Mixed Density Methods for Approximate Dynamic Programming

Max L. Greene, Patryk Deptula, Rushikesh Kamalapurkar, and Warren E. Dixon

Abstract This chapter discusses mixed density reinforcement learning (RL)-based approximate optimal control methods applied to deterministic systems. Such methods typically require a persistence of excitation (PE) condition for convergence. In this chapter, data-based methods will be discussed to soften the stringent PE condition by learning via simulation-based extrapolation. The development is based on the observation that, given a model of the system, RL can be implemented by evaluating the Bellman error (BE) at any number of desired points in the state space, thus virtually simulating the system. The sections will discuss necessary and sufficient conditions for optimality, regional model-based RL, local (StaF) RL, combining regional and local model-based RL, and RL with sparse BE extrapolation. Notes on stability follow within each method's respective section.

M. L. Greene · W. E. Dixon (✉)
Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, USA
e-mail: wdixon@ufl.edu

M. L. Greene
e-mail: maxgreene12@ufl.edu

P. Deptula
The Charles Stark Draper Laboratory, Inc., Cambridge, MA, USA
e-mail: pdeptula@draper.com

R. Kamalapurkar
Department of Mechanical and Aerospace Engineering, Mechanical and Aerospace Engineering, Stillwater, OK, USA
e-mail: rushikesh.kamalapurkar@okstate.edu

K. G. Vamvoudakis et al. (eds.), *Handbook of Reinforcement Learning and Control*, Studies in Systems, Decision and Control 325,
https://doi.org/10.1007/978-3-030-60990-0_5

490511_1_En_5_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:19/11/2020 Pages: xxx Layout: T1-Standard

1 Introduction

Reinforcement learning (RL) enables a cognitive agent to learn desirable behavior from interactions with its environment. In control theory, the desired behavior is typically quantified using a cost function, and the control problem is formulated to find the optimal policy that minimizes a cumulative cost function. Leveraging function approximation architectures, RL-based techniques have been developed to approximately solve optimal control problems for continuous-time and discrete-time deterministic systems by computing the optimal policy based on an estimate of the optimal cost-to-go function, i.e., the value function (cf., [1–12]). In RL-based approximate online optimal control, the Hamilton–Jacobi–Bellman equation (HJB), along with an estimate of the state derivative (cf. [6, 9]), or an integral form of the HJB (cf. [13, 14]), is utilized to approximately measure the quality of the estimate of the value function evaluated at each visited state along the system trajectory. This measurement is called the Bellman error (BE).

In online RL-based techniques, estimates for the uncertain parameters in the value function are updated using the BE as a performance metric. Hence, the unknown value function parameters are updated based on the evaluation of the BE along the system trajectory. In particular, the integral BE is meaningful as a measure of quality of the value function estimate only if evaluated along the system trajectories, and state derivative estimators can only generate numerical estimates of the state derivative along the system trajectories. Online RL-based techniques can be implemented in either model-based or model-free form. Generally speaking, both approaches have their respective advantages and disadvantages. Model-free approaches learn optimal actions without requiring knowledge of the system [15]. Model-based approaches 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 [15].

Methods that seek online solutions to optimal control problems are comparable to adaptive control (cf., [2, 7, 9, 11, 16, 17] and the references therein), where the estimates for the uncertain parameters in the plant model are updated using the tracking error as a performance metric. Similarly, in approximate dynamic programming (ADP), the BE is used as a performance metric. Parameter convergence has long been a focus of research in adaptive control. 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.

Least squares and gradient descent-based update laws are used in RL-based techniques to solve optimal control problems online [15, 18]. Such update laws generally require persistence of excitation (PE) in the system state for parameter estimate convergence. Hence, the challenges are that the updated law must be PE and the system trajectory needs to visit enough points in the state space to generate a sufficient approximation of the value function over the entire domain of operation. These chal-

490511_1_En_5_Chapter ✓ TYPESET ☐ DISK ☐ LE ✓ CP Disp.:19/11/2020 Pages: xxx Layout: T1-Standard

lenges are often addressed in the related literature (cf. [4, 7, 9, 19–25]) by adding an exploration signal to the control input. However, no analytical methods exist to compute the appropriate exploration signal when the system dynamics are nonlinear.

Unlike results requiring PE, this chapter discusses model-based approaches used to mitigate the need to inject probing signals into the system to facilitate learning. In Sect. 2, the infinite horizon optimal control problem is introduced along with conditions which establish the optimal control policy. It is shown that the value function is the optimal cost-to-go and satisfies the HJB equation. In Sect. 3, the regional model-based RL (R-MBRL) method is presented where unknown weights in the value function are adjusted based on least square minimization of the BE evaluated at any number of user-selected arbitrary trajectories in the state space. Since the BE can be evaluated at any desired point in the state space, sufficient exploration is achieved by selecting points distributed over the system's operating domain. R-MBRL establishes online approximate learning of the optimal controller while maintaining overall system stability. In Sect. 4, the local state following MBRL (StaF-RL) method is presented where the computational complexity of MBRL problems is reduced by estimating the optimal value function within a local domain around the state. A reduction in the computational complexity via StaF-RL is achieved by reducing the number of basis functions required for approximation. 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. That is, since StaF-RL estimates the value function in a local domain around the state, the value function approximation is a local solution. In Sect. 5, a strategy that uses R-MBRL and StaF-RL together to approximate the value function is described. This technique eliminates the need to perform BE extrapolation over a large region of the state space, as in R-MBRL, and the inability for the StaF method to develop a global estimate of the value function. Specifically, using knowledge about where the system is to converge, a R-MBRL approximation is used in the regional neighborhood to maintain an accurate approximation of the value function of the goal location. Moreover, to ensure stability of the system before entering the regional neighborhood, StaF-RL is leveraged to guide the system to the regional neighborhood. In Sect. 6, a strategy is described to overcome the computational cost of R-MBRL by using a set of sparse off-policy trajectories, which are used to calculate extrapolated BEs. Furthermore, the state-space is divided into a user-selected number of segments. Drawing inspiration from the approach in Sect. 5, a certain set of trajectories, and, hence, sets of extrapolated BEs, can be marked active when the state enters the corresponding segment, i.e., the only active set of extrapolated BEs are those that belong to the same segment as the current trajectory. Sparse neural networks (SNNs) could then be used within each segment to extrapolate the BE due to their small amount of active neurons, whose activity can be switched on or off based on the active segment, to make BE extrapolation more computationally efficient.

490511_1_En_5_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:19/11/2020 Pages: xxx Layout: T1-Standard

2 Unconstrained Affine-Quadratic Regulator

Consider a controlled dynamical system described by the initial value problem

$$\dot{x} = f(x, u, t), \; x(t_0) = x_0, , \tag{1}$$

where t_0 is the initial time, $x \in \mathbb{R}^n$ denotes the system state, $u \in U \subset \mathbb{R}^m$ denotes the control input, and U denotes the action space. Consider a family (parameterized by t) of optimal control problems described by the cost functionals

$$J(t, y, u(\cdot)) = \int_t^\infty L(\phi(\tau; t, y, u(\cdot)), u(\tau), \tau)\, \mathrm{d}\tau, \tag{2}$$

where $L : \mathbb{R}^n \times U \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ is the Lagrange cost, with $L(x, u, t) \geq 0$, for all $(x, u, t) \in \mathbb{R}^n \times U \times \mathbb{R}_{\geq 0}$, and the notation $\phi(\tau; t, y, u(\cdot))$ is used to denote a trajectory of the system in (1), evaluated at time τ, under the controller $u : \mathbb{R}_{\geq t_0} \to U$, starting at the initial time t, and with the initial state y. The short notation $x(\tau)$ is used to denote $\phi(\tau; t, y, u(\cdot))$ when the controller, the initial time, and the initial state are clear from the context. Throughout this discussion, it is assumed that the controllers and the dynamical systems are such that the initial value problem in (1) admits a unique complete solution starting from any initial condition.

Let the optimal value function $V^* : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ be defined as

$$V^*(x, t) : \inf_{u_{[t,\infty)} \in \mathcal{U}_{(t,x)}} J(t, x, u(\cdot)), \tag{3}$$

where the notation $u_{[t,\infty)}$ for $t \geq t_0$ denotes the controller $u(\cdot)$ restricted to the time interval $[t, \infty)$ and $\mathcal{U}_{(t,x)}$ denotes the set of controllers that are admissible for x. The following theorem is a generalization of [26, Theorem 1.2] to infinite horizon problems.

Theorem 1 *Given $t_0 \in \mathbb{R}_{\geq 0}$, $x_0 \in \mathbb{R}^n$, let $\mathcal{U}_{(t_0,x_0)}$ include all Lebesgue measurable locally bounded controllers so that the initial value problem in (1) admits a unique complete solution starting from (t_0, x_0). Assume $V^* \in \mathcal{C}^1\left(\mathbb{R}^n \times \mathbb{R}_{\geq t_0}, \mathbb{R}\right)$. If there exists a function $V : \mathbb{R}^n \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$ such that*

1. $V \in \mathcal{C}^1\left(\mathbb{R}^n \times \mathbb{R}_{\geq t_0}, \mathbb{R}\right)$ and V satisfies the HJB equation

$$0 = -\nabla_t V(x, t) - \inf_{\mu \in U} \left\{ L(x, \mu, t) + V'^T(x, t) f(x, \mu, t) \right\}, \tag{4}$$

   for all $t \in [t_0, \infty)$ and all $x \in \mathbb{R}^n$,[1]

[1] The notation $\nabla_x h(x, y, t)$ denotes the partial derivative of generic function $h(x, y, t)$ with respect to generic variable x. The notation $h'(x, y)$ denotes the gradient with respect to the first argument of the generic function, $h(\cdot, \cdot)$, e.g., $h'(x, y) = \nabla_x h(x, y)$.

2. for every controller $v(\cdot) \in \mathcal{U}_{(t_0,x_0)}$ for which there exists $M_v \geq 0$ so that $\int_{t_0}^{t} L(\phi(\tau, t_0, x_0, v(\cdot)), v(\tau), \tau)\, d\tau \leq M_v$ for all $t \in \mathbb{R}_{\geq t_0}$, the function V, evaluated along the resulting trajectory, satisfies

$$\lim_{t\to\infty} V(\phi(t; t_0, x_0, v(\cdot))) = 0, \tag{5}$$

and
3. there exists $u(\cdot) \in \mathcal{U}_{(t_0,x_0)}$, such that the function V, the controller $u(\cdot)$, and the trajectory $x(\cdot)$ of (1) under $u(\cdot)$ with the initial condition $x(t_0) = x_0$, satisfy, the Hamiltonian minimization condition

$$\begin{aligned} &L(x(t), u(t), t) + V'^T(x(t), t) f(x(t), u(t), t) \\ &= \min_{\mu\in U} \left\{ L(x(t), \mu, t) + V'^T(x(t), t) f(x(t), \mu, t) \right\}, \quad \forall t \in \mathbb{R}_{\geq t_0}, \end{aligned} \tag{6}$$

and the bounded cost condition

$$\exists M_u \geq 0 \quad | \quad \int_{t_0}^{t} L(x(\tau), v(\tau), \tau)\, d\tau \leq M_u, \quad \forall t \in \mathbb{R}_{\geq t_0}, \tag{7}$$

then, $V(t_0, x_0)$ is the optimal cost (i.e., $V(t_0, x_0) = V^*(t_0, x_0)$) and $u(\cdot)$ is an optimal controller.

Proof Let $x(\cdot) \triangleq \phi(\cdot; t_0, x_0, u(\cdot))$, where $u(\cdot)$ is an admissible controller that satisfies (6) and (7), and $y(\cdot) \triangleq \phi(\cdot; t_0, x_0, v(\cdot))$ where $v(\cdot)$ is any other admissible controller. The Hamiltonian minimization condition in (6) implies that along the trajectory $x(\cdot)$, the control $\mu = u(t)$ achieves the infimum in (4) for all $t \in \mathbb{R}_{\geq t_0}$. Thus, along the trajectory $x(\cdot)$, (4) implies that

$$-\nabla_t V(x(t), t) - V'^T(x(t), t) f(x(t), u(t), t) = L(x(t), u(t), t).$$

That is,

$$-\frac{d}{dt} V(x(t), t) = L(x(t), u(t), t).. \tag{8}$$

Since V satisfies the HJB equation everywhere, it is clear that along the trajectory $y(\cdot)$,

$$\begin{aligned} &\inf_{\mu\in U} \left\{ L(y(t), \mu, t) + V'^T(y(t), t) f(y(t), \mu, t) \right\} \\ &\qquad \leq L(y(t), v(t), t) + V'^T(y(t), t) f(y(t), v(t), t) \end{aligned}$$

and as a result, the HJB equation, evaluated along $y(\cdot)$, yields

$$0 \geq -\nabla_t V(y(t), t) - V'^T(y(t), t) f(y(t), v(t), t) - L(y(t), v(t), t).$$

That is,

$$-\frac{\mathrm{d}}{\mathrm{d}t}V\left(y\left(t\right),t\right)\leq L\left(y\left(t\right),v\left(t\right),t\right).. \tag{9}$$

Integrating (8) and (9) over a finite interval $[t_0, T]$,

$$\begin{aligned}-\int_{t_0}^{T}\frac{\mathrm{d}}{\mathrm{d}t}V\left(x\left(t\right),t\right)\,\mathrm{d}t&=\left(V\left(x\left(t_0\right),t_0\right)-V\left(x\left(T\right),T\right)\right)\\&=\int_{t_0}^{T}L\left(x\left(t\right),u\left(t\right),t\right)\,\mathrm{d}t\end{aligned}$$

and

$$\begin{aligned}-\int_{t_0}^{T}\frac{\mathrm{d}}{\mathrm{d}t}V\left(y\left(t\right),t\right)\,\mathrm{d}t&=\left(V\left(y\left(t_0\right),t_0\right)-V\left(y\left(T\right),T\right)\right)\\&\leq\int_{t_0}^{T}L\left(y\left(t\right),v\left(t\right),t\right)\,\mathrm{d}t.\end{aligned}$$

Since $x\left(t_0\right)=y\left(t_0\right)=x_0$, it can be concluded that

$$\begin{aligned}V\left(x_0,t_0\right)&=\int_{t_0}^{T}L\left(x\left(t\right),u\left(t\right),t\right)\,\mathrm{d}t+V\left(x\left(T\right),T\right)\\&\leq\int_{t_0}^{T}L\left(y\left(t\right),v\left(t\right),t\right)\,\mathrm{d}t+V\left(y\left(T\right),T\right),\forall T\in\mathbb{R}_{\geq t_0},\end{aligned}$$

and as a result,

$$\begin{aligned}V\left(x_0,t_0\right)&=\lim_{T\to\infty}\int_{t_0}^{T}L\left(x\left(t\right),u\left(t\right),t\right)\,\mathrm{d}t+V\left(x\left(T\right),T\right)\\&\leq\lim_{T\to\infty}\int_{t_0}^{T}L\left(y\left(t\right),v\left(t\right),t\right)\,\mathrm{d}t+V\left(y\left(T\right),T\right).\end{aligned}$$

Since $u\left(\cdot\right)$ satisfies (7) and $\left(x,u,t\right)\mapsto L\left(x,u,t\right)$ is nonnegative,

$$\int_{t_0}^{\infty}L\left(x\left(t\right),u\left(t\right),t\right)\,\mathrm{d}t$$

exists, is bounded, and equal to the total cost $J\left(t_0,x_0,u\left(\cdot\right)\right)$. Taking (5) into account, it can thus be concluded that

$$\lim_{T\to\infty}\int_{t_0}^{T}L\left(x\left(t\right),u\left(t\right),t\right)\,\mathrm{d}t+V\left(x\left(T\right),T\right)=J\left(t_0,x_0,u\left(\cdot\right)\right).$$

If $v(\cdot)$ satisfies (7), then a similar analysis yields

$$\lim_{T\to\infty}\int_{t_0}^{T} L\left(y(t), v(t), t\right)\,\mathrm{d}t + V\left(y(T), T\right) = J\left(t_0, x_0, v(\cdot)\right),$$

and as a result,

$$V\left(x_0, t_0\right) = J\left(t_0, x_0, u(\cdot)\right) \le J\left(t_0, x_0, v(\cdot)\right). \tag{10}$$

If $v(\cdot)$ does not satisfy (7), then nonnegativity of $(x, u, t) \mapsto L(x, u, t)$ implies that the total cost resulting from $v(\cdot)$ is unbounded, and (10) holds canonically. In conclusion, $V(t_0, x_0)$ is the optimal cost (i.e., $V(t_0, x_0) = V^*(t_0, x_0)$) and $u(\cdot)$ is an optimal controller.

For the remainder of this section, a controller $v : \mathbb{R}_{\ge t_0} \to U$ is said to be admissible for a given initial state (t_0, x_0) if it is bounded, generates a unique bounded trajectory starting from x_0, and results in bounded total cost. An admissible controller that results in the smallest cost among all admissible controllers is called an optimal controller. The dynamics and the Lagrange cost are assumed to be time-invariant, and as a result, if $v : \mathbb{R}_{\ge t_0} \to U$ is admissible for a given initial state (t_0, x_0), then $v' : \mathbb{R}_{\ge t_1} \to U$, defined as $v'(t) \triangleq v(t + t_0 - t_1)$, for all $t \in \mathbb{R}_{\ge t_1}$ is admissible for (t_1, x_0), and trajectories of the system starting from (t_0, x_0) under $v(\cdot)$ and those starting from (t_1, x_0) under $v'(\cdot)$ are identical. As a result, the set of admissible controllers, system trajectories, value functions, and total costs can be considered independent of t_0 without loss of generality. The following two theorems prove the claim in item 2 of Theorem 1.

Theorem 2 *Consider the optimal control problem*

$$P: \qquad \min_{u(\cdot)\in\mathcal{U}_{x_0}} \quad J\left(x_0, u(\cdot)\right) \triangleq \int_{t_0}^{\infty} r\left(\phi\left(\tau; t_0, x_0, u(\cdot)\right), u(\tau)\right)\mathrm{d}\tau \tag{11}$$

subject to

$$\dot{x} = f(x) + g(x)\,u, \tag{12}$$

where the local cost $r : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is defined as $r(x, u) \triangleq Q(x) + u^T R u$, with $Q : \mathbb{R}^n \to \mathbb{R}$, a continuously differentiable positive definite function and $R \in \mathbb{R}^{m\times m}$, a symmetric positive definite matrix. Assume further that the optimal value function $V^* : \mathbb{R}^n \to \mathbb{R}$ corresponding to P is continuously differentiable.

If $x \mapsto V(x)$ is positive definite and satisfies the closed-loop HJB equation

$$r\left(x, \psi(x)\right) + V'(x)\left(f(x) + g(x)\,\psi(x)\right) = 0, \qquad \forall x \in \mathbb{R}^n, \tag{13}$$

with

$$\psi(x) = -\frac{1}{2} R^{-1} g^T(x) \left(V'(x)\right)^{\mathrm{T}}, \tag{14}$$

then $V(\cdot)$ is the optimal value function and the the state feedback law $u(t) = \psi(x(t))$ is the optimal controller.

Proof Note that (13) and the positive definiteness of Q, R, and V, imply that under the state feedback law $u(t) = \psi(x(t))$, the closed-loop system $\dot{x} = f(x) + g(x)\psi(x)$ is globally asymptotically stable. Furthermore, since $V(0) = 0$, every trajectory of the closed-loop system converges to the origin and since (13) holds for all $x \in \mathbb{R}^n$, and in particular, holds along every trajectory of the closed-loop system, it can be concluded that

$$\int_t^\infty r(x(\tau), \psi(x(\tau)))\, \mathrm{d}t = V(x(t)) = J(x(t), \psi(x(\cdot))), \forall t \in \mathbb{R}$$

along every trajectory of the closed-loop system. As a result, all control signals resulting from the state-feedback law $u(t) = \psi(x(t))$ are admissible for all initial conditions. For each $x \in \mathbb{R}^n$ it follows that

$$\frac{\partial\left(r(x,u) + V'(x)(f(x) + g(x)u)\right)}{\partial u} = 2u^T R + V'(x) g(x).$$

Hence, $u = -\frac{1}{2} R^{-1} g^T(x) \left(V'(x)\right)^T = \psi(x)$ extremizes

$$r(x,u) + V'(x)(f(x) + g(x)u).$$

Furthermore, the Hessian

$$\frac{\partial^2\left(r(x,u) + V'(x)(f(x) + g(x)u)\right)}{\partial^2 u} = 2R$$

is positive definite. Hence, $u = \psi(x)$ minimizes

$$u \mapsto r(x,u) + V'(x)(f(x) + g(x)u)$$

for each $x \in \mathbb{R}^n$.

As a result, the closed-loop HJB equation (13), along with the control law (14) are equivalent to the HJB equation (4). Furthermore, all trajectories starting from all initial conditions in response to the controller $u(t) = \psi(x(t))$ satisfy the Hamiltonian minimization condition (6) and the bounded cost condition (7). In addition, given any initial condition x_0 and a controller $v(\cdot)$ that is admissible for x_0, boundedness of the controller $v(\cdot)$ and the resulting trajectory $\phi(\cdot; t_0, x_0, v(\cdot))$, continuity of $x \mapsto f(x,u)$ and $x \mapsto g(x,u)$, and continuity of $x \mapsto Q'(x)$ can be used to conclude that $t \mapsto Q(\phi(t; t_0, x_0, v(\cdot)))$ is uniformly continuous.

Admissibility of $v(\cdot)$ and positive definiteness of R imply that

$$\left|\int_{t_0}^{T} Q\left(\phi\left(t; t_0, x_0, v\left(\cdot\right)\right)\right) \mathrm{d}t\right| \leq M$$

for all $T \geq t_0$ and some $M \geq 0$. Furthermore, positive definiteness of $x \mapsto Q\left(x\right)$ implies monotonicity of $T \mapsto \int_{t_0}^{T} Q\left(\phi\left(t; t_0, x_0, v\left(\cdot\right)\right)\right) \mathrm{d}t$. As a result, the limit $\lim_{T\to\infty} \int_{t_0}^{T} Q\left(\phi\left(t; t_0, x_0, v\left(\cdot\right)\right)\right) \mathrm{d}t$ exists and is bounded.

By Barbalat's lemma, $\lim_{t\to\infty} Q\left(\phi\left(t; t_0, x_0, v\left(\cdot\right)\right)\right) = 0$, which, due to positive definiteness and continuity of $x \mapsto Q\left(x\right)$ implies that

$$\lim_{t\to\infty} \phi\left(t; t_0, x_0, v\left(\cdot\right)\right) = 0,$$

and finally, from continuity and positive definiteness of V,

$$\lim_{t\to\infty} V\left(\phi\left(t; t_0, x_0, v\left(\cdot\right)\right)\right) = 0,$$

which establishes (5).

Arguments similar to the proof of Theorem 1 can then be invoked to conclude that $V\left(x_0\right)$ is the optimal cost and $u\left(t\right) = \psi\left(x\left(t\right)\right)$ is the unique optimal controller among all admissible controllers. Since the initial condition was arbitrary, the proof of Theorem 2 is complete.

To facilitate the following discussion, let $\mathcal{U}_{x,[t_1,t_2]}$ denote the space of controllers that are restrictions over $[t_1, t_2]$ of controllers admissible for x. The task is then to show that value functions satisfy HJB equations.

Theorem 3 *Consider the optimal control problem P stated in Theorem 2 and assume that for every initial condition x_0, an optimal controller that is admissible for x_0 exists. If the optimal value function corresponding to P, defined as*

$$V^*\left(x\right) \triangleq \inf_{u(\cdot)\in\mathcal{U}_x} \int_{t}^{\infty} r\left(\phi\left(\tau; t, x, u\left(\cdot\right)\right), u\left(\tau\right)\right) \mathrm{d}\tau, \tag{15}$$

is continuously differentiable then it satisfies the HJB equation

$$r\left(x, \psi\left(x\right)\right) + V^{*\prime}\left(x\right)\left(f\left(x\right) + g\left(x\right)\psi^*\left(x\right)\right) = 0, \qquad \forall x \in \mathbb{R}^n, \tag{16}$$

with

$$\psi^*\left(x\right) = -\frac{1}{2}R^{-1}g^T\left(x\right)\left(V^{*\prime}\left(x\right)\right)^{\mathrm{T}} \tag{17}$$

being the optimal feedback policy.

490511_1_En_5_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:19/11/2020 Pages: xxx Layout: T1-Standard

Proof First, it is shown that the value function satisfies the principle of optimality. To facilitate the discussion, given $x \in \mathbb{R}^n$, let $v^*_{(x,t)} : \mathbb{R}_{\geq t} \to U$ denote an optimal controller starting from the initial state x and initial time t.

Claim (Principle of optimality under admissibility restrictions) For all $x \in \mathbb{R}^n$, and for all $\Delta t > 0$,

$$V^*(x) = \inf_{u(\cdot) \in \mathcal{U}_{x,[t,t+\Delta t]}} \left\{ \int_t^{t+\Delta t} r\left(\phi\left(\tau; t, x, u(\cdot)\right), u(\tau)\right) \mathrm{d}\tau + V^*\left(x\left(t+\Delta t\right)\right) \right\}. \tag{18}$$

Proof Consider the function $V : \mathbb{R}^n \to \mathbb{R}$ defined as

$$V(x) \triangleq \inf_{u(\cdot) \in \mathcal{U}_{x,[t,t+\Delta t]}} \left\{ \int_t^{t+\Delta t} r\left(\phi\left(\tau; t, x, u(\cdot)\right), u(\tau)\right) \mathrm{d}\tau + V^*\left(x\left(t+\Delta t\right)\right) \right\}.$$

Based on the definition in (15)

$$V(x) = \inf_{u(\cdot) \in \mathcal{U}_{x,[t,t+\Delta t]}} \left\{ \int_t^{t+\Delta t} r\left(\phi\left(\tau; t, x, u(\cdot)\right), u(\tau)\right) \mathrm{d}\tau + \inf_{v(\cdot) \in \mathcal{U}_{x(t+\Delta t)}} \int_{t+\Delta t}^{\infty} r\left(\phi\left(\tau; t, x\left(t+\Delta t\right), v(\cdot)\right), v(\tau)\right) \mathrm{d}\tau \right\}.$$

Since the first integral is independent of the control over $\mathbb{R}_{\geq t+\Delta t}$,

$$V(x) = \inf_{u(\cdot) \in \mathcal{U}_{x,[t,t+\Delta t]}} \inf_{v(\cdot) \in \mathcal{U}_{x(t+\Delta t)}} \left\{ \int_t^{t+\Delta t} r\left(\phi\left(\tau; t, x, u(\cdot)\right), u(\tau)\right) \mathrm{d}\tau + \int_{t+\Delta t}^{\infty} r\left(\phi\left(\tau; t, x\left(t+\Delta t\right), v(\cdot)\right), v(\tau)\right) \mathrm{d}\tau \right\}.$$

Combining the integrals and using the fact that concatenation of admissible restrictions and admissible controllers result in admissible controllers, $\inf_{u(\cdot) \in \mathcal{U}_{x,[t,t+\Delta t]}} \inf_{v(\cdot) \in \mathcal{U}_{x(t+\Delta t)}}$ is equivalent to $\inf_{w(\cdot) \in \mathcal{U}_x}$, where $w : \mathbb{R}_{\geq t} \to U$ is defined as $w(\tau) : \begin{cases} u(\tau) & t \leq \tau \leq t + \Delta t, \\ v(\tau) & \tau > t + \Delta t, \end{cases}$ it can be concluded that

$$V(x) = \inf_{w(\cdot)\in\mathcal{U}_x} \int_t^{\infty} r\left(\phi\left(\tau; t, x, w(\cdot)\right), w(\tau)\right) \mathrm{d}\tau = V^*(x).$$

Thus,

$$V(x) \geq V^*(x). \tag{19}$$

On the other hand, by the definition of the infimum, for all $\epsilon > 0$, there exists an admissible controller $u_\epsilon(\cdot)$ such that

$$V^*(x) + \epsilon \geq J(x, u_\epsilon(\cdot)).$$

Let $x_\epsilon : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ denote the trajectory corresponding to $u_\epsilon(\cdot)$. Since the restriction $u_{\epsilon,\mathbb{R}_{\geq t_1}}(\cdot)$ of $u_\epsilon(\cdot)$ to $\mathbb{R}_{\geq t_1}$ is admissible for $x_\epsilon(t_1)$ for all $t_1 > t_0$,

$$\begin{aligned} J(x, u_\epsilon(\cdot)) &= \int_t^{t+\Delta t} r\left(x_\epsilon(\tau), u_\epsilon(\tau)\right) \mathrm{d}\tau + J\left(x_\epsilon(t+\Delta t), u_{\epsilon,\mathbb{R}_{\geq t+\Delta t}}(\cdot)\right) \\ &\geq \int_t^{t+\Delta t} r\left(x_\epsilon(\tau), u_\epsilon(\tau)\right) \mathrm{d}\tau + V^*\left(x_\epsilon(t+\Delta t)\right) \geq V(x). \end{aligned}$$

Thus, $V(x) \leq V^*(x)$, which, along with (19), implies $V(x) = V^*(x)$.

Since $V^* \in \mathcal{C}^1(\mathbb{R}^n, \mathbb{R})$, given any admissible $u(\cdot)$ and corresponding trajectory $x(\cdot)$,

$$V^*(x(t+\Delta t)) = V^*(x) + V^{*\prime}(x)\left(\left(f(x) + g(x)u(t)\right)\Delta t\right) + o(\Delta t).$$

Furthermore,

$$\int_t^{t+\Delta t} r\left(x(\tau), u(\tau)\right) \mathrm{d}\tau = r\left(x, u(t)\right)\Delta t + o(\Delta t).$$

From the principle of optimality in (18),

$$\begin{aligned} V^*(x) = \inf_{u(\cdot)\in\mathcal{U}_{x,[t,t+\Delta t]}} &\Big\{ r\left(x, u(t)\right)\Delta t + V^*(x) \\ &+ V^{*\prime}(x)\left(\left(f(x) + g(x)u(t)\right)\Delta t\right) + o(\Delta t)\Big\}. \end{aligned}$$

That is,

$$\begin{aligned} 0 = \inf_{u(\cdot)\in\mathcal{U}_{x,[t,t+\Delta t]}} &\{ r\left(x, u(t)\right)\Delta t \\ &+ V^{*\prime}(x)\left(\left(f(x) + g(x)u(t)\right)\Delta t\right) + o(\Delta t)\Big\}. \end{aligned}$$

Dividing by Δt and taking the limit as Δt goes to zero,

$$0 = \inf_{u \in U} \left\{ r\left(x, u\right) + V^{*\prime}\left(x\right)\left(f\left(x\right) + g\left(x\right)u\right) \right\}, \quad \forall x \in \mathbb{R}^{n}.$$

In conclusion, under the assumptions made in this section, the optimal value function is continuously differentiable, positive definite, and satisfies the HJB equation. All functions that are continuously differentiable and positive definite and satisfy the HJB equation are optimal value functions, and optimal value functions are, by definition, unique. As a result, if there is a continuously differentiable and positive definite solution of the HJB equation then it is unique and is also the optimal value function.

3 Regional Model-Based Reinforcement Learning

The following section examines the dynamical system in (1) and a controller, u, is designed to solve the infinite horizon optimal control problem via a R-MBRL approach. The R-MBRL technique, described in detail in [15], uses a data-based approach 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 arbitrary points in the state space. For instance, in an infinite horizon regulation problem, the BE can be computed at points uniformly distributed in a neighborhood around the origin of the state space. 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 to cover the domain of operation.

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 the experience. Learning based on simulation of experience has been investigated in results such as [27–32] for stochastic model-based RL; however, these results solve the optimal control problem offline 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.

The following subsections explore nonlinear, control affine plants and provides an online solution to a deterministic infinite horizon optimal regulation problems by leveraging BE extrapolation.

3.1 Preliminaries

Consider a control affine nonlinear dynamical system in (12).[2]

Assumption 1 The drift dynamic, f, is a locally Lipschitz function such that $f(0) = 0$, and the control effectiveness, g, is a known bounded locally Lipschitz function. Furthermore, $f' : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ is continuous.

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 function in (11) under the dynamic constraint in (12) while regulating the system state to the origin.

It is well known that since the functions f, g, , and Q are stationary (time-invariant) and the time horizon is infinite, the optimal control input is a stationary state feedback policy. Furthermore, the value function is also a stationary function [33, Eq. 5.19]. Hence, the optimal value function $V^* : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ can be expressed in (15) while regulating the system states to the origin (i.e., $x = 0$) for $\tau \in \mathbb{R}_{\geq t}$, with $u(\tau) \in \mathbb{U} | \tau \in \mathbb{R}_{\geq t}$, where $\mathbb{U} \in \mathbb{R}^m$ denotes the set of admissible inputs, which, by Theorem 2, are admissible for all initial conditions. In (15), $r : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ denotes the instantaneous cost defined as $r(x, u) \triangleq x^T Q x + u^T R u$, where $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ are constant positive definite (PD) symmetric matrices.

Property 1 *The PD matrix Q satisfies $\underline{q} I_n \leq Q \leq \overline{q} I_n$ for $\underline{q}, \overline{q} \in \mathbb{R}_{>0}$.*[3]

Provided the assumptions and conditions in Sect. 2 regarding a unique solution hold, the optimal value function, V^*, is the solution to the corresponding HJB equation in (16) with boundary condition $V^*(0) = 0$ [34, Sect. 3.11]. From Theorems 2 and 3, provided the HJB in (16) admits a continuously differentiable and positive definite solution, it constitutes a necessary and sufficient condition for optimality. The optimal control policy, $u^* : \mathbb{R}^n \to \mathbb{R}^m$, is defined as

$$u^* = -\frac{1}{2} R^{-1} g^T(x) \left(V^{*\prime}(x) \right)^T. \tag{20}$$

3.2 Regional Value Function Approximation

Approximations of the optimal value function, V^*, and the optimal policy, u^*, are designed based on neural network (NN) representations. Given any compact set $\chi \subset \mathbb{R}^n$ and a positive constant $\overline{\epsilon} \in \mathbb{R}$, the universal function approximation property of NNs can be exploited to represent the optimal value function as

---

[2]For notational brevity, unless otherwise specified, the domain of all the functions is assumed to be $\mathbb{R}_{\geq 0}$, where $\mathbb{R}_{\geq a}$ denotes the interval $[a, \infty)$. The notation $\|\cdot\|$ denotes the Euclidean norm for vectors and the Frobenius norm for matrices.

[3]*The notation I_n denotes the $n \times n$ identity matrix.*

$$V^*(x) = W^T \sigma(x) + \epsilon(x), \tag{21}$$

for all $x \in \chi$, where $W \in \mathbb{R}^L$ is the ideal weight matrix bounded above by a known positive constant $\bar{W}$ in the sense that $\|W\| \le \bar{W}$, $\sigma : \mathbb{R}^n \to \mathbb{R}^L$ 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 $\sup_{x\in\chi} |\epsilon(x)| \le \bar{\epsilon}$ and$\sup_{x\in\chi} |\epsilon'(x)| \le \bar{\epsilon}$.

Using Assumptions 1, conditions in Sect. 2, and based on the NN representation of the value function, a NN representation of the optimal controller is derived from (20), where $\hat{V} : \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}$ and $\hat{u} : \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}^m$ denote value function and controller estimates defined as

$$\hat{V}\left(x, \hat{W}_c\right) \triangleq \hat{W}_c^T \sigma(x), \tag{22}$$

$$\hat{u}\left(x, \hat{W}_a\right) \triangleq -\frac{1}{2} R^{-1} g^T(x) \left(\sigma'(x)\right)^T \hat{W}_a. \tag{23}$$

In (22) and (23), $\hat{W}_c \in \mathbb{R}^L$ and $\hat{W}_a \in \mathbb{R}^L$ denote the critic and actor estimates of W, respectively. 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 critic weight estimates $\hat{W}_c$, enabling a least squares-based adaptive update law.

3.3 Bellman Error

In traditional RL-based algorithms, the value function and policy estimates 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., [6, 35, 36]), whereas in deterministic systems, a probing noise is added to the derived control law (cf., [7, 9, 37–39]).

Learning-based techniques often require PE to achieve convergence. The PE condition is relaxed in [24] to a finite excitation condition by using integral RL along with experience replay, where each evaluation of the BE along the system trajectory is interpreted as gained experience. These experiences are stored in a history stack and are repeatedly used in the learning algorithm to improve data efficiency. In this chapter, a different approach is used to circumvent the PE condition. Using (22) and (23) in (13) results in the BE $\delta : \mathbb{R}^n \times \mathbb{R}^L \times \mathbb{R}^L \to \mathbb{R}$, defined as

$$\begin{aligned}\delta\left(x, \hat{W}_c, \hat{W}_a\right) \triangleq\ & \hat{V}'\left(x, \hat{W}_c\right)\left(f(x) + g(x)\,\hat{u}\left(x, \hat{W}_a\right)\right) \\ & + \hat{u}\left(x, \hat{W}_a\right)^T R \hat{u}\left(x, \hat{W}_a\right) + x^T Q x.\end{aligned} \tag{24}$$

Given a model of the system and the current parameter estimates $\hat{W}_c(t)$ and $\hat{W}_a(t)$, the BE in (24) can be evaluated at any point $x_i \in \chi$. The critic can gain experience on how well the value function is estimated at any arbitrary point x_i in the state space without actually visiting x_i. Given a fixed state x_i and a corresponding planned action $\hat{u}\left(x_i, \hat{W}_a\right)$, the critic can use the dynamic model to simulate a visit to x_i by computing the state derivative at x_i. This results in simulated experience quantified by the BE. The technique developed in this section implements simulation of experience in a model-based RL scheme by extrapolating the approximate BE to a user-specified set of trajectories $\{x_i \in \mathbb{R}^n \mid i = 1, \cdots, N\}$ in the state space. The BE in (24) is evaluated along the trajectories of (12) to get the instantaneous BE $\delta_t : \mathbb{R}_{\geq t_0} \to \mathbb{R}$ defined as $\delta_t(t) \triangleq \delta\left(x(t), \hat{W}_c(t), \hat{W}_a(t)\right)$. Moreover, extrapolated trajectories $\{x_i \in \mathbb{R}^n \mid i = 1, \cdots, N\}$ are leveraged to generate an extrapolated BE $\delta_{ti} : \mathbb{R}_{\geq t_0} \to \mathbb{R}$ defined as $\delta_{ti}(t) \triangleq \delta\left(x_i, \hat{W}_c(t), \hat{W}_a(t)\right)$.

Defining the mismatch between the estimates and the ideal values as $\tilde{W}_c \triangleq W - \hat{W}_c$ and $\tilde{W}_a \triangleq W - \hat{W}_a$, substituting (20) and (21) in (13), and subtracting from (24) yields the analytical BE given by

$$\delta = \omega^T \tilde{W}_c + \frac{1}{4} \tilde{W}_a^T G_\sigma \tilde{W}_a + O(\varepsilon), \tag{25}$$

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

$$\omega\left(x, \hat{W}_a\right) \triangleq \sigma'(x)\left(f(x) + g(x)\hat{u}\left(x, \hat{W}_a\right)\right),$$

and $O(\varepsilon) \triangleq \frac{1}{4} G_\varepsilon - \varepsilon' f + \frac{1}{2} W^T \sigma' G \varepsilon'^T$.[4] Since the HJB in (13) is equal to zero for all $x \in \mathbb{R}^n$, the aim is to find critic and actor weight estimates, $\hat{W}_c$ and $\hat{W}_a$, respectively, such that $\hat{\delta} \to 0$ as $t \to \infty$. Intuitively, the state trajectory, x, needs to visit as many points in the operating domain as possible to approximate the optimal value function over an operating domain. The simulated experience is then used along with gained experience by the critic to approximate the value function.

3.3.1 Extension to Unknown Dynamics

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 esti-

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

mate as the initial guess and to employ a dynamic system identification technique capable of refining the initial guess based on input–output data.

To facilitate online system identification, let $f(x) = Y(x)\theta$ denote the linear parametrization of the function f, where $Y : \mathbb{R}^n \to \mathbb{R}^{n\times p}$ is the regression matrix and $\theta \in \mathbb{R}^p$ is the vector of constant unknown parameters. Let $\hat{\theta} \in \mathbb{R}^p$ be an estimate of the unknown parameter vector θ. The following development assumes that an adaptive system identifier that satisfies conditions detailed in Assumption 2 is available.

Assumption 2 A compact set $\Theta \subset \mathbb{R}^p$ such that $\theta \in \Theta$ is known a priori. The estimates $\hat{\theta} : \mathbb{R}_{\geq t_0} \to \mathbb{R}^p$ are updated based on a switched update law of the form

$$\dot{\hat{\theta}}(t) = f_{\theta s}\left(\hat{\theta}(t), t\right), \tag{26}$$

$\hat{\theta}(t_0) = \hat{\theta}_0 \in \Theta$, where $s \in \mathbb{N}$ denotes the switching index and $\{f_{\theta s} : \mathbb{R}^p \times \mathbb{R}_{\geq 0} \to \mathbb{R}^p\}_{s\in\mathbb{N}}$ denotes a family of continuously differentiable functions. The dynamics of the parameter estimation error $\tilde{\theta} : \mathbb{R}_{\geq t_0} \to \mathbb{R}^p$, defined as $\tilde{\theta}(t) \triangleq \theta - \hat{\theta}(t)$ can be expressed as $\dot{\tilde{\theta}}(t) = f_{\theta s}\left(\theta - \tilde{\theta}(t), t\right)$. Furthermore, there exists a continuously differentiable function $V_\theta : \mathbb{R}^p \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ that satisfies

$$\underline{v}_\theta\left(\left\|\tilde{\theta}\right\|\right) \leq V_\theta\left(\tilde{\theta}, t\right) \leq \overline{v}_\theta\left(\left\|\tilde{\theta}\right\|\right), \tag{27}$$

$$V_\theta'\left(\tilde{\theta}, t\right)\left(-f_{\theta s}\left(\theta - \tilde{\theta}, t\right)\right) + \frac{\partial V_\theta\left(\tilde{\theta}, t\right)}{\partial t} \leq -K\left\|\tilde{\theta}\right\|^2 + D\left\|\tilde{\theta}\right\|, \tag{28}$$

for all $s \in \mathbb{N}$, $t \in \mathbb{R}_{\geq t_0}$, and $\tilde{\theta} \in \mathbb{R}^p$, where $\underline{v}_\theta$, $\overline{v}_\theta : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ are class $\mathcal{K}$ functions, $K \in \mathbb{R}_{>0}$ is an adjustable parameter, and $D \in \mathbb{R}_{>0}$ is a positive constant.[5]

Using an estimate $\hat{\theta}$, the BE in (24) can be approximated by $\hat{\delta} : \mathbb{R}^{n+2L+p} \to \mathbb{R}$ as

$$\begin{aligned}\hat{\delta}\left(x, \hat{W}_c, \hat{W}_a, \hat{\theta}\right) &= x^T Q x + \hat{u}^T\left(x, \hat{W}_a\right) R \hat{u}\left(x, \hat{W}_a\right) \\ &+ \hat{V}'\left(x, \hat{W}_c\right)\left(Y(x)\hat{\theta} + g(x)\hat{u}\left(x, \hat{W}_a\right)\right).\end{aligned} \tag{29}$$

In the following, the approximate BE in (29) is used to obtain an approximate solution to the HJB equation in (13).

[5]The subsequent analysis in Sect. 3.5 indicates that when a system identifier that satisfies Assumption 2 is employed to facilitate online optimal control, the ratio $\frac{D}{K}$ needs to be sufficiently small to establish set-point regulation and convergence to optimality.

3.4 Actor and Critic Update Laws

A least squares update law for the critic weights is designed based on the stability analysis in [15] as

$$\dot{\hat{W}}_c(t) = -\eta_{c1}\Gamma\frac{\omega(t)}{\rho(t)}\hat{\delta}_t(t) - \frac{\eta_{c2}}{N}\Gamma\sum_{i=1}^{N}\frac{\omega_i(t)}{\rho_i(t)}\hat{\delta}_{ti}(t), \tag{30}$$

$$\dot{\Gamma}(t) = \left(\beta\Gamma(t) - \eta_{c1}\frac{\Gamma(t)\,\omega(t)\,\omega(t)^T\,\Gamma(t)}{\rho^2(t)}\right)\mathbf{1}_{\{\|\Gamma\|\le\overline{\Gamma}\}}, \tag{31}$$

where $\Gamma : \mathbb{R}_{\geq t_0} \to \mathbb{R}^{L\times L}$ is a time-varying least squares gain matrix, $\|\Gamma(t_0)\| \le \overline{\Gamma}$, $\omega(t) \triangleq \omega\left(x_i, \hat{W}_a(t)\right)$, $\omega_i(t) \triangleq \omega\left(x(t), \hat{W}_a(t)\right)$, $\rho(t) \triangleq 1+\nu\omega^T(t)\,\Gamma(t)\,\omega(t)$, and $\rho_i(t) \triangleq 1+\nu\omega_i^T(t)\,\Gamma(t)\,\omega_i(t)$. In addition, $\nu \in \mathbb{R}_{>0}$ is a constant normalization gain, $\overline{\Gamma} \in \mathbb{R}_{>0}$ is a saturation constant, $\beta \in \mathbb{R}_{>0}$ is a constant forgetting factor, and $\eta_{c1}, \eta_{c2} \in \mathbb{R}_{>0}$ are constant adaptation gains.

Motivate by the subsequent stability analysis, the actor weights are updated as

$$\begin{aligned}\dot{\hat{W}}_a(t) = &-\eta_{a1}\left(\hat{W}_a(t) - \hat{W}_c(t)\right) - \eta_{a2}\hat{W}_a(t) + \frac{\eta_{c1}G_\sigma^T(t)\,\hat{W}_a(t)\,\omega^T(t)}{4\rho(t)}\hat{W}_c(t)\\ &+\sum_{i=1}^{N}\frac{\eta_{c2}G_{\sigma i}^T\hat{W}_a(t)\,\omega_i^T(t)}{4N\rho_i(t)}\hat{W}_c(t),\end{aligned} \tag{32}$$

where $\eta_{a1}, \eta_{a2} \in \mathbb{R}_{>0}$ are constant adaptation gains and $G_{\sigma i} \triangleq G_\sigma(x_i)$. The update law in (31) ensures that the adaptation gain matrix is bounded such that

$$\underline{\Gamma} \le \|\Gamma(t)\| \le \overline{\Gamma},\ \forall t \in \mathbb{R}_{\geq t_0}. \tag{33}$$

Using the weight estimates $\hat{W}_a$, the controller for the system in (12) is designed as

$$u(t) = \hat{u}\left(x(t), \hat{W}_a(t)\right). \tag{34}$$

The following rank condition facilitates the subsequent stability analysis.

Assumption 3 There exists a finite set of fixed points $\{x_i \in \mathbb{R}^n \mid i = 1, \cdots, N\}$ such that $\forall t \in \mathbb{R}_{\geq t_0}$

$$0 < \underline{c} \triangleq \frac{1}{N}\left(\inf_{t\in\mathbb{R}_{\geq t_0}}\left(\lambda_{\min}\left\{\sum_{i=1}^{N}\frac{\omega_i(t)\,\omega_i^T(t)}{\rho_i(t)}\right\}\right)\right). \tag{35}$$

Compared to the typical PE condition, the condition in (35) can be verified online at each time t. Furthermore, the condition in (35) can be heuristically met by collecting redundant data (i.e., by selecting more points than the number of neurons by choosing $N \gg L$).

3.5 Stability Analysis

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 (13) from (24), an unmeasurable form of the instantaneous BE can be expressed as

$$\begin{aligned}\hat{\delta}_t &= -\omega^T \tilde{W}_c - W^T \sigma' Y \tilde{\theta} + \frac{1}{4} \tilde{W}_a^T G_\sigma \tilde{W}_a \\ &\quad + \frac{1}{4} G_\epsilon - \epsilon' f + \frac{1}{2} W^T \sigma' G \epsilon'^T,\end{aligned} \tag{36}$$

Similarly, the approximate BE evaluated at the sampled states $\{x_i \mid i = 1, \cdots, N\}$ can be expressed as

$$\hat{\delta}_{ti} = -\omega_i^T \tilde{W}_c + \frac{1}{4} \tilde{W}_a^T G_{\sigma i} \tilde{W}_a - W^T \sigma_i' Y_i \tilde{\theta} + \Delta_i, \tag{37}$$

where $\epsilon_i' = \epsilon'(x_i)$, $f_i = f(x_i)$, $G_i \triangleq g_i R^{-1} g_i^T \in \mathbb{R}^{n \times n}$, $G_{\epsilon i} \triangleq \epsilon_i' G_i \epsilon_i'^T \in \mathbb{R}$, and $\Delta_i \triangleq \frac{1}{2} W^T \sigma_i' G_i \epsilon_i'^T + \frac{1}{4} G_{\epsilon i} - \epsilon_i' f_i \in \mathbb{R}$ is a constant.

On any compact set $\chi \subset \mathbb{R}^n$ the function Y is Lipschitz continuous, and hence, there exists a positive constant $L_Y \in \mathbb{R}$ such that[6]

$$\|Y\| \leq L_Y \|x\|, \forall x \in \chi. \tag{38}$$

Using (33), the normalized regressor $\frac{\omega}{\rho}$ can be bounded as

$$\sup_{t \in \mathbb{R}_{\geq t_0}} \left\| \frac{\omega}{\rho} \right\| \leq \frac{1}{2\sqrt{\nu \underline{\Gamma}}}. \tag{39}$$

For brevity of notation, the following positive constants are defined:

$$\vartheta_1 \triangleq \frac{\eta_{c1} L_Y \|\theta\| \bar{\epsilon}'}{4\sqrt{\nu \underline{\Gamma}}}, \quad \vartheta_2 \triangleq \sum_{i=1}^{N} \left(\frac{\eta_{c2} \left\| \sigma_i' Y_i \right\| \overline{W}}{4N\sqrt{\nu \underline{\Gamma}}} \right),$$

$$\vartheta_3 \triangleq \frac{L_Y \eta_{c1} \overline{W \|\sigma'\|}}{4\sqrt{\nu \underline{\Gamma}}}, \quad \vartheta_4 \triangleq \overline{\left\| \frac{1}{4} G_\epsilon \right\|},$$

[6] The Lipschitz property is exploited here for clarity of exposition. The bound in (38) can be easily generalized to $\|Y(x)\| \leq L_Y(\|x\|) \|x\|$, where $L_Y : \mathbb{R} \to \mathbb{R}$ is a positive, non-decreasing function.

$$\vartheta_5 \triangleq \frac{\eta_{c1}\overline{\left\|2W^T\sigma' G\epsilon'^T + G_\epsilon\right\|}}{8\sqrt{\nu\underline{\Gamma}}} + \left\|\sum_{i=1}^{N}\frac{\eta_{c2}\omega_i\Delta_i}{N\rho_i}\right\|,$$

$$\vartheta_6 \triangleq \overline{\left\|\frac{1}{2}W^T G_\sigma + \frac{1}{2}\epsilon' G^T\sigma'^T\right\|} + \vartheta_7\overline{W}^2 + \eta_{a2}\overline{W},$$

$$\vartheta_7 \triangleq \frac{\eta_{c1}\overline{\|G_\sigma\|}}{8\sqrt{\nu\underline{\Gamma}}} + \sum_{i=1}^{N}\left(\frac{\eta_{c2}\left\|G_{\sigma i}\right\|}{8N\sqrt{\nu\underline{\Gamma}}}\right), \quad \underline{q} \triangleq \lambda_{\min}\{Q\},$$

$$v_l = \frac{1}{2}\min\left(\frac{\underline{q}}{2}, \frac{\eta_{c2}\underline{c}}{3}, \frac{\eta_{a1}+2\eta_{a2}}{6}, \frac{K}{4}\right),$$

$$\iota = \frac{3\vartheta_5^2}{4\eta_{c2}\underline{c}} + \frac{3\vartheta_6^2}{2\left(\eta_{a1}+2\eta_{a2}\right)} + \frac{D^2}{2K} + \vartheta_4, \tag{40}$$

where $\overline{(\cdot)} \triangleq \sup_{x\in\chi}(\cdot) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. Let $Z : \mathbb{R}_{\geq t_0} \to \mathbb{R}^{n+2L+p}$ be defined as

$$Z(t) \triangleq \left[x^T(t), \tilde{W}_c^T(t), \tilde{W}_a^T(t), \tilde{\theta}^T(t)\right]^T, \tag{41}$$

where $x(\cdot)$, $\tilde{W}_c(\cdot)$, $\tilde{W}_a(\cdot)$, and $\tilde{\theta}(\cdot)$ denote the solutions of the differential equations in (12), (30), and (32), respectively, with appropriate initial conditions. The sufficient conditions for ultimate boundedness of $Z(\cdot)$ are derived based on the subsequent stability analysis as

$$\begin{aligned}\frac{\eta_{a1}+2\eta_{a2}}{6} &> \vartheta_7\overline{W}\left(\frac{2\zeta_2+1}{2\zeta_2}\right),\\ \frac{K}{4} &> \frac{\vartheta_2+\zeta_1\zeta_3\vartheta_3\overline{Z}}{\zeta_1},\\ \frac{\eta_{c2}}{3} &> \frac{\zeta_2\vartheta_7\overline{W}+\eta_{a1}+2\left(\vartheta_1+\zeta_1\vartheta_2+\left(\vartheta_3/\zeta_3\right)\overline{Z}\right)}{2\underline{c}},\\ \frac{\underline{q}}{2} &> \vartheta_1,\end{aligned} \tag{42}$$

where $\overline{Z} \triangleq \underline{v}^{-1}\left(\overline{v}\left(\max\left(\|Z(t_0)\|, \sqrt{\frac{\iota}{v_l}}\right)\right)\right)$, $\zeta_1, \zeta_2, \zeta_3 \in \mathbb{R}$ are known positive adjustable constants, and $\underline{v}$ and $\overline{v}$ are subsequently defined class $\mathcal{K}$ functions. The Lipschitz constants in (38) and the NN function approximation errors depend on the underlying compact set; hence, given a bound on the initial condition $Z(t_0)$ for the concatenated state $Z(\cdot)$, a compact set that contains the concatenated state trajectory needs to be established before adaptation gains satisfying the conditions in (42) can be selected. Based on the subsequent stability analysis, an algorithm to compute the required compact set, denoted by $\mathcal{Z} \subset \mathbb{R}^{2n+2L+p}$, is developed in [15]. Since the constants ι and υ_l depend on L_Y only through the products $L_Y\overline{\epsilon}$ and $\frac{L_Y}{\zeta_3}$, proper gain selection ensures that

$$\sqrt{\frac{\iota}{\upsilon_l}} \leq \frac{1}{2} diam\left(\mathcal{Z}\right), \tag{43}$$

where $diam\left(\mathcal{Z}\right)$ denotes the diameter of the set $\mathcal{Z}$ defined as $diam\left(\mathcal{Z}\right) \triangleq \sup\left\{\|x-y\| \mid x, y \in \mathcal{Z}\right\}$. The main result of this section can now be stated as follows.

Theorem 4 *Provided Assumptions 1–3 hold and gains $\underline{q}$, η_{c2}, η_{a2}, and K are sufficiently large, the controller in (34) along with the adaptive update laws in (30) and (32) ensure that the $x\left(\cdot\right)$, $\tilde{W}_c\left(\cdot\right)$, $\tilde{W}_a\left(\cdot\right)$, and $\tilde{\theta}\left(\cdot\right)$ are uniformly ultimately bounded (UUB).*

Proof For brevity, a sketch of the proof is provided, the detailed proof can be seen in [15]. Consider a candidate Lyapunov function candidate $V_L : \mathbb{R}^{n+2L+p} \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$ defined as

$$V_L\left(Z, t\right) \triangleq V^*\left(x\right) + \frac{1}{2}\tilde{W}_c^T \Gamma^{-1}\left(t\right)\tilde{W}_c + \frac{1}{2}\tilde{W}_a^T\tilde{W}_a + V_\theta\left(\tilde{\theta}, t\right). \tag{44}$$

Since the optimal value function is positive definite, (33) and [40, Lemma 4.3] can be used to show that the candidate Lyapunov function satisfies the following bounds

$$\underline{\upsilon_l}\left(\|Z\|\right) \leq V_L\left(Z, t\right) \leq \overline{\upsilon_l}\left(\|Z\|\right), \tag{45}$$

for all $t \in \mathbb{R}_{\geq t_0}$ and for all $Z \in \mathbb{R}^{n+2L+p}$. In (45), $\underline{\upsilon_l}, \overline{\upsilon_l} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ are class $\mathcal{K}$ functions. Taking the time -derivative of (44) along the system trajectory, substituting (30)–(32) along with (36) and (37). Bounding and enforcing (42) produces the stated result.

3.6 Summary

In this section, the PE condition is replaced by a set of rank conditions that can be verified online using current and recorded observations. 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. While the result in Sect. 3 demonstrates online approximate optimal regulation using BE extrapolation, it tends to be computationally inefficient since it performs value function approximations across the entire state-space. In Sect. 4, a computationally efficient method is discussed by performing local value function approximations.

4 Local (State-Following) Model-Based Reinforcement Learning

Sufficiently accurate approximation of the value function over a sufficiently large neighborhood often requires a large number of basis functions, and hence, introduces a large number of unknown parameters. One way to achieve accurate function approximation with fewer unknown parameters is to use prior knowledge about the system to determine the basis functions. However, generally, prior knowledge of the features of the optimal value function is not available; hence, a large number of generic basis functions is often the only feasible option.

Fast approximation of the value function over a large neighborhood requires sufficiently rich data to be available for learning. In traditional ADP methods such as [7, 9, 38], richness of data manifests itself as the amount of excitation in the system. In experience replay-based techniques such as [24, 41–43], richness of data is quantified by eigenvalues of a recorded history stack. In R-MBRL techniques such as [44–46], richness of data corresponds to the eigenvalues of a learning matrix. As the dimension of the system and the number of basis functions increases, the richer data is required to achieve learning. In experience replay-based ADP methods and in R-MBRL, the demand for richer data causes exponential growth in the required data storage. Hence, implementations of traditional ADP techniques such as [1–11, 38] and data-driven ADP techniques such as [24, 44–48] in high-dimensional systems are scarcely found in the literature.

This section presents a MBRL technique with a lower computational cost than current data-driven ADP techniques. Motivated by the fact that the computational effort required to implement ADP and the data-richness required to achieve convergence both decrease with decreasing number of basis functions, this technique reduces the number of basis functions used for value function approximation.

A key contribution of [18, 49] is the observation that online implementation of an ADP-based approximate optimal controller does not require an estimate of the optimal value function over the entire domain of operation of the system. Instead, only an estimate of the value function gradient at the current state is required. Since it is reasonable to postulate that approximation of the value function over a local domain would require fewer basis functions than approximation over the entire domain of operation, this section focuses on the reduction of the size of approximation domain. Such a reduction is achieved via the selection of basis functions that travel with the system state (referred to as state-following (StaF) kernels).

Unlike traditional value function approximation, where the unknown parameters are constants, the unknown parameters corresponding to the StaF kernels are functions of the system state. To facilitate the proof of continuous differentiability, the StaF kernels are selected from a reproducing kernel Hilbert space (RKHS). Other function approximation methods, such as radial basis functions, sigmoids, higher order NNs, support vector machines, etc., can potentially be utilized in a state-following manner to achieve similar results provided continuous differentiability of the ideal weights can be established.

4.1 StaF Kernel Functions

In this section, Theorem 5 motivates the use of StaF kernels for model-based RL, and Theorem 6 facilitates implementation of gradient-based update laws to learn the time-varying ideal weights in real-time.

Theorem 5 *Let $\epsilon, r > 0$ and let p denote a polynomial that approximates $\overline{V}^*$ within an error ϵ over $B_r(x)$. Let $N(r, x, \epsilon)$ denote the degree of p. Let $k(y, x) = e^{y^T x}$ be the exponential kernel function, which corresponds to a universal RKHS. Then, for each $x \in \chi$, there exists a finite number of centers, $c_1, c_2, ..., c_{M(r,x,\epsilon)} \in B_r(x)$ and weights $w_1, w_2, ..., w_{M(r,x,\epsilon)}$ such that*

$$\left\| \overline{V}^*(y) - \sum_{i=1}^{M(r,x,\epsilon)} w_i e^{y^T c_i} \right\|_{B_r(x),\infty} < \epsilon,$$

where $M(r, x, \epsilon) < \binom{n+N(r,x,\epsilon)+S(r,x,\epsilon)}{N(r,x,\epsilon)+S(r,x,\epsilon)}$, asymptotically, for some $S(r, x, \epsilon) \in \mathbb{N}$. Moreover, r, $N(r, x, \epsilon)$ and $S(r, x, \epsilon)$ can be bounded uniformly over χ for any fixed ϵ [18].[7]

The Weierstrass theorem indicates that as r decreases, the degree $N(r, x, \epsilon)$ of the polynomial needed to achieve the same error ϵ over $B_r(x)$ decreases [50]. Hence, by Theorem 5, approximation of a function over a smaller domain requires a smaller number of exponential kernels. Furthermore, provided the region of interest is small enough, the number of kernels required to approximate continuous functions with arbitrary accuracy can be reduced to $\binom{n+2}{2}$.

In the StaF approach, the centers are selected to follow the current state x, i.e., the locations of the centers are defined as a function of the system state. Since the system state evolves in time, the ideal weights are not constant. To approximate the ideal weights using gradient-based algorithms, it is essential that the weights change smoothly with respect to the system state. The following theorem allows the use of gradient-based update laws to determine the time-varying ideal weights of the value function.

Theorem 6 *Let the kernel function k be such that the functions $k(\cdot, x)$ are $l-$times continuously differentiable for all $x \in \chi$. Let $C \triangleq [c_1, c_2, ..., c_L]^T$ be a set of distinct centers such that $c_i \in B_r(x)$, $\forall i = 1, \cdots, L$, be an ordered collection of L distinct centers with associated ideal weights*

$$W_{H_{x,r}}(C) = \arg\min_{a \in R^M} \left\| \sum_{i=1}^{M} a_i k(\cdot, c_i) - V(\cdot) \right\|_{H_{x,r}}. \tag{46}$$

Then, the function $W_{H_{x,r}}$ is $l-$times continuously differentiable with respect to each component of C [18].

[7] The notation $\binom{a}{b}$ denotes the combinatorial operation "a choose b".

490511_1_En_5_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:19/11/2020 Pages: xxx Layout: T1-Standard

4.2 Local Value Function Approximation

Similar to Sect. 3, an approximate solution to the HJB equation is sought. The optimal value function V^* is approximated using a parametric estimate. The expression for the optimal policy in (20) indicates that, to compute the optimal action when the system is at any given state x, one only needs to evaluate the gradient $V^{*\prime}$ at x. Hence, to compute the optimal policy at x, one only needs to approximate the value function over a small neighborhood around the current state, x. As established in Theorem 5, the number of basis functions required to approximate the value function decreases if the approximation space decreases (with respect to the ordering induced by set containment). The aim is to obtain a uniform approximation of the value function over a small neighborhood around the system state.

To facilitate the development, let $\chi \subset \mathbb{R}^n$ be compact and let x be in the interior of χ. Then, for all $\epsilon > 0$, there exists a function $\overline{V}^* \in H_{x,r}$ such that $\sup_{y \in B_r(x)} \left| V^*(y) - \overline{V}^*(y) \right| < \epsilon$, where $H_{x^o,r}$ is a restriction of a universal RKHS, H, introduced in Sect. 4.1, to $B_r(x)$. In the developed StaF-based method, a small compact set $B_r(x)$ around the current state x is selected for value function approximation by selecting the centers $C \in B_r(x)$ such that $C = c(x)$ for some continuously differentiable function $c : \chi \to \chi^L$. Using StaF kernels centered at a point x, the value function can be represented as

$$V^*(y) = W(x)^T \sigma(y, c(x)) + \epsilon(x, y).$$

Since the centers of the kernel functions change as the system state changes, the ideal weights also change as the system state changes. The state-dependent nature of the ideal weights differentiates this approach from aforementioned ADP methods in the sense that the stability analysis needs to account for changing ideal weights. Based on Theorem 6, the ideal weight function $W : \chi \to \mathbb{R}^L$ defined as $W(x) \triangleq W_{H_{x,r}}(c(x))$, where $W_{H_{x,r}}$ was introduced in (46), is continuously differentiable, provided the functions σ and c are continuously differentiable.

The approximate value function $\hat{V} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}$ and the approximate policy $\hat{u} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}^m$, evaluated at a point $y \in B_r(x)$, using StaF kernels centered at x, can then be expressed as

$$\hat{V}\left(y, x, \hat{W}_c\right) \triangleq \hat{W}_c^T \sigma(y, c(x)), \tag{47}$$

$$\hat{u}\left(y, x, \hat{W}_a\right) \triangleq -\frac{1}{2} R^{-1} g^T(y) \sigma'(y, c(x))^T \hat{W}_a. \tag{48}$$

490511_1_En_5_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:19/11/2020 Pages: xxx Layout: T1-Standard

4.3 Actor and Critic Update Laws

In this section, the BE, weight update laws, and ω, are redefined to clarify the distinction that the BE is calculated from time-varying points in the neighborhood of the current trajectory. The critic uses the BEs

$$\delta_t\left(t\right) \triangleq \delta\left(x\left(t\right), x\left(t\right), \hat{W}_c\left(t\right), \hat{W}_a\left(t\right)\right), \tag{49}$$

and

$$\delta_{ti}\left(t\right) = \delta\left(x_i\left(x\left(t\right), t\right), x\left(t\right), \hat{W}_c\left(t\right), \hat{W}_a\left(t\right)\right). \tag{50}$$

to improve the StaF-based estimate $\hat{W}_c\left(t\right)$ using the recursive least squares-based update law

$$\dot{\hat{W}}_c\left(t\right) = -k_{c1}\Gamma\left(t\right)\frac{\omega\left(t\right)}{\rho\left(t\right)}\delta_t\left(t\right) - \frac{k_{c2}}{N}\Gamma\left(t\right)\sum_{i=1}^{N}\frac{\omega_i\left(t\right)}{\rho_i\left(t\right)}\delta_{ti}\left(t\right), \tag{51}$$

where $\rho_i\left(t\right) \triangleq \sqrt{1+\gamma_1\omega_i^T\left(t\right)\omega_i\left(t\right)}$, $\rho\left(t\right) \triangleq \sqrt{1+\gamma_1\omega^T\left(t\right)\omega\left(t\right)}$, $k_{c1}, k_{c2}, \gamma_1 \in \mathbb{R}_{>0}$ are constant learning gains,

$$\begin{aligned}\omega\left(t\right) \triangleq\ & \sigma'\left(x\left(t\right), c\left(x\left(t\right)\right)\right) f\left(x\left(t\right)\right)\\ & + \sigma'\left(x\left(t\right), c\left(x\left(t\right)\right)\right) g\left(x\left(t\right)\right)\hat{u}\left(x\left(t\right), x\left(t\right), \hat{W}_a\left(t\right)\right),\end{aligned}$$

and

$$\begin{aligned}\omega_i\left(t\right) \triangleq\ & \sigma'\left(x_i\left(x\left(t\right)\right), c\left(x\left(t\right)\right)\right) f\left(x_i\left(x\left(t\right), t\right)\right)\\ & + \sigma'\left(x_i\left(x\left(t\right)\right), c\left(x\left(t\right)\right)\right) g\left(x_i\left(x\left(t\right), t\right)\right)\hat{u}\left(x_i\left(x\left(t\right), t\right), x\left(t\right), \hat{W}_a\left(t\right)\right)\end{aligned}$$

In (51), $\Gamma\left(t\right)$ denotes the least-square learning gain matrix updated according to

$$\begin{aligned}\dot{\Gamma}\left(t\right) &= \beta\Gamma\left(t\right) - k_{c1}\Gamma\left(t\right)\frac{\omega\left(t\right)\omega^T\left(t\right)}{\rho^2\left(t\right)}\Gamma\left(t\right)\\ &\quad - \frac{k_{c2}}{N}\Gamma\left(t\right)\sum_{i=1}^{N}\frac{\omega_i\left(t\right)\omega_i^T\left(t\right)}{\rho_i^2\left(t\right)}\Gamma\left(t\right),\\ \Gamma\left(t_0\right) &= \Gamma_0,\end{aligned} \tag{52}$$

where $\beta \in \mathbb{R}_{>0}$ is a constant forgetting factor. Motivated by a Lyapunov-based stability analysis, the update law for the actor is designed as

$$\begin{aligned}\dot{\hat{W}}_a(t) = &-k_{a1}\left(\hat{W}_a(t) - \hat{W}_c(t)\right) - k_{a2}\hat{W}_a(t)\\ &+ \frac{k_{c1}G_\sigma^T(t)\hat{W}_a(t)\omega(t)^T}{4\rho(t)}\hat{W}_c(t)\\ &+ \sum_{i=1}^{N}\frac{k_{c2}G_{\sigma i}^T(t)\hat{W}_a(t)\omega_i^T(t)}{4N\rho_i(t)}\hat{W}_c(t),\end{aligned} \tag{53}$$

where $k_{a1}, k_{a2} \in \mathbb{R}_{>0}$ are learning gains,

$$\begin{aligned}G_\sigma(t) \triangleq\, &\sigma'(x(t), c(x(t)))\, g(x(t))\, R^{-1} g^T(x(t))\\ &\cdot \sigma'^T(x(t), c(x(t))),\end{aligned}$$

and

$$\begin{aligned}G_{\sigma i}(t) \triangleq\, &\sigma'(x_i(x(t), t), c(x(t)))\, g(x_i(x(t), t))\\ &\cdot R^{-1} g^T(x_i(x(t), t))\, \sigma'^T(x_i(x(t), t), c(x(t))).\end{aligned}$$

4.4 Analysis

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 $\overline{\|h\|} \triangleq \sup_{\xi\in\chi}\|h(\xi)\|$, for some continuous function $h : \mathbb{R}^n \to \mathbb{R}^k$. To facilitate the subsequent stability analysis, the BEs in are expressed in terms of the weight estimation errors $\tilde{W}_c$ and $\tilde{W}_a$, defined in Sect. 3, as

$$\begin{aligned}\delta_t &= -\omega^T\tilde{W}_c + \frac{1}{4}\tilde{W}_a G_\sigma \tilde{W}_a + \Delta(x),\\ \delta_{ti} &= -\omega_i^T\tilde{W}_c + \frac{1}{4}\tilde{W}_a^T G_{\sigma i}\tilde{W}_a + \Delta_i(x),\end{aligned} \tag{54}$$

where the functions $\Delta, \Delta_i : \mathbb{R}^n \to \mathbb{R}$ are uniformly bounded over χ such that the bounds $\overline{\|\Delta\|}$ and $\overline{\|\Delta_i\|}$ decrease with decreasing $\overline{\|\varepsilon^\nabla\|}$ and $\overline{\|\nabla W\|}$. To facilitate learning, the system states x and the selected functions x_i are assumed to satisfy the following.

Assumption 4 There exist constants $T \in \mathbb{R}_{>0}$ and $\underline{c}_1, \underline{c}_2, \underline{c}_3 \in \mathbb{R}_{\geq 0}$, such that

$$\underline{c}_1 I_L \leq \int_t^{t+T} \left(\frac{\omega(\tau)\,\omega^T(\tau)}{\rho^2(\tau)} \right) d\tau, \ \forall t \in \mathbb{R}_{\geq t_0},$$

$$\underline{c}_2 I_L \leq \inf_{t \in \mathbb{R}_{\geq t_0}} \left(\frac{1}{N} \sum_{i=1}^{N} \frac{\omega_i(t)\,\omega_i^T(t)}{\rho_i^2(t)} \right),$$

$$\underline{c}_3 I_L \leq \frac{1}{N} \int_t^{t+T} \left(\sum_{i=1}^{N} \frac{\omega_i(\tau)\,\omega_i^T(\tau)}{\rho_i^2(\tau)} \right) d\tau, \ \forall t \in \mathbb{R}_{\geq t_0},$$

where at least one of the constants $\underline{c}_1$, $\underline{c}_2$, and $\underline{c}_3$ is strictly positive.

Assumption 4 only requires either the regressor ω or the regressor ω_i to be PE. 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.

Selection of a single time-varying BE extrapolation function results in virtual excitation. That is, instead of using input–output data from a persistently excited system, the dynamic model is used to simulate PE to facilitate parameter convergence.

Lemma 1 *Provided Assumption 4 holds and* $\lambda_{\min}\left\{\Gamma_0^{-1}\right\} > 0$*, the update law in (52) ensures that the least squares gain matrix satisfies*

$$\underline{\Gamma} I_L \leq \Gamma(t) \leq \overline{\Gamma} I_L, \tag{55}$$

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, $\underline{\Gamma} > 0$ *[18, 26].*

4.5 Stability Analysis

To facilitate the analysis, let $\underline{c} \in \mathbb{R}_{>0}$ be a constant defined as

$$\underline{c} \triangleq \frac{\beta}{2\overline{\Gamma} k_{c2}} + \frac{\underline{c}_2}{2}, \tag{56}$$

and let $\iota \in \mathbb{R}_{>0}$ be a constant defined as

$$\begin{aligned}
\iota \triangleq & \frac{3\left(\frac{(k_{c1}+k_{c2})\overline{\|\Delta\|}}{\sqrt{\upsilon}} + \frac{\overline{\|\nabla W f\|}}{\underline{\Gamma}} + \frac{\overline{\|\Gamma^{-1} G_{W\sigma} W\|}}{2}\right)^2}{4k_{c2}\underline{c}} \\
& + \frac{1}{(k_{a1}+k_{a2})}\left(\frac{\overline{\|G_{W\sigma} W\|} + \overline{\|G_{V\sigma}\|}}{2} + k_{a2}\overline{\|W\|}\right. \\
& \left. + \overline{\|\nabla W f\|} + \frac{(k_{c1}+k_{c2})\overline{\|G_\sigma\|\|W\|}^2}{4\sqrt{\upsilon}}\right)^2 \\
& + \frac{1}{2}\overline{\|G_{VW}\sigma\|} + \frac{1}{2}\overline{\|G_{V\varepsilon}\|},
\end{aligned}$$

where $G_{W\sigma} \triangleq \nabla W G \sigma'^T$, $G_{V\sigma} \triangleq V^{*\prime} G \sigma'^T$, $G_{VW} \triangleq V^{*\prime} G \nabla W^T$, and $G_{V\epsilon} \triangleq V^{*\prime} G \epsilon'^T$. Let $\upsilon_l : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a class $\mathcal{K}$ function such that

$$\upsilon_l(\|Z\|) \leq \frac{Q(x)}{2} + \frac{k_{c2}\underline{c}}{6}\left\|\tilde{W}_c\right\|^2 + \frac{(k_{a1}+k_{a2})}{8}\left\|\tilde{W}_a\right\|^2.$$

The sufficient conditions for Lyapunov-based stability are given by

$$\frac{k_{c2}\underline{c}}{3} \geq \frac{\left(\frac{\overline{\|G_{W\sigma}\|}}{2\underline{\Gamma}} + \frac{(k_{c1}+k_{c2})\overline{\|W^T G_\sigma\|}}{4\sqrt{\upsilon}} + k_{a1}\right)^2}{(k_{a1}+k_{a2})}, \tag{57}$$

$$\frac{(k_{a1}+k_{a2})}{4} \geq \left(\frac{\overline{\|G_{W\sigma}\|}}{2} + \frac{(k_{c1}+k_{c2})\overline{\|W\|\|G_\sigma\|}}{4\sqrt{\upsilon}}\right), \tag{58}$$

$$\upsilon_l^{-1}(\iota) < \overline{\upsilon_l}^{-1}\left(\underline{\upsilon_l}(\zeta)\right). \tag{59}$$

The sufficient condition in (57) can be satisfied provided the points for BE extrapolation are selected such that the minimum eigenvalue $\underline{c}$, introduced in (56) is large enough. The sufficient condition in (58) can be satisfied without affecting (57) by increasing the gain k_{a2}. The sufficient condition in (59) can be satisfied provided $\underline{c}$, k_{a2}, and the state penalty $Q(x)$ are selected to be sufficiently large and the StaF kernels for value function approximation are selected such that $\overline{\|\nabla W\|}$, $\overline{\|\varepsilon\|}$, and

$\overline{\|\nabla\varepsilon\|}$ are sufficiently small.[8] To improve computational efficiency, the size of the domain around the current state where the StaF kernels provide good approximation of the value function is desired to be small. Smaller approximation domain results in almost identical extrapolated points, which in turn, results in smaller $\underline{c}$. Hence, the approximation domain cannot be selected to be arbitrarily small and needs to be large enough to meet the sufficient conditions in (57)–(59).

Theorem 7 *Provided Assumption 4 holds and the sufficient gain conditions in (57)–(59) are satisfied, the controller $u(t)$ and the update laws in (51)–(53) ensure that the state x and the weight estimation errors $\tilde{W}_c$ and $\tilde{W}_a$ are UUB.*

Proof The proof follows from Theorem 4, see [18] for a detailed analysis.

4.6 Summary

In this section, an infinite horizon optimal control problem is solved using an approximation methodology called the StaF kernel method. Motivated by the fact that a smaller number of basis functions is required to approximate functions on smaller domains, the StaF kernel method aims to maintain a good approximation of the value function over a small neighborhood of the current state. Computational efficiency of model-based RL is improved by allowing selection of fewer time-varying extrapolation trajectories instead of a large number of autonomous extrapolation functions.

Methods to solve infinite horizon optimal control problems online aim to approximate the value function over the entire operating domain. Since the approximate optimal policy is completely determined by the value function estimate, solutions generate policies that are valid over the entire state space but at a high computational cost. Since the StaF kernel method aims at maintaining local approximation of the value function around the current system state, the StaF kernel method lacks memory, in the sense that the information about the ideal weights over a region of interest is lost when the state leaves the region of interest. Thus, unlike aforementioned techniques, the StaF method trades global optimality for computational efficiency to generate a policy that is near-optimal only over a small neighborhood of the origin. A memory-based modification to the StaF technique that retains and reuses past information is a subject for the following section.

The technique developed in this section can be extended to a class of trajectory tracking problems in the presence of uncertainties in the system drift dynamics by using a concurrent learning-based adaptive system identifier (cf., [15, 18, 26, 45]).

[8] Similar to NN-based approximation methods such as [1–8], the function approximation error, ε, is unknown, and in general, infeasible to compute for a given function, since the ideal NN weights are unknown. Since a bound on ε is unavailable, the gain conditions in (57)–(59) cannot be formally verified. However, they can be met using trial and error by increasing the gain k_{a2}, the number of StaF basis functions, and $\underline{c}$, by selecting more points to extrapolate the BE.

5 Combining Regional and Local State-Following Approximations

Reduction in the number of unknown parameters motivates the use of StaF basis functions as described in the previous section (c.f., [51]), 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 [52], a 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. The ability of R-MBRL (c.f., [15]) to approximate the value function over a predefined region and the computational efficiency of the StaF method [18] in approximating the value function locally along the state trajectory motivates the additional development. Instead of generating an approximation of the value function over the entire operating region, which is computationally expensive, the operating domain can be 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 injects discontinuities to the system and result in a non-smooth value function which would introduce discontinuities in the control signal. To overcome this challenge, a state-varying convex combination of the two approximation methods can be 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 this region, R-MBRL regulates the state to the origin. The developed result can be generalized to allow for the use of any R-MBRL method. This strategy is motivated by the observation that in many applications such as station keeping of marine craft, like in [53], 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 BE. To provide insights into how to combine StaF and R-MBRL while also preserving stability, see [52].

6 Reinforcement Learning with Sparse Bellman Error Extrapolation

Motivated by additional computational efficiency, sparsification techniques are motivated to collectively perform BE extrapolation in segmented parts of the operating domain. Sparsification techniques enable local approximation across the segments, which allows characterization of regions with significantly varying dynamics or unknown uncertainties.

Sparse neural networks (SNNs), like conventional NNs, are a tool to facilitate learning in uncertain systems (cf., [54–62]). SNNs have been used to reduce the computational complexity in NNs by decreasing the number of active neurons; hence, reducing the number of computations overall. Sparse adaptive controllers have been developed to update a small number of neurons at certain points in the state space in works such as [60]. Sparsification encourages local learning through intelligent segmentation [56], and encourages learning without relying on a high adaptive learning rate [62]. In practice, high learning rates can cause oscillations or instability due to unmodeled dynamics in the control bandwidth [62]. SNNs create a framework for switching and segmentation as well as computational benefits due to the small number of active neurons. Sparsification techniques enable local approximation across the segments, which characterizes regions with significantly varying dynamics or unknown uncertainties.

In [61], a method is developed to better estimate the value function across the entire state space by using a set of sparse off-policy trajectories, which are used to calculate extrapolated BEs. The set of off-policy trajectories will be determined by the location in the state space of the system. Hence, sets of input–output data pairs corresponding to each segment of the operating domain are developed and used in the actor and critic update laws. Compared to results such as [15, 63, 64], this technique does not perform BE extrapolation over the entire operating domain at each time instance. Instead, the operating domain is divided into segments where a certain set of trajectories, and, hence, sets of extrapolated BEs, are active when the state enters the corresponding segment. SNNs are used within each segment to extrapolate the BE due to their small amount of active neurons, whose activity can be switched on or off based on the active segment, to make BE extrapolation more computationally efficient. Using the increased computational efficiency of SNNs and segmentation to extrapolate the BE, the BE can be estimated across the entire state space.

7 Conclusion

This chapter discussed mixed density RL-based approximate optimal control methods applied to deterministic systems. Implementations of model-based RL to solve approximate optimal regulation problems online using different value function approximation and BE extrapolation techniques were discussed. While the mixed density methods presented in this chapter shed some light on potential solutions, methods must be developed and refined to address future needs.

In Sect. 2, the infinite horizon optimal control problem is introduced along with conditions that establish the optimal control policy. It is shown that the value function is the optimal cost-to-go and satisfies the HJB equation.

In Sect. 3, the R-MBRL method is presented where unknown weights in the value function are adjusted based on least squares minimization of the BE evaluated at any number of user-selected arbitrary trajectories in the state space. Since the BE can be evaluated at any desired point in the state space, sufficient exploration is achieved by selecting points distributed over the system's operating domain. R-MBRL utilizes BE extrapolation over a large region of the state space but is computationally complex. The strategies in Sects. 4–6 address the computational constraints of this method. Future work includes extending this result to hybrid systems.

In Sect. 4, the StaF-RL method is presented where the computational complexity of R-MBRL problems is reduced by estimating the optimal value function within a local domain around the state. Future work will focus on extending this method to nonaffine systems [18].

In Sect. 5, a strategy that uses R-MBRL and StaF-RL together to approximate the value function is described. This technique eliminates the need to perform BE extrapolation over a large region of the state space, as in R-MBRL, and the inability for the StaF method to develop a global estimate of the value function. Future work includes investigating the rate at which the optimal value function is learned and how it changes based on the size of the R-MBRL region [52].

In Sect. 6, a strategy is described to overcome the computational cost of R-MBRL by using a set of sparse off-policy trajectories, which are used to calculate extrapolated BEs. Furthermore, the state space is divided into a user-selected number of segments. SNNs could then be used within each segment to extrapolate the BE due to their small amount of active neurons, whose activity can be switched on or off based on the active segment, to make BE extrapolation more computationally efficient. Future work will study the accuracy of using SNNs as opposed to conventional NNs, quantity the computational savings of using SNNs, and generating a Zeno-free switched system (i.e., exclude Zeno behavior with respect to switching).

References

1. Doya, K.: Reinforcement learning in continuous time and space. Neural Comput. **12**(1), 219–245 (2000)
2. Padhi, R., Unnikrishnan, N., Wang, X., Balakrishnan, S.: A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems. Neural Netw. **19**(10), 1648–1660 (2006)
3. Al-Tamimi, A., Lewis, F.L., Abu-Khalaf, M.: Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof. IEEE Trans. Syst. Man Cybern. Part B Cybern. **38**, 943–949 (2008)

4. Lewis, F.L., Vrabie, D.: Reinforcement learning and adaptive dynamic programming for feedback control. IEEE Circuits Syst. Mag. **9**(3), 32–50 (2009)
5. Dierks, T., Thumati, B., Jagannathan, S.: Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence. Neural Netw. **22**(5–6), 851–860 (2009)
6. Mehta, P., Meyn, S.: Q-learning and pontryagin's minimum principle. In: Proceedings of the IEEE Conference on Decision and Control, pp. 3598–3605
7. Vamvoudakis, K.G., Lewis, F.L.: Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. Automatica **46**(5), 878–888 (2010)
8. Zhang, H., Cui, L., Zhang, X., Luo, Y.: Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method. IEEE Trans. Neural Netw. **22**(12), 2226–2236 (2011)
9. Bhasin, S., Kamalapurkar, R., Johnson, M., Vamvoudakis, K.G., Lewis, F.L., Dixon, W.E.: A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems. Automatica **49**(1), 89–92 (2013)
10. Zhang, H., Cui, L., Luo, Y.: Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network adp. IEEE Trans. Cybern. **43**(1), 206–216 (2013)
11. Zhang, H., Liu, D., Luo, Y., Wang, D.: Adaptive Dynamic Programming for Control Algorithms and Stability, ser. Communications and Control Engineering. Springer, London (2013)
12. Kaelbling, L., Littman, M., Moore, A.: Reinforcement learning: a survey. J. Artif. Intell. Res. **4**, 237–285 (1996)
13. Vrabie, D.: Online adaptive optimal control for continuous-time systems, Ph.D. dissertation, University of Texas at Arlington (2010)
14. Vamvoudakis, K.G., Vrabie, D., Lewis, F.L.: Online adaptive algorithm for optimal control with integral reinforcement learning. Int. J. Robust Nonlinear Control **24**(17), 2686–2710 (2014)
15. Kamalapurkar, R., Walters, P., Dixon, W.E.: Model-based reinforcement learning for approximate optimal regulation. Automatica **64**, 94–104 (2016)
16. He, P., Jagannathan, S.: Reinforcement learning neural-network-based controller for nonlinear discrete-time systems with input constraints. IEEE Trans. Syst. Man Cybern. Part B Cybern. **37**(2), 425–436 (2007)
17. Zhang, H., Wei, Q., Luo, Y.: A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy hdp iteration algorithm. SIEEE Trans. Syst. Man Cybern. Part B Cybern. **38**(4), 937–942 (2008)
18. Kamalapurkar, R., Rosenfeld, J., Dixon, W.E.: Efficient model-based reinforcement learning for approximate online optimal control. Automatica **74**, 247–258 (2016)
19. Al-Tamimi, A., Lewis, F.L., Abu-Khalaf, M.: Model-free q-learning designs for linear discrete-time zero-sum games with application to H_∞ control. Automatica **43**, 473–481 (2007)
20. Vamvoudakis, K.G., Lewis, F.L.: Multi-player non-zero-sum games: Online adaptive learning solution of coupled hamilton-jacobi equations. Automatica **47**, 1556–1569 (2011)
21. Vamvoudakis, K.G., Lewis, F.L., Hudas, G.R.: Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality. Automatica **48**(8), 1598–1611 (2012). http://www.sciencedirect.com/science/article/pii/S0005109812002476
22. Modares, H., Lewis, F.L., Naghibi-Sistani, M.-B.: Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks. IEEE Trans. Neural Netw. Learn. Syst. **24**(10), 1513–1525 (2013)
23. Kiumarsi, B., Lewis, F.L., Modares, H., Karimpour, A., Naghibi-Sistani, M.-B.: Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. Automatica **50**(4), 1167–1175 (2014)
24. Modares, H., Lewis, F.L., Naghibi-Sistani, M.-B.: Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems. Automatica **50**(1), 193–202 (2014)
25. Modares, H., Lewis, F.L.: Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning. Automatica **50**(7), 1780–1792 (2014)

26. Kamalapurkar, R., Walters, P.S., Rosenfeld, J.A., Dixon, W.E.: Reinforcement Learning for Optimal Feedback Control: A Lyapunov-Based Approach. Springer, Berlin (2018)
27. Singh, S.P.: Reinforcement learning with a hierarchy of abstract models. AAAI Natl. Conf. Artif. Intell. **92**, 202–207 (1992)
28. Atkeson, C.G., Schaal, S.: Robot learning from demonstration. Int. Conf. Mach. Learn. **97**, 12–20 (1997)
29. Abbeel, P., Quigley, M., Ng, A.Y.: Using inaccurate models in reinforcement learning. In: International Conference on Machine Learning, pp. 1–8. ACM, New York (2006)
30. Deisenroth, M.P.: Efficient Reinforcement Learning Using Gaussian Processes. KIT Scientific Publishing (2010)
31. Mitrovic, D., Klanke, S., Vijayakumar, S.: Adaptive optimal feedback control with learned internal dynamics models. In: Sigaud, O., Peters, J., (eds.), From Motor Learning to Interaction Learning in Robots. Series Studies in Computational Intelligence, vol. 264, pp. 65–84. Springer Berlin (2010)
32. Deisenroth, M.P., Rasmussen, C.E., Pilco: a model-based and data-efficient approach to policy search. In: International Conference on Machine Learning 2011, pp. 465–472 (2011)
33. Liberzon, D.: Calculus of Variations and Optimal Control Theory: A Concise Introduction. Princeton University Press, Princeton (2012)
34. Kirk, D.: Optimal Control Theory: An Introduction. Dover, Mineola (2004)
35. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
36. Konda, V., Tsitsiklis, J.: On actor-critic algorithms. SIAM J. Control Optim. **42**(4), 1143–1166 (2004)
37. Dierks, T., Jagannathan, S.: Optimal tracking control of affine nonlinear discrete-time systems with unknown internal dynamics. In: Proceedings of the IEEE Conference on Decision and Control, Shanghai, CN, Dec. 2009, pp. 6750–6755 (2009)
38. Vamvoudakis, K.G., Lewis, F.L.: Online synchronous policy iteration method for optimal control. In: Yu, W. (ed.) Recent Advances in Intelligent Control Systems, pp. 357–374. Springer, London (2009)
39. Dierks, T., Jagannathan, S.: Optimal control of affine nonlinear continuous-time systems. In: Proceedings of the American Control Conference, 2010, pp. 1568–1573 (2010)
40. Khalil, H.K.: Nonlinear Systems, 3rd edn. Prentice Hall, Upper Saddle River (2002)
41. Chowdhary, G., Concurrent learning for convergence in adaptive control without persistency of excitation, Ph.D. dissertation, Georgia Institute of Technology (2010)
42. Chowdhary, G., Johnson, E.: A singular value maximizing data recording algorithm for concurrent learning. In: Proceedings of the American Control Conference, 2011, pp. 3547–3552 (2011)
43. Chowdhary, G., Yucelen, T., Mühlegg, M., Johnson, E.N.: Concurrent learning adaptive control of linear systems with exponentially convergent bounds. Int. J. Adapt. Control Signal Process. **27**(4), 280–301 (2013)
44. Kamalapurkar, R., Walters, P., Dixon, W.E.: Concurrent learning-based approximate optimal regulation. In: Proceedings of the IEEE Conference on Decision and Control, Florence, IT, Dec. 2013, pp. 6256–6261 (2013)
45. Kamalapurkar, R., Andrews, L., Walters, P., Dixon, W.E.: Model-based reinforcement learning for infinite-horizon approximate optimal tracking. In: Proceedings of the IEEE Conference on Decision and Control, Los Angeles, CA, Dec. 2014, pp. 5083–5088 (2014)
46. Kamalapurkar, R., Klotz, J., Dixon, W.E.: Concurrent learning-based online approximate feedback Nash equilibrium solution of N-player nonzero-sum differential games. IEEE/CAA J. Autom. Sin. **1**(3), 239–247 (2014)
47. Luo, B., Wu, H.-N., Huang, T., Liu, D.: Data-based approximate policy iteration for affine nonlinear continuous-time optimal control design. Automatica (2014)
48. Yang, X., Liu, D., Wei, Q.: Online approximate optimal control for affine non-linear systems with unknown internal dynamics using adaptive dynamic programming. IET Control Theory Appl. **8**(16), 1676–1688 (2014)

49. Rosenfeld, J.A., Kamalapurkar, R., Dixon, W.E.: The state following (staf) approximation method. IEEE Trans. Neural Netw. Learn. Syst. **30**(6), 1716–1730 (2019)
50. Lorentz, G.G.: Bernstein Polynomials, 2nd edn. Chelsea Publishing Co., New York (1986)
51. Rosenfeld, J.A., Kamalapurkar, R., Dixon, W.E.: State following (StaF) kernel functions for function approximation Part I: theory and motivation. In: Proceedings of the American Control Conference, 2015, pp. 1217–1222 (2015)
52. Deptula, P., Rosenfeld, J., Kamalapurkar, R., Dixon, W.E.: Approximate dynamic programming: combining regional and local state following approximations. IEEE Trans. Neural Netw. Learn. Syst. **29**(6), 2154–2166 (2018)
53. Walters, P.S.: Guidance and control of marine craft: an adaptive dynamic programming approach, Ph.D. dissertation, University of Florida (2015)
54. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceeding of the International Conference on Artificial Intelligence and Statistics, 2011, pp. 315–323 (2011)
55. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: Proceedings of the Advances in Neural Information Processing Systems, 2007, pp. 801–808 (2007)
56. Nivison, S.A., Khargonekar, P.: Improving long-term learning of model reference adaptive controllers for flight applications: a sparse neural network approach. In: Proceedings of the AIAA Guidance, Navigation and Control Conference, Jan. 2017 (2017)
57. Nivison, S.A., Khargonekar, P.P.: Development of a robust deep recurrent neural network controller for flight applications. In: Proceedings of the American Control Conference, IEEE, 2017, pp. 5336–5342 (2017)
58. lan Boureau, Y., Cun, Y.L., Ranzato, M.: Sparse feature learning for deep belief networks. In: Proceedings of the Advances in Neural Information Processing Systems, 2008, pp. 1185–1192 (2008)
59. Nivison, S.A., Khargonekar, P.P.: Development of a robust, sparsely-activated, and deep recurrent neural network controller for flight applications. In: Proceedings of the IEEE Conference on Decision and Control, pp. 384–390. IEEE (2018)
60. Nivison, S.A., Khargonekar, P.: A sparse neural network approach to model reference adaptive control with hypersonic flight applications. In: Proceedings of the AIAA Guidance, Navigation and Control Conference, 2018, p. 0842 (2018)
61. Greene, M.L., Deptula, P., Nivison, S., Dixon, W.E.: Reinforcement learning with sparse bellman error extrapolation for infinite-horizon approximate optimal regulation. In: Proceedings of the IEEE Conference on Decision and Control, Nice, France (2019)
62. Nivison, S.A.: Sparse and deep learning-based nonlinear control design with hypersonic flight applications, Ph.D. dissertation, University of Florida (2017)
63. Walters, P., Kamalapurkar, R., Voight, F., Schwartz, E., Dixon, W.E.: Online approximate optimal station keeping of a marine craft in the presence of an irrotational current. IEEE Trans. Robot. **34**(2), 486–496 (2018)
64. Fan, Q.-Y., Yang, G.-H.: Active complementary control for affine nonlinear control systems with actuator faults. IEEE Trans. Cybern. **47**(11), 3542–3553 (2016)

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	• • • under matter to remain	
Insert in text the matter indicated in the margin		New matter followed by or
Delete	/ through single character, rule or underline or through all characters to be deleted	or
Substitute character or substitute part of one or more word(s)	/ through letter or through characters	new character / or new characters /
Change to italics	— under matter to be changed	
Change to capitals	under matter to be changed	
Change to small capitals	under matter to be changed	
Change to bold type	under matter to be changed	
Change to bold italic	under matter to be changed	
Change to lower case	Encircle matter to be changed	
Change italic to upright type	(As above)	
Change bold to non-bold type	(As above)	
Insert 'superior' character	/ through character or where required	or under character e.g. or
Insert 'inferior' character	(As above)	over character e.g.
Insert full stop	(As above)	
Insert comma	(As above)	,
Insert single quotation marks	(As above)	or and/or or
Insert double quotation marks	(As above)	or and/or or
Insert hyphen	(As above)	
Start new paragraph		
No new paragraph		
Transpose		
Close up	linking characters	
Insert or substitute space between characters or words	/ through character or where required	
Reduce space between characters or words	between characters or words affected	