. Balakrishnan, and T. Randolph, "Adaptive-critic based optimal neuro control synthesis for distributed parameter systems," *Automatica* **37**(8), pp. 1223–1234, 2001.
3. R. Padhi, N. Unnikrishnan, X. Wang, and S. Balakrishnan, "A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear

systems," *Neural Netw.* **19**(10), pp. 1648–1660, 2006.
4. A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **38**, pp. 943–949, 2008.
5. F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.* **9**(3), pp. 32–50, 2009.
6. T. Dierks, B. Thumati, and S. Jagannathan, "Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence," *Neural Netw.* **22**(5-6), pp. 851–860, 2009.
7. P. Mehta and S. Meyn, "Q-learning and pontryagin's minimum principle," in *Proc. IEEE Conf. Decis. Control*, pp. 3598 –3605, Dec. 2009.
8. K. Vamvoudakis and F. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica* **46**(5), pp. 878–888, 2010.
9. 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.* **22**(12), pp. 2226–2236, 2011.
10. F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*, Wiley, 3 ed., 2012.
11. D. Wang, D. Liu, and Q. Wei, "Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach," *Neurocomputing* **78**(1), pp. 14 – 22, 2012.
12. S. Bhasin, R. Kamalapurkar, M. Johnson, K. Vamvoudakis, F. L. Lewis, and W. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica* **49**(1), pp. 89–92, 2013.
13. 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.* **43**(1), pp. 206–216, 2013.
14. H. Zhang, D. Liu, Y. Luo, and D. Wang, *Adaptive Dynamic Programming for Control Algorithms and Stability*, Communications and Control Engineering, Springer-Verlag, London, 2013.
15. R. Kamalapurkar, P. Walters, and W. E. Dixon, "Concurrent learning-based approximate optimal regulation," in *Proc. IEEE Conf. Decis. Control*, pp. 6256–6261, (Florence, IT), Dec. 2013.
16. Q. Wei and D. Liu, "Optimal tracking control scheme for discrete-time nonlinear systems with approximation errors," in *Advances in Neural Networks - ISNN 2013*, C. Guo, Z.-G. Hou, and Z. Zeng, eds., *Lecture Notes in Computer Science* **7952**, pp. 1–10, Springer Berlin Heidelberg, 2013.
17. A. Heydari and S. Balakrishnan, "Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics," *IEEE Trans. Neural Netw. Learn. Syst.* **24**(1), pp. 145–157, 2013.
18. A. Heydari and S. N. Balakrishnan, "Fixed-final-time optimal control of nonlinear systems with terminal constraints," *Neural Netw.* **48**, pp. 61–71, 2013.
19. H. Modares, F. Lewis, and M.-B. Naghibi-Sistani, "Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks," *IEEE Trans. Neural Netw. Learn. Syst.* **24**(10), pp. 1513–1525, 2013.
20. R. Kamalapurkar, J. Klotz, and W. Dixon, "Concurrent learning-based online approximate feedback Nash equilibrium solution of N -player nonzero-sum differential games," *IEEE/CAA J. Autom. Sin.* **1**, pp. 239–247, July 2014.
21. R. Kamalapurkar, J. Klotz, and W. E. Dixon, "Model-based reinforcement learning for on-line feedback-nash equilibrium solution of n-player nonzero-sum differential games," in *Proc. Am. Control Conf.*, pp. 3000–3005, 2014.
22. R. Kamalapurkar, L. Andrews, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for infinite-horizon approximate optimal tracking," in *Proc. IEEE Conf. Decis. Control*, pp. 5083–5088, 2014.

23. X. Yang, D. Liu, and Q. Wei, "Online approximate optimal control for affine non-linear systems with unknown internal dynamics using adaptive dynamic programming," *IET Control Theory Appl.* **8**(16), pp. 1676–1688, 2014.
24. X. Yang, D. Liu, and D. Wang, "Reinforcement learning for adaptive optimal control of unknown continuous-time nonlinear systems with input constraints," *Int. J. Control* **87**(3), pp. 553–566, 2014.
25. H. Modares and F. L. Lewis, "Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning," *Automatica* **50**(7), pp. 1780 – 1792, 2014.
26. 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* **50**(1), pp. 193–202, 2014.
27. 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* **50**, pp. 1167–1175, April 2014.
28. A. Heydari and S. N. Balakrishnan, "Adaptive critic-based solution to an orbital rendezvous problem," *J. Guid. Control Dynam.* **37**, pp. 344–350, 2014.
29. T. Bian, Y. Jiang, and Z.-P. Jiang, "Adaptive dynamic programming and optimal control of nonlinear nonaffine systems," *Automatica* **50**(10), pp. 2624 – 2632, 2014.
30. R. Kamalapurkar, H. Dinh, S. Bhasin, and W. E. Dixon, "Approximate optimal trajectory tracking for continuous-time nonlinear systems," *Automatica* **51**, pp. 40–48, January 2015.
31. R. Kamalapurkar, J. Rosenfeld, and W. E. Dixon, "State following (StaF) kernel functions for function approximation Part ii: Adaptive dynamic programming," in *Proc. Am. Control Conf.*, pp. 521–526, 2015.
32. T. Bian, Y. Jiang, and Z.-P. Jiang, "Decentralized adaptive optimal control of large-scale systems with application to power systems," *IEEE Trans. Ind. Electron.* **62**, pp. 2439–2447, April 2015.
33. H. Modares, F. L. Lewis, and Z.-P. Jiang, "H∞ tracking control of completely unknown continuous-time systems via off-policy reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.* , 2015, to appear.
34. R. Song, F. Lewis, Q. Wei, H.-G. Zhang, Z.-P. Jiang, and D. Levine, "Multiple actor-critic structures for continuous-time optimal control using input-output data," *IEEE Trans. Neural Netw. Learn. Syst.* **26**, pp. 851–865, April 2015.
35. P. He and S. Jagannathan, "Reinforcement learning neural-network-based controller for nonlinear discrete-time systems with input constraints," *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **37**(2), pp. 425–436, 2007.
36. H. Zhang, Q. Wei, and Y. Luo, "A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy hdp iteration algorithm," *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **38**(4), pp. 937–942, 2008.
37. K. S. Narendra and A. M. Annaswamy, "A new adaptive law for robust adaptive control without persistent excitation," *IEEE Trans. Autom. Control* **32**, pp. 134–145, 1987.
38. K. Narendra and A. Annaswamy, *Stable Adaptive Systems*, Prentice-Hall, Inc., 1989.
39. S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence, and Robustness*, Prentice-Hall, Upper Saddle River, NJ, 1989.
40. P. Ioannou and J. Sun, *Robust Adaptive Control*, Prentice Hall, 1996.
41. D. Vrabie, *Online Adaptive Optimal Control For Continuous-time Systems*. PhD thesis, University of Texas at Arlington, 2010.
42. A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free q-learning designs for linear discrete-time zero-sum games with application to H_∞ control," *Automatica* **43**, pp. 473–481, 2007.
43. K. Vamvoudakis and F. Lewis, "Multi-player non-zero-sum games: Online adaptive learning solution of coupled hamilton-jacobi equations," *Automatica* **47**, pp. 1556–1569, 2011.

44. K. G. Vamvoudakis, F. L. Lewis, and G. R. Hudas, "Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality," *Automatica* **48**(8), pp. 1598 – 1611, 2012.
45. S. P. Singh, "Reinforcement learning with a hierarchy of abstract models," in *AAAI Natl. Conf. Artif. Intell.*, **92**, pp. 202–207, 1992.
46. C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *Int. Conf. Mach. Learn.*, **97**, pp. 12–20, 1997.
47. P. Abbeel, M. Quigley, and A. Y. Ng, "Using inaccurate models in reinforcement learning," in *Int. Conf. Mach. Learn.*, pp. 1–8, ACM, (New York, NY, USA), 2006.
48. M. P. Deisenroth, *Efficient reinforcement learning using Gaussian processes*, KIT Scientific Publishing, 2010.
49. D. Mitrovic, S. Klanke, and S. Vijayakumar, "Adaptive optimal feedback control with learned internal dynamics models," in *From Motor Learning to Interaction Learning in Robots*, O. Sigaud and J. Peters, eds., *Studies in Computational Intelligence* **264**, pp. 65–84, Springer Berlin Heidelberg, 2010.
50. M. P. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Int. Conf. Mach. Learn.*, pp. 465–472, 2011.
51. R. Kamalapurkar, *Model-Based Reinforcement Learning for Online Approximate Optimal Control*. PhD thesis, University of Florida, 2014.
52. D. Kirk, *Optimal Control Theory: An Introduction*, Dover, 2004.
53. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, USA, 1998.
54. V. Konda and J. Tsitsiklis, "On actor-critic algorithms," *SIAM J. Control Optim.* **42**(4), pp. 1143–1166, 2004.
55. T. Dierks and S. Jagannathan, "Optimal tracking control of affine nonlinear discrete-time systems with unknown internal dynamics," in *Proc. IEEE Conf. Decis. Control*, pp. 6750–6755, 2009.
56. K. Vamvoudakis and F. Lewis, "Online synchronous policy iteration method for optimal control," in *Recent Advances in Intelligent Control Systems*, W. Yu, ed., pp. 357–374, Springer, 2009.
57. T. Dierks and S. Jagannathan, "Optimal control of affine nonlinear continuous-time systems," in *Proc. Am. Control Conf.*, pp. 1568–1573, 2010.
58. W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*, Birkhauser: Boston, 2003.
59. G. V. Chowdhary and E. N. Johnson, "Theory and flight-test validation of a concurrent-learning adaptive controller," *J. Guid. Control Dynam.* **34**, pp. 592–607, March 2011.
60. 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.* **27**(4), pp. 280–301, 2013.
61. H. K. Khalil, *Nonlinear Systems*, Prentice Hall, Upper Saddle River, NJ, USA, 3 ed., 2002.
62. D. Garg, W. W. Hager, and A. V. Rao, "Pseudospectral methods for solving infinite-horizon optimal control problems," *Automatica* **47**(4), pp. 829 – 837, 2011.
63. W. Schmidt, "Springs of Florida," Bulletin 66, Florida Geological Survey, 2004.
64. N. Fischer, D. Hughes, P. Walters, E. Schwartz, and W. E. Dixon, "Nonlinear rise-based control of an autonomous underwater vehicle," *IEEE Trans. Robot.* **30**, pp. 845–852, Aug. 2014.
65. G. Chowdhary, *Concurrent learning adaptive control for convergence without persistencey of excitation*. PhD thesis, Georgia Institute of Technology, December 2010.
66. T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, Wiley, 2011.