Adaptive Dynamic Programming: An Introduction

©STOCKBYTE

Fei-Yue Wang, *Chinese Academy of Sciences, CHINA and University of Arizona, USA*
Huaguang Zhang, *Northeastern University, CHINA*
and Derong Liu, *Chinese Academy of Sciences, CHINA*

Abstract: In this article, we introduce some recent research trends within the field of adaptive/approximate dynamic programming (ADP), including the variations on the structure of ADP schemes, the development of ADP algorithms and applications of ADP schemes. For ADP algorithms, the point of focus is that iterative algorithms of ADP can be sorted into two classes: one class is the iterative algorithm with initial stable policy; the other is the one without the requirement of initial stable policy. It is generally believed that the latter one has less computation at the cost of missing the guarantee of system stability during iteration process. In addition, many recent papers have provided convergence analysis associated with the algorithms developed. Furthermore, we point out some topics for future studies.

Digital Object Identifier 10.1109/MCI.2009.932261

1556-603X/09/$25.00©2009IEEE

Introduction

As is well known, there are many methods for designing stable control for nonlinear systems. However, stability is only a bare minimum requirement in a system design. Ensuring optimality guarantees the stability of the nonlinear system. Dynamic programming is a very useful tool in solving optimization and optimal control problems by employing the principle of optimality. In [16], the principle of optimality is expressed as: "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision." There are several spectrums about the dynamic programming. One can consider discrete-time systems or continuous-time systems, linear systems or nonlinear systems, time-invariant systems or time-varying systems, deterministic systems or stochastic systems, etc.

We first take a look at nonlinear discrete-time (time-varying) dynamical (deterministic) systems. Time-varying nonlinear systems cover most of the application areas and discrete-time is the basic consideration for digital computation. Suppose that one is given a discrete-time nonlinear (time-varying) dynamical system

$$x(k+1) = F[(x(k), u(k), k], \ k = 0, 1, \ldots \tag{1}$$

where $x \in \mathbb{R}^n$ represents the state vector of the system and $u \in \mathbb{R}^m$ denotes the control action and F is the system function. Suppose that one associates with this system the performance index (or cost)

$$J[x(i), i] = \sum_{k=i}^{\infty} \gamma^{k-i} U[x(k), u(k), k] \tag{2}$$

where U is called the utility function and γ is the discount factor with $0 < \gamma \le 1$. Note that the function J is dependent on the initial time i and the initial state $x(i)$, and it is referred to as the cost-to-go of state $x(i)$. The objective of dynamic programming problem is to choose a control sequence $u(k)$, $k = i, i+1, \ldots,$ so that the function J (i.e., the cost) in (2) is minimized. According to Bellman, the optimal cost from time k is equal to

$$J^*(x(k)) = \min_{u(k)} \{U(x(k), u(k)) + \gamma J^*(x(k+1))\}. \tag{3}$$

The optimal control $u^*(k)$ at time k is the $u(k)$ which achieves this minimum, i.e.,

$$u^*(k) = \arg\min_{u(k)} \{U(x(k), u(k)) + \gamma J^*(x(k+1))\}. \tag{4}$$

Equation (3) is the principle of optimality for discrete-time systems. Its importance lies in the fact that it allows one to optimize over only one control vector at a time by working backward in time.

In nonlinear continuous-time case, the system can be described by

$$\dot{x}(t) = F[x(t), u(t), t], \ t \ge t_0. \tag{5}$$

The cost in this case is defined as

$$J(x(t)) = \int_t^{\infty} U(x(\tau), u(\tau)) d\tau. \tag{6}$$

For continuous-time systems, Bellman's principle of optimality can be applied, too. The optimal cost $J^*(x_0) = \min J(x_0, u(t))$ will satisfy the Hamilton-Jacobi-Bellman Equation

$$\begin{aligned} -\frac{\partial J^*(x(t))}{\partial t} &= \min_{u \in U} \left\{ U(x(t), u(t), t) + \left(\frac{\partial J^*(x(t))}{\partial x(t)}\right)^T \times F(x(t), u(t), t) \right\} \\ &= U(x(t), u^*(t), t) + \left(\frac{\partial J^*(x(t))}{\partial x(t)}\right)^T \\ &\quad \times F(x(t), u^*(t), t). \end{aligned} \tag{7}$$

Equations (3) and (7) are called the optimality equations of dynamic programming which are the basis for implementation of dynamic programming. In the above, if the function F in (1) or (5) and the cost function J in (2) or (6) are known, the solution of $u(k)$ becomes a simple optimization problem. If the system is modeled by linear dynamics and the cost function to be minimized is quadratic in the state and control, then the optimal control is a linear feedback of the states, where the gains are obtained by solving a standard Riccati equation [47]. On the other hand, if the system is modeled by nonlinear dynamics or the cost function is nonquadratic, the optimal state feedback control will depend upon solutions to the Hamilton-Jacobi-Bellman (HJB) equation [48] which is generally a nonlinear partial differential equation or difference equation. However, it is often computationally untenable to run true dynamic programming due to the backward numerical process required for its solutions, i.e., as a result of the well-known "curse of dimensionality" [16], [28]. In [69], three curses are displayed in resource management and control problems to show the cost function J, which is the theoretical solution of the Hamilton-Jacobi-Bellman equation, is very difficult to obtain, except for systems satisfying some very good conditions. Over the years, progress has been made to circumvent the "curse of dimensionality" by building a system, called "critic", to approximate the cost function in dynamic programming (cf. [10], [60], [61], [63], [70], [78], [92], [94], [95]). The idea is to approximate dynamic programming solutions by using a function approximation structure such as neural networks to approximate the cost function.

The Basic Structures of ADP

In recent years, adaptive/approximate dynamic programming (ADP) has gained much attention from many researchers in order to obtain approximate solutions of the HJB equation, cf. [2], [3], [5], [8], [11]–[13], [21], [22], [25], [30], [31], [34], [35], [40], [46], [49], [52], [54], [55], [63], [70], [76], [80], [83], [95], [96], [99], [100]. In 1977, Werbos [91] introduced an approach for ADP that was later called adaptive critic designs (ACDs). ACDs were proposed in [91], [94], [97] as a way for solving dynamic programming problems forward-in-time. In the literature, there are several synonyms used for "Adaptive Critic Designs" [10], [24], [39], [43], [54], [70], [71], [87], including "Approximate Dynamic Programming" [69], [82], [95], "Asymptotic Dynamic Programming" [75], "Adaptive Dynamic Programming" [63], [64], "Heuristic Dynamic Programming" [46], [93], "Neuro-Dynamic Programming" [17], "Neural Dynamic Programming" [82], [101], and "Reinforcement Learning" [84].

Bertsekas and Tsitsiklis gave an overview of the neuro-dynamic programming in their book [17]. They provided the background, gave a detailed introduction to dynamic programming, discussed the neural network architectures and methods for training them, and developed general convergence theorems for stochastic approximation methods as the foundation for analysis of various neuro-dynamic programming algorithms. They provided the core neuro-dynamic programming methodology, including many mathematical results and methodological insights. They suggested many useful methodologies for applications to neuro-dynamic programming, like Monte Carlo simulation, on-line and off-line temporal difference methods, Q-learning algorithm, optimistic policy iteration methods, Bellman error methods, approximate linear programming, approximate dynamic programming with cost-to-go function, etc. A particularly impressive success that greatly motivated subsequent research, was the development of a backgammon playing program by Tesauro [85]. Here a neural network was trained to approximate the optimal cost-to-go function of the game of backgammon by using simulation, that is, by letting the program play against itself. Unlike chess programs, this program did not use lookahead of many steps, so its success can be attributed primarily to the use of a properly trained approximation of the optimal cost-to-go function.

To implement the ADP algorithm, Werbos [95] proposed a means to get around this numerical complexity by using "approximate dynamic programming" formulations. His methods approximate the original problem with a discrete formulation. Solution to the ADP formulation is obtained through neural network based adaptive critic approach. The main idea of ADP is shown in Fig. 1.

He proposed two basic versions which are heuristic dynamic programming (HDP) and dual heuristic programming (DHP).

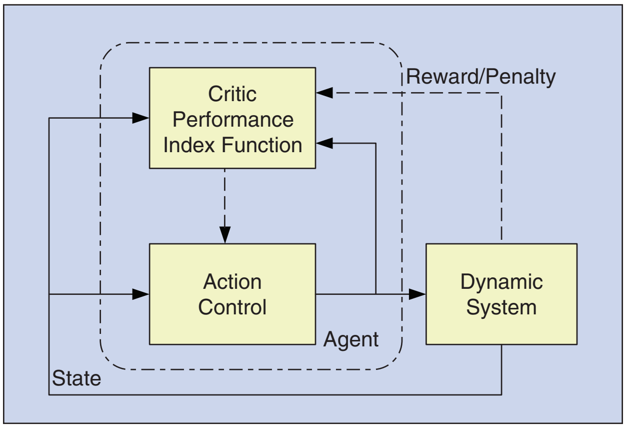

FIGURE 1 Learn from the environment.

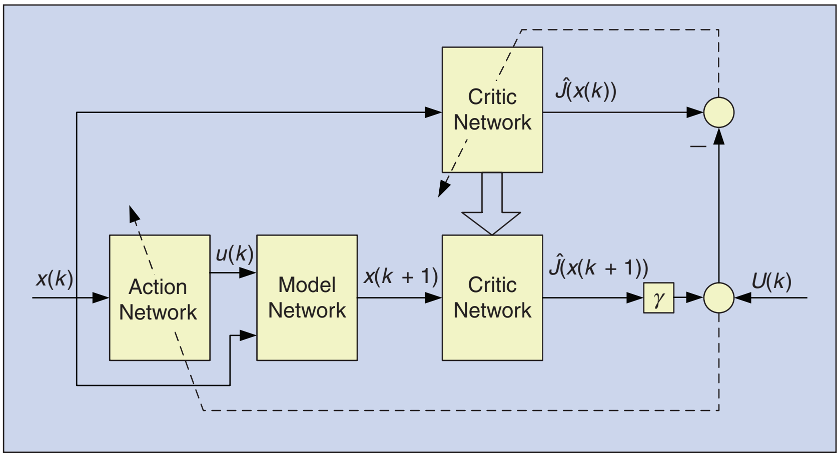

FIGURE 2 The HDP structure.

Heuristic Dynamic Programming (HDP)

HDP is the most basic and widely applied structure of ADP [13], [38], [72], [79], [90], [93], [104], [106]. The structure of HDP is shown in Fig. 2. HDP is a method for estimating the cost function. Estimating the cost function for a given policy only requires samples from the instantaneous utility function U, while models of the environment and the instantaneous reward are needed to find the cost function corresponding to the optimal policy.

In HDP, the output of the critic network is $\hat{J}$, which is the estimate of J in equation (2). This is done by minimizing the following error measure over time

$$\|E_h\| = \sum_k E_h(k) = \frac{1}{2}\sum_k [\hat{J}(k) - U(k) - \gamma\hat{J}(k+1)]^2, \quad (8)$$

where $\hat{J}(k) = \hat{J}[x(k), u(k), k, W_C]$ and W_C represents the parameters of the critic network. When $E_h = 0$ for all k, (8) implies that

$$\hat{J}(k) = U(k) + \gamma\hat{J}(k+1) \quad (9)$$

and

$$\hat{J}(k) = \sum_{i=k}^{\infty} \gamma^{i-k} U(i) \text{ which is the same as (2).}$$

Dual Heuristic Programming (DHP)

Dual heuristic programming is a method for estimating the gradient of the cost function, rather than J itself. To do this, a function is needed to describe the gradient of the instantaneous cost function with respect to the state of the system. In the DHP structure, the action network remains the same as the one for HDP, but for the second network, which is called the critic network, with the costate as its output and the state variables as its inputs.

The critic network's training is more complicated than that in HDP since we need to take into account all relevant pathways of backpropagation.

This is done by minimizing the following error measure over time

$$\|E_D\| = \sum_k E_D(k) = \frac{1}{2}\sum_k \left[\frac{\partial\hat{J}(k)}{\partial x(k)} - \frac{\partial U(k)}{\partial x(k)} - \gamma\frac{\partial\hat{J}(k+1)}{\partial x(k)}\right]^2, \tag{10}$$

where $\partial\hat{J}(k)/\partial x(k) = \partial\hat{J}[x(k), u(k), k, W_C]/\partial x(k)$ and W_C represents theparameters of the critic network. When $E_h = 0$ for all k, (10) implies that

$$\frac{\partial\hat{J}(k)}{\partial x(k)} = \frac{\partial U(k)}{\partial x(k)} + \gamma\frac{\partial J(k+1)}{\partial x(k)}. \tag{11}$$

Theoretical Developments

In [82], Si *et al* summarizes the cross-disciplinary theoretical developments of ADP and overviews DP and ADP; and discusses their relations to artificial intelligence, approximation theory, control theory, operations research, and statistics.

In [69], Powell shows how ADP, when coupled with mathematical programming, can solve (approximately) deterministic or stochastic optimization problems that are far larger than anything that could be solved using existing techniques and shows the improvement directions of ADP.

The Development of Structures

In [95], Werbos further gave two other versions called "action-dependent critics," namely, ADHDP (also known as Q-learning [89]) and ADDHP. In the two ADP structures, the control is also the input of the critic networks. In 1997, Prokhorov and Wunsch [70] presented more algorithms according to ACDs. They discussed the design families of HDP, DHP, and globalized dual heuristic programming (GDHP). They suggested some new improvements to the original GDHP design. They promised to be useful for many engineering applications in the areas of optimization and optimal control. Based on one of these modifications, they present a unified approach to all ACDs. This leads to a generalized training procedure for ACDs. In [26], a realization of ADHDP was suggested: a least squares support vector machine (SVM) regressor has been used for generating the control actions, while an SVM-based tree-type neural network (NN) is used as the critic. The GDHP or ADGDHP structure minimizes the error with respect to both the cost and its derivatives. While it is more complex to do this simultaneously, the resulting behavior is expected to be superior. So in [102], GDHP serves as a reconfigurable controller to deal with both abrupt and incipient changes in the plant dynamics due to faults. A novel fault tolerant control (FTC) supervisor is combined with GDHP for the purpose of improving the performance of GDHP for fault tolerant control. When the plant is affected by a known abrupt fault, the new initial conditions of GDHP are loaded from dynamic model bank (DMB). On the other hand, if the fault is incipient, the reconfigurable controller maintains performance by continuously modifying itself without supervisor intervention. It is noted that the training of three networks used to implement the GDHP is in an online fashion by utilizing two distinct networks to implement the critic. The first critic network is trained at every iterations while the second one is updated with a copy of the first one at a given period of iterations.

All the ADP structures can realize the same function that is to obtain the optimal control policy while the computation precision and running time are different from each other. Generally speaking, the computation burden of HDP is low but the computation precision is also low; while GDHP has better precision but the computation process will take longer time and the detailed comparison can be seen in [70].

In [30], [33] and [83], the schematic of direct heuristic dynamic programming is developed. Using the approach of [83], the model network in Fig. 1 is not needed anymore. Reference [101] makes significant contributions to model-free adaptive critic designs. Several practical examples are included in [101] for demonstration which include single inverted pendulum and triple inverted pendulum. A reinforcement learning-based controller design for nonlinear discrete-time systems with input constraints is presented by [36], where the nonlinear tracking control is implemented with filtered tracking error using direct HDP designs. Similar works also see [37]. Reference [54] is also about model-free adaptive critic designs. Two approaches for the training of critic network are provided in [54]: A forward-in-time

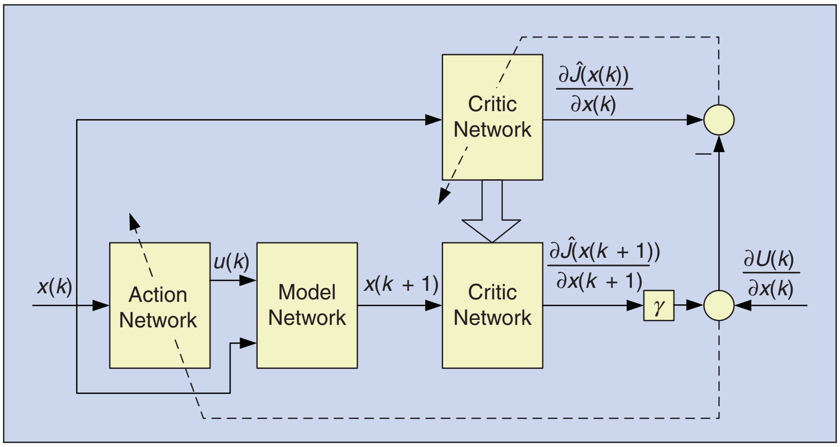
Critic Network
$\frac{\partial\hat{J}(x(k))}{\partial x(k)}$
–
x(k)
Action Network
u(k)
Model Network
x(k + 1)
Critic Network
$\frac{\partial\hat{J}(x(k+1))}{\partial x(k+1)}$
γ
$\frac{\partial U(k)}{\partial x(k)}$

FIGURE 3 The DHP structure.

approach and a backward-in-time approach. Fig. 4 shows the diagram of forward-in-time approach. In this approach, we view $\hat{J}(k)$ in (8) as the output of the critic network to be trained and choose $U(k) + \gamma\hat{J}(k+1)$ as the training target. Note that $\hat{J}(k)$ and $\hat{J}(k+1)$ are obtained using state variables at different time instances. Fig. 5 shows the diagram of backward-in-time approach. In this approach, we view $\hat{J}(k+1)$ in (8) as the output of the critic network to be trained and choose $(\hat{J}(k) - U(k))/\gamma$ as the training target. The training approach of [101] can be considered as a backward-in-time approach. In Fig. 4 and Fig. 5, $x(k+1)$ is the output of the model network.

All the ADP structures can realize the same function that is to obtain the optimal control policy while the computation precision and running time are different from each other.

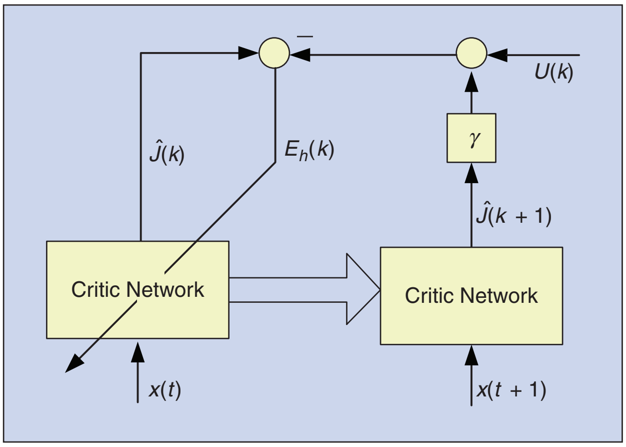

FIGURE 4 Forward-in-time approach.

An improvement and modification to the two network architecture, which is called the "single network adaptive critic (SNAC)" was presented in [65], [66]. This approach eliminates the action network. As a consequence, the SNAC architecture offers three potential advantages: a simpler architecture, lesser computational load (about half of the dual network algorithms), and no approximate error due to the fact that the action network is eliminated. The SNAC approach is applicable to a wide class of nonlinear systems where the optimal control (stationary) equation can be explicitly expressed in terms of the state and the costate variables. Most of the problems in aerospace, automobile, robotics, and other engineering disciplines can be characterized by the nonlinear control-affine equations that yield such a relation. SNAC-based controllers yield excellent tracking performances in applications to microelectronic mechanical systems, chemical reactor, and high-speed reentry problems. Padhi *et al.* [65] have proved that for linear systems (where the mapping between the costate at stage $k+1$ and the state at stage k is linear), the solution obtained by the algorithm based on the SNAC structure converges to the solution of discrete Riccati equation.

Algorithms and Convergence Analysis

The exact solution of the HJB equation is generally impossible to obtain for nonlinear systems. To overcome the difficulty in solving the HJB equation, recursive methods are employed to obtain the solution of HJB equation indirectly. In 1983, Barto *et al* [12] developed a neural computation-based adaptive critic learning method. They divides the state space into boxes and stores learned information for each box. The algorithm works well but the number of boxes can be very large for a complicated system. In 1991, Lin and Kim [51] integrate the cerebellar model articulation controller technique [1] with the box-based scheme. Large state space is mapped into a smaller physical memory space. With the distributed information storage, there is no need to reserve memory for useless boxes; this makes the structure applicable to problems of larger size. Kleinman [42] pointed out that the solution of the Riccati equation can be obtained by successively solving a sequence of Lyapunov equations, which is linear with respect to the cost function of the system, and thus, it is easier to solve than a Riccati equation, which is nonlinear with respect to the cost function. Saridis and Lee [77] extended this idea to the case of nonlinear continuous-time systems where a recursive method is used to obtain the optimal control of continuous system by successively solving the generalized Hamilton–Jacobi–Bellman (GHJB) equation, and then, updating the control action if an admissible initial control is given.

Although the GHJB equation is linear and easier to solve than HJB equation, no general solution for GHJB is demonstrated. Therefore, successful application of the successive approximation method was limited until the novel work of Beard *et al.* [15] where they used a Galerkin spectral approximation method at each iteration to find approximate solutions to the GHJB equations. And then Beard and Saridis [14] employed a series of polynomial functions as basic functions to solve the approximate GHJB equation in continuous time but this method requires the computation of a large number of integrals and it is not obvious how to handle explicit constraints on the controls. In [79], the

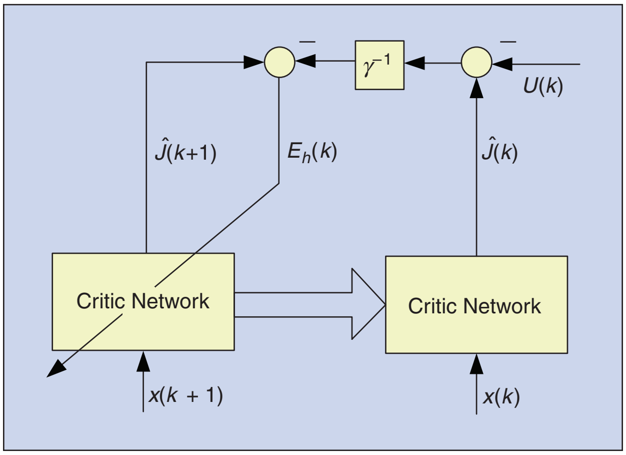

FIGURE 5 Backward-in-time approach.

Equations (3) and (7) are called the optimality equations of dynamic programming which are the basis for implementation of dynamic programming.

HJB equations are motivated and proven on time scales. The authors connected the calculus of time scales and stochastic control via ADP algorithm and further pointed out three significant directions for the investigation of ADP on time scales. Park [68] employed interpolating wavelets as the basic functions. On the other hand, Lewis and Abu-Khalaf presented how to formulate the associated Hamilton–Jacobi–Isaac (HJI) equation using special nonquadratic supply rates to obtain the nonlinear state feedback control in [9]. Next, the fixed-final-time-constrained optimal control of nonlinear systems is studied in [22], [23] based on the neural network solution of the GHJB equation. In order to enhance learning speed and final performance, Wiering and Hasselt combined multiple different reinforcement learning algorithms to design and implement four different ensemble methods in [98]. In [41], a new algorithm for the closed loop parallel optimal control of weakly coupled nonlinear systems is developed using the successive Galerkin approximation. In [45], the author inspired researchers to develop the experience-based approach which selected a controller that is appropriate to the current situation from a repository of existing controller solutions.

Although many papers have discussed the GHJB method for continuous-time systems, there is very minimal work available on the GHJB method for discrete-time nonlinear systems. Discrete-time version of the approximate GHJB-equation-based control is important since all the controllers are typically implemented by using embedded digital hardware. In [21] a successive approximation method using GHJB equation is proposed to solve the near-optimal control problem for affine nonlinear discrete-time systems, which requires small perturbation assumption and an initially stable policy. The theory of GHJB in discrete-time has also been applied to the linear discrete-time case which indicates that the optimal control is nothing but the solution of the standard Riccati equation.

On the other hand, in [19], Bradtke *et al.* implemented a Q-learning policy iteration method for the discrete-time linear quadratic optimal control problem which required an initial stable policy. Furthermore, Landelius [44] applied HDP, DHP, ADHDP and ADDHP techniques to the discrete-time linear quadratic optimal control problem without the initial stable conditions and discussed their convergence.

Based on the work of [44], the improvement of ADP to the discrete-time linear quadratic zero-sum game that appearing in the H_∞ optimal control problem is concerned in [2], [4]. The optimal strategies for discrete-time quadratic zero-sum games related to the H_∞ optimal control problem are solved in forward time. The idea is to solve for an action dependent cost function $Q(x, u, w)$ of the zero-sum game instead of solving for the state dependent cost function $J(x)$ which satisfies a corresponding game algebraic Riccati equation (GARE). Using the Kronecker method, two action networks and one critic network are used that are adaptively tuned in forward time using adaptive critic methods without the system model information. The convergence analysis is also given to guarantee the cost function to reach the saddle point of the game.

Moreover, in [3], a greedy HDP iteration scheme is proposed for solving the optimal control problem for nonlinear discrete-time systems with known mathematical model, which does not require an initially stable policy. The discrete-time system can be written as

$$x(k+1) = f(x(k)) + g(x(k))u(k), \tag{12}$$

with the cost function

$$J(x(k)) = \sum_{i=k}^{+\infty}(x^T(i)Qx(i) + u^T(i)Ru(i)), \tag{13}$$

where $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{n\times n}$ are positive definite matrices. Similar to [78], an iterative process, which is referred as Heuristic Dynamic Programming (HDP, cf. [95]), is proposed to obtain the optimal control law. Starting from $V_0(x) = 0$, define

$$\begin{cases} u_i(x(k)) = -\dfrac{1}{2}R^{-1}g^T(x(k))\left[\dfrac{\partial V_i(x(k+1))}{\partial x(k+1)}\right]^T \\ V_{i+1}(x(k)) = x^T(k)Qx(k) + u_i^T(k)(x(k))Ru_i(x(k)), \\ \qquad + V_i(x(k+1)) \end{cases} \tag{14}$$

where $x(k+1) = f(x(k)) + g(x(k))u_i(x(k))$. Al-Tamimi and Lewis [3] also provided a proof to show the cost function converges to the optimal one satisfying discrete-time Hamilton-Jacobi-Bellman (DT HJB). Zhang, Wei and Luo [104] applied the greedy iterative HDP algorithm to solve the optimal tracking problem. In [104], a new performance index is introduced to obtain a better tracking results. Using a system transformation, the optimal tracking problem is changed into an optimal regulator problem and then the greedy iterative HDP algorithm is introduced to obtain the optimal control for the transformed system. In [105], Zhang and Luo proposed a iteration scheme without requirement of initial stable policy, and proved that the cost function sequence will converge to the optimal cost function when the number of iteration steps goes to infinity. And the critic and action networks can be tuned adaptively without the system plant information via a model network. On the other hand, the optimal control problem for linear continuous-time systems without initial stable policy is studied in [88].

Murray *et al.* [63] proposed an iterative ADP scheme for continuous-time nonlinear systems with respect to quadratic cost function and succeeded to improve the autolanding control of aircraft. The iteration is required to begin with an initial stable policy, and after each iteration the cost function is updated. So the iterative policy is also called "cost iteration". The system is described by the following continuous-time differential equation

$$\dot{x} = F(x) + B(x)u,\ x(t_0) = x_0, \tag{15}$$

with the cost function

$$J(x) = \int_{t_0}^{\infty} U(x(\tau),\ u(\tau))d\tau, \tag{16}$$

where $U(x, u) = q(x) + u^T r(x)u$ is a nonnegative function and $r(x) > 0$. Similar to [78], an iterative process is proposed to obtain the control law. In this case, the optimal control can be simplified to

$$u^*(x) = -\frac{1}{2} r^{-1}(x) B^T(x) \left[\frac{dJ^*(x)}{dx}\right]^T. \tag{17}$$

Starting from any stable Lyapunov function J_0 (or alternatively, starting from an arbitrary stable controller u_0) and replacing J^* by J_i, (17) becomes

$$u_i(x) = -\frac{1}{2} r^{-1}(x) B^T(x) \left[\frac{dJ_i(x)}{dx}\right]^T, \tag{18}$$

where $J_i = \int_{t_0}^{+\infty} U(x_{i-1}, u_{i-1}) d\tau$ is the cost of the trajectory $x_{i-1}(t)$ of plant (15) under the input $u(t) = u_{i-1}(t)$. Furthermore, Murray *et al.* gave the convergence analysis of the iterative ADP scheme and the stability proof of the system. Before that most of the ADP analysis is based on the Riccati equation for linear systems. In [8], based on the work of Lyshevski [58], [59], an iterative ADP method is used to obtain an approximate solution of the cost function of the HJB equation using neural networks (NNs). A monotonic odd function is introduced to change the saturating actuators into a nonsaturating one. This in turn results in a nearly optimal constrained input state feedback controller suitable for saturated actuators. Different from the iterative ADP scheme in [63], the iterative scheme in [8] adopt policy iteration which means that after each iteration the policy (or control) function is updated. The convergence and stability analysis can also be found in [8].

Vrabie *et al.* [88] proposed a new policy iteration technique to solve online the continuous-time LQR problem for a partially model-free system (internal dynamics unknown). They presented an online adaptive critic algorithm in which the actor performs continuous-time control, whereas the critic's correction of the actor's behavior is discrete in time until best performance is obtained. The critic evaluates the actor's performance over a period of time and formulates it in a parameterized form. Policy update is a function of the critic's evaluation of the actor. Convergence of the proposed algorithm is established by proving equivalence with an established algorithm [42]. Numerical results using the short period dynamics of an F-16 aircraft are presented. In [32], a novel linear parameter-varying (LPV) approach for designing the ADP neural network controllers is presented. The control performance and the closed-loop stability of the LPV regime are formulated as a set of design equations that are linear with respect to matrix functions of NN parameters.

Applications

As for industrial applications of ADP algorithms, focuses have been on missile systems [18], autopilot [31], [50], generators [67], power systems [62], [72], communication systems [55], biochemical processes [57] and so on. In [103], an improved reinforcement learning methods are proposed to perform navigation in dynamic environments. The difficulties of the traditional reinforcement learning are presented in autonomous navigating and three effective solutions are proposed to overcome these difficulties which are forgetting Q-learning, feature based Q-learning, and hierarchical Q-learning, respectively. Forgetting Q-learning is proposed to improve performance in a dynamic environment by maintaining possible navigation paths that would be considered unacceptable by traditional Q-learning. Hierarchical Q-learning is proposed as a method of subdividing the problem domain into a set of more manageable ones. Feature based Q-learning is proposed as a method of enhancing hierarchical Q-learning. In [27], an incoherent control scheme for accomplishing the state control of a class of quantum systems which have wavefunction-controllable subspaces is proposed. This incoherent control scheme provides an alternative quantum engineering strategy for locally controllable quantum systems. In the scheme, the initial states can be unknown identical states, and the controlled system is not necessarily initially controllable.

Applications of adaptive critics in the continuous-time domain were mainly done by using discretization and well-established discrete-time results (e.g., [86]). Various schemes of continuous-time dynamic reinforcement learning were discussed in Campos and Lewis [20] and Rovithakis [74], where the derivative of Lyapunov function is approximated.

Lu, Si and Xie [56] applied a direct heuristic dynamic programming (direct HDP) to a large power system stability control problem. A direct HDP controller learns to cope with model deficiencies for nonlinearities and uncertainties on the basis of real system responses instead of a system model. Ray *et al.* [73] reported a comparison of adaptive critic-based and classical wide-area controllers for power systems. Liu *et al.* [53] demonstrated a good engine torque and exhaust air-fuel ratio (AFR) control with adaptive critic techniques for an engine application. The design was based on neural networks to automatically learn the inherent dynamics and it advanced the development of a virtual powertrain to improve its performance during the actual vehicle operations.

Enns and Si [29] presented an article on model-free approach to helicopter control. Jagannathan [81] has extended stability proofs for systems with observers in the feedback loop and applied to spark engine EGR operation on the basis of reinforcement learning dual control [37]. Al-Tamimi, Abu-Khalaf and Lewis [2] used HDP and DHP structures to solve problems formulated with game theoretic notions. Their formulation leads to a forward-in-time reinforcement learning algorithm that converges to the Nash equilibrium of the corresponding zero-sum game and they have provided performance comparisons with an F-16 autopilot problem.

Al-Tamimi *et al.* [6], [7] extended these results to a model-free environment for linear systems for the control of a power system generator. In these papers, they presented online model-free adaptive critic schemes to solve optimal control problems in both discrete-time and continuous-time domains for linear systems with unknown dynamics. In the discrete-time case, the solution process leads to solving the underlying game algebraic Riccati equation (GARE) of the corresponding optimal control problem or zero-sum game. In the continuous-time domain, the ADP scheme solves the underlying ARE of the optimal control problem. It is shown that continuous-time ADP scheme is nothing but a quasi-Newton method to solve the ARE. Either in continuous-time domain or in discrete-time domain, the adaptive critic algorithms are easy to implement the fact that initial policies are not required to be stabilizing. For the model-based paper, the authors have proved the convergence of the presented algorithm.

Concluding Remarks

In this article, we presented the variations on the structure of ADP schemes and stated the development on the iterative ADP algorithms, and at last we summarized industrial applications of ADP schemes. In the future, the study of ADP algorithms for nonlinear continuous-time systems without the requirement of initially stable policy is important. And also how to extend the ADP algorithms to time-variant and time-delay uncertain nonlinear systems with stability guarantee is another interesting topic. In addition, practical applications of ADP with significant economic impact are of great demand.

Acknowledgment

The authors would like to thank Dr. Ning Jin and Dr. Qinglai Wei for their help in preparing this manuscript.

References

[1] J. S. Albus, "A new approach to manipulator control: The cerebellar model articulation controller (CMAC)," *Trans. ASME, J. Dyn. Syst., Meas., Control*, vol. 97, pp. 220–227, Sept. 1975.

[2] A. Al-Tamimi, M. Abu-Khalaf, and F. L. Lewis, "Adaptive critic designs for discrete-time zero-sum games with application to H∞ control," *IEEE Trans. Syst., Man, Cybern. B*, vol. 37, no. 1, pp. 240–247, Feb. 2007.

[3] A. Al-Tamimi and F. L. Lewis, "Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof," in *Proc. IEEE Int. Symp. Approximate Dynamic Programming and Reinforcement Learning,* Honolulu, HI, Apr. 2007, pp. 38–43.

[4] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control," *Automatica*, vol. 43, no. 3, pp. 473–481, 2007.

[5] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," *IEEE Trans. Syst., Man., Cybern. B*, vol. 38, no. 4, pp. 943–949, Aug. 2008.

[6] A. Al-Tamimi, F. L. Lewis, and Y. Wang, "Model-free H-infinity loadfrequency controller design for power systems," in *Proc. IEEE Int. Symp. Intelligent Control,* 2007, pp. 118–125.

[7] A. Al-Tamimi, D. Vrabie, M. Abu-Khalaf, and F. L. Lewis, "Model-free approximate dynamic programming schemes for linear systems," in *Proc. IJCNN,* 2007, pp. 371–378.

[8] M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach," *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

[9] M. Abu-Khalaf, F. L. Lewis, and J. Huang, "Policy iterations on the Hamilton-Jacobi-Isaacs equation for state feedback control with input saturation," *IEEE Trans. Automat. Contr.*, vol. 51, no. 12, pp. 1989–1995, Dec. 2006.

[10] S. N. Balakrishnan and V. Biega, "Adaptive-critic-based neural networks for aircraft optimal control," *J. Guid. Control Dyn.*, vol. 19, pp. 893–898, July 1996.

[11] S. N. Balakrishnan, J. Ding, and F. L. Lewis, "Issues on stability of ADP feedback controllers for dynamical systems," *IEEE Trans. Syst., Man., Cybern. B*, vol. 38, no. 4, pp. 913–917, Aug. 2008.

[12] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Trans. Syst., Man., Cybern.*, vol. 13, no. 5, pp. 835–846, 1983.

[13] R. W. Beard, "*Improving the closed-loop performance of nonlinear systems,*" Ph.D. dissertation, Elect. Eng. Dept., Rensselaer Polytech. Inst., Troy, NY, 1995.

[14] R. W. Beard and G. N. Saridis, "Approximate solutions to the timeinvariant Hamilton-Jacobi-Bellman equation," *J. Optim. Theory Appl.*, vol. 96, no. 3, pp. 589–626, 1998.

[15] R. W. Beard, G. N. Saridis, and J. T. Wen, "Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation," *Automatica*, vol. 33, no. 12, pp. 2159–2177, 1997.

[16] R. E. Bellman, *Dynamic Programming.* Princeton, NJ: Princeton Univ. Press, 1957.

[17] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming.* Belmont, MA: Athena Scientific, 1996.

[18] D. P. Bertsekas, M. L. Homer, D. A. Logan, S. D. Patek, and N. R. Sandell, "Missile defense and interceptor allocation by neuro-dynamic programming," *IEEE Trans. Syst., Man, Cybern. A*, vol. 30, no. 1, pp. 42–51, Jan. 2000.

[19] S. J. Bradtke, B. E. Ydestie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," in *Proc. American Control Conf.,* Baltimore, MD, June 1994, pp. 3475–3476.

[20] J. Campos and F. L. Lewis, "Adaptive critic neural network for feedforward compensation," in *Proc. American Control Conf.,* San Diego, CA, June 1999, pp. 2813–2818.

[21] Z. Chen and S. Jagannathan, "Generalized Hamilton-Jacobi-Bellman formulation-based neural network control of affine nonlinear discrete-time systems," *IEEE Trans. Neural Networks*, vol. 19, no. 1, pp. 90–106, Jan. 2008.

[22] T. Cheng, F. L. Lewis, and M. Abu-Khalaf, "Fixed-final-time-constrained optimal control of nonlinear systems using neural network HJB approach," *IEEE Trans. Neural Networks*, vol. 18, no. 6, pp. 1725–1736, Nov. 2007.

[23] T. Cheng, F. L. Lewis, and M. Abu-Khalaf, "A neural network solution for fixed-final time optimal control of nonlinear systems," *Automatica*, vol. 43, no. 3, pp. 482–490, 2007.

[24] J. Dalton and S. N. Balakrishnan, "A neighboring optimal adaptive critic for missile guidance," *Math. Comput. Model.*, vol. 23, pp. 175–188, Jan. 1996.

[25] J. Dankert, Y. Lei, and J. Si, "A performance gradient perspective on approximate dynamic programming and its application to partially observable markov decision processes," in *Proc. Int. Symp. Intelligent Control,* Munich, Oct. 2006, pp. 458–463.

[26] A. K. Deb, Jayadeva, M. Gopal, and S. Chandra, "SVM-based tree-type neural networks as a critic in adaptive critic designs for control," *IEEE Trans. Neural Networks*, vol. 18, no. 4, pp. 1016–1030, July 2007.

[27] D. Dong, C. Chen, T. J. Tarn, A. Pechen, and H. Rabitz, "Incoherent control of quantum systems with wavefunction-controllable subspaces via quantum reinforcement learning," *IEEE Trans. Syst., Man, Cybern. B*, vol. 38, no. 4, pp. 957–962, Aug. 2008.

[28] S. E. Dreyfus and A. M. Law, *The Art and Theory of Dynamic Programming.* New York, NY: Academic, 1977.

[29] R. Enns and J. Si, "Apache helicopter stabilization using neural dynamic programming," *J. Guid. Control Dyn.*, vol. 25, no. 1, pp. 19–25, 2002.

[30] R. Enns and J. Si, "Helicopter trimming and tracking control using direct neural dynamic programming," *IEEE Trans. Neural Networks*, vol. 14, no. 4, pp. 929–939, July 2003.

[31] S. Ferrari and R. F. Stengel, "Online adaptive critic flight control," *J. Guid. Control Dyn.*, vol. 27, no. 5, pp. 777–786, 2004.

[32] S. Ferrari, J. E. Steck, and R. Chandramohan, "Adaptive feedback control by constrained approximate dynamic programming," *IEEE Trans. Syst., Man., Cybern. B*, vol. 38, no. 4, pp. 982–987, Aug. 2008.

[33] J. Govindhassamy, S. Mcloone, and G. Irwin, "Second-order training of adaptive critics for online process control," *IEEE Trans. Syst., Man., Cybern. B*, vol. 35, no. 2, pp. 381–385, Apr. 2005.

[34] T. Hanselmann, L. Noakes, and A. Zaknich, "Continuous-time adaptive critics," *IEEE Trans. Neural Networks*, vol. 18, no. 3, pp. 631–647, May 2007.

[35] R. Havira and J. Lewis, "Computation of quantized controls using differential dynamic programming," *IEEE Trans. Automat. Contr.*, vol. 17, no. 2, pp. 191–196, Apr. 1972.

[36] P. He and S. Jagannathan, "Reinforcement learning-based output feedback control of nonlinear systems with input constraints," *IEEE Trans. Syst., Man., Cybern. B*, vol. 35, no. 1, pp. 150–154, Jan. 2005.

[37] P. He and S. Jagannathan, "Reinforcement learning neural-network-based controller for nonlinear discrete-time systems with input constraints," *IEEE Trans. Syst., Man., Cybern. B*, vol. 37, no. 2, pp. 425–436, Apr. 2007.

[38] Z. G. Hou and C. P. Wu, "A dynamic programming neural network for large-scale optimization problems," *Acta Autom. Sin.*, vol. 25, no. 1, pp. 46–51, 2005.

[39] H. Javaherian, D. Liu, Y. Zhang, and O. Kovalenko, "Adaptive critic learning techniques for automotive engine control," in *Proc. American Control Conf.,* Boston, MA, June 2004, pp. 4066–4071.

[40] N. Jin, D. Liu, T. Huang, and Z. Pang, "Discrete-time adaptive dynamic programming using wavelet basis function neural networks," in *Proc. IEEE Int. Symp. Approximate Dynamic Programming and Reinforcement Learning,* Honolulu, HI, Apr. 2007, pp. 135–142.

[41] Y. J. Kim and M. T. Lim, "Parallel optimal control for weakly coupled nonlinear systems using successive Galerkin approximation," *IEEE Trans. Automat. Contr.*, vol. 53, no. 6, pp. 1542–1547, July 2008.

[42] D. Kleinman, "On a iterative technique for Riccati equation computations," *IEEE Trans. Automat. Contr.*, vol. 13, no. 1, pp. 114–115, Feb. 1968.

[43] N. V. Kulkarni and K. KrishnaKumar, "Intelligent engine control using an adaptive critic," *IEEE Trans. Contr. Syst. Technol.*, vol. 11, pp. 164–173, Mar. 2003.

[44] T. Landelius, *"Reinforcement learning and distributed local model synthesis,"* Ph.D. dissertation, Linkoping University, Sweden, 1997.

[45] G. G. Lendaris, "Higher level application of ADP A nextphase for the control field," *IEEE Trans. Syst., Man., Cybern. B*, vol. 38, no. 4, pp. 901–912, Aug. 2008.

[46] G. G. Lendaris and C. Paintz, "Training strategies for critic and action neural networks in dual heuristic programming method," in *Proc. 1997 IEEE Int. Conf. Neural Networks,* Houston, TX, June 1997, pp. 712–717.

[47] F. L. Lewis, *Applied Optimal Control and Estimation.* Upper Saddle River, NJ: Prentice-Hall, 1992.

[48] F. L. Lewis and V. L. Syrmos, *Optimal Control.* New York, NY: Wiley, 1995.

[49] B. Li and J. Si, "Robust dynamic programming for discounted infinite-horizon Markov decision processes with uncertain stationary transition matrices," in *Proc. IEEE Int. Symp. Approximate Dynamic Programming and Reinforcement Learning,* Honolulu, HI, Apr. 2007, pp. 96–102.

[50] C. K. Lin, "Adaptive critic autopilot design of bank-to-turn missiles using fuzzy basis function networks," *IEEE Trans. Syst., Man., Cybern. B*, vol. 35, no. 2, pp.197–207, Apr. 2005.

[51] C. S. Lin and H. Kim, "CMAC-based adaptive critic self-learning control," *IEEE Trans. Neural Networks*, vol. 2, no. 5, pp. 530–533, Sept. 1991.

[52] X. Liu and S. N. Balakrishnan, "Convergence analysis of adaptive critic based optimal control," in *Proc. American Control Conf.,* Chicago, IL, June 2000, pp. 1929–1933.

[53] D. Liu, H. Javaherian, O. Kovalenko, and T. Huang, "Adaptive critic learning techniques for engine torque and air–fuel ratio control," *IEEE Trans. Syst., Man., Cybern. B*, vol. 38, no. 4, pp. 988–993, Aug. 2008.

[54] D. Liu, X. Xiong, and Y. Zhang, "Action-dependent adaptive critic designs," in *Proc. Int. Joint Conf. Neural Networks,* Washington, D.C., July 2001, vol. 2, pp. 990–995.

[55] D. Liu, Y. Zhang, and H. Zhang, "A self-learning call admission control scheme for CDMA cellular networks," *IEEE Trans. Neural Networks*, vol. 16, no. 5, pp. 1219–1228, Sept. 2005.

[56] C. Lu, J. Si, and X. Xie, "Direct heuristic dynamic programming for damping oscillations in a large power system," *IEEE Trans. Syst., Man., Cybern. B*, vol. 38, no. 4, pp. 1008–1013, Aug. 2008.

[57] M. S. Iyer and D. C. Wunsch, "Dynamic re-optimization of a fed-batch fermentor using adaptive critic designs," *IEEE Trans. Neural Networks*, vol. 12, no. 6, pp. 1433–1444, Nov. 2001.

[58] S. E. Lyshevski, "Optimization of dynamic systems using novel performance functionals," in *Proc. 41st Conf. Decision Control,* Las Vegas, NV, Dec. 2002, pp. 753–758.

[59] S. E. Lyshevski, "Optimal control of nonlinear continuous-time systems: Design of bounded controllers via generalized nonquadratic functionals," in *Proc. American Control Conf.,* Philadelphia, PA, June 1998, pp. 205–209.

[60] S. Mohahegi, G. K. Venayagamoorth, and R. G. Harley, "Adaptive critic design based neuro-fuzzy controller for a static compensator in a multimachine power system," *IEEE Trans. Power Syst.*, vol. 21, no. 4, pp. 1744–1754, Nov. 2006.

[61] S. Mohahegi, G. K. Venayagamoorth, and R. G. Harley, "Fully evolvable optimal neurofuzzy controller using adaptive critic designs," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 6, pp. 1450–1461, Dec. 2008.

[62] S. Mohagheghi, Y. D. Valle, G. K. Venayagamoorthy, and R. G. Harley, "A proportional-integrator type adaptive critic design-based neurocontroller for a static compensator in a multimachine power system," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 86–96, Feb. 2007.

[63] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst., Man., Cybern. C*, vol. 32, no. 2, pp. 140–153, May 2002.

[64] J. J. Murray, C. J. Cox, and R. E. Saeks, "The adaptive dynamic programming theorem," in *Stability and Control of Dynamical Systems with Applications,* D. Liu and P. J. Antsaklis, Eds. Boston, MA: Birkhaüser, 2003, pp. 379–394.

[65] R. Padhi, N. Unnikrishnan, X. Wang, and S. N. Balakrishnan, "A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems," *Neural Netw.*, vol. 19, no. 10, pp. 1648–1660, Dec. 2006.

[66] R. Padhi and S. N. Balakrishnan, "Optimal management of beaver population using a reduced-order distributed parameter model and single network adaptive critics," *IEEE Trans. Contr. Syst. Technol.*, vol. 14, no. 4, pp. 628–640, July 2006.

[67] J. W. Park, R. G. Harley, and G. K. Venayagamoorthy, "Adaptive-critic-based optimal neurocontrol for synchronous generators in a power system using MLP/RBF neural networks," *IEEE Trans. Ind. Applicat.*, vol. 39, no. 5, pp. 1529–1540, Sept. 2003.

[68] C. Park and P. Tsiotras, "Approximations to optimal feedback control using a successive wavelet collocation algorithm," in *Proc. American Control Conf.,* 2003, vol. 3, pp. 1950–1955.

[69] W. B. Powell, *Approximate Dynamic Programming Solving the Curses of Dimensionality.* Princeton, NJ: Wiley, 2007.

[70] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Networks*, vol. 8, no. 5, pp. 997–1007, Sept. 1997.

[71] D. V. Prokhorov, R. A. Santiago, and D. C. Wunsch, "Adaptive critic designs: A case study for neurocontrol," *Neural Netw.*, vol. 8, pp. 1367–1372, 1995.

[72] W. Qiao, R. G. Harley, and G. K. Venayagamoorthy, "Coordinated reactive power control of a large wind farm and a STATCOM using heuristic dynamic programming," *IEEE Trans. Energy Conversion*, to be published.

[73] S. Ray, G. K. Venayagamoorthy, B. Chaudhuri, and R. Majumder, "Comparison of adaptive critic-based and classical wide-area controllers for power systems," *IEEE Trans. Syst., Man., Cybern. B*, vol. 38, no. 4, pp. 1002–1007, Aug. 2008.

[74] G. A. Rovithakis, "Stable adaptive neuro-control design via Lyapunov function derivative estimation," *Automatica*, vol. 37, no. 8, pp. 1213–1221, Aug. 2001.

[75] R. E. Saeks, C. J. Cox, K. Mathia, and A. J. Maren, "Asymptotic dynamic programming: Preliminary concepts and results," in *Proc. 1997 IEEE Int. Conf. Neural Networks,* Houston, TX, June 1997, pp. 2273–2278.

[76] G. Saridis and C. S. Lee, "An approximation theory of optimal control for trainable manipulators," *IEEE Trans. Syst., Man., Cybern.*, vol. 9, no. 3, pp. 152–159, 1979.

[77] G. N. Saridis and C. S. Lee, "An approximation theory of optimal control for trainable manipulators," *IEEE Trans. Syst., Man., Cybern.*, vol. 9, no. 3, pp. 152–159, 1979.

[78] G. N. Saridis and F. Y. Wang, "Suboptimal control of nonlinear stochastic systems," *Control Theory Adv. Technol.*, vol. 10, no. 4, pp. 847–871, 1994.

[79] J. Seiffertt, S. Sanyal, and D. C. Wunsch, "Hamilton–Jacobi–Bellman equations and approximate dynamic programming on time scales," *IEEE Trans. Syst., Man, Cybern. B*, vol. 38, no. 4, pp. 918–923, 2008.

[80] S. Shervais, T. T. Shannon, and G. G. Lendaris, "Intelligent supply chain management using adaptive critic learning," *IEEE Trans. Syst., Man, Cybern. A*, vol. 33, no. 2, pp. 235–244, Mar. 2003.

[81] P. Shih, B. Kaul, S. Jagannathan, and J. Drallmeier, "Near optimal output-feedback control of nonlinear discrete-time systems in nonstrict feedback form with application to engines," in *Proc. IJCNN Conf.,* 2007, pp. 396–401.

[82] J. Si, A. Barto, W. Powell, and D. Wunsch, *Handbook of Learning Dynamic Programming.* Hoboken, New Jersey: Wiley, 2004.

[83] J. Si and Y. T. Wang, "On-line learning control by association and reinforcement," *IEEE Trans. Neural Networks*, vol. 12, no. 2, pp. 264–276, Mar. 2001.

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

[85] G. J. Tesauro, "Practical issues in temporal difference learning," *Mach. Learn.*, vol. 8, pp. 257–277, 2000.

[86] J. N. Tsitsiklis, "Efficient algorithms for globally optimal trajectories," *IEEE Trans. Automat. Contr.*, vol. 40, no. 9, pp. 1528–1538, Sept. 1995.

[87] G. K. Venayagamoorthy, R. G. Harley, and D. G. Wunsch, "Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator," *IEEE Trans. Neural Networks*, vol. 13, pp. 764–773, May 2002.

[88] D. Vrabie, M. Abu-Khalaf, F. L. Lewis, and Y. Wang, "Continuous-time ADP for linear systems with partially unknown dynamics," in *Proc. 2007 IEEE Symp. Approximate Dynamic Programming and Reinforcement Learning,* Honolulu, HI, USA, pp. 247–253.

[89] C. Watkins, *"Learning from delayed rewards,"* Ph.D. dissertation, Cambridge Univ., Cambridge, England, 1989.

[90] Q. Wei, H. Zhang, and J. Dai, "Model-free multiobjective approximate dynamic programming for discrete-time nonlinear systems with general performance index functions," *Neurocomputing*, vol. 72, no. 7–9, pp. 1839–1848, 2009.

[91] P. J. Werbos, "Advanced forecasting methods for global crisis warning and models of intelligence," *Gen. Syst. Yearbk.*, vol. 22, pp. 25–38, 1977.

[92] P. J. Werbos, "Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research," *IEEE Trans. Syst., Man, Cybern.*, vol. 17, no. 1, pp. 7–20, Jan. 1987.

[93] P. J. Werbos, "Consistency of HDP applied to a simple reinforcement learning problem," *Neural Netw.*, vol. 3, no. 2, pp. 179–189, 1990.

[94] P. J. Werbos, "A menu of designs for reinforcement learning over time," in *Neural Networks for Control,* W. T. Miller, R. S. Sutton, and P. J. Werbos, Eds. Cambridge, MA: MIT Press, 1990, pp. 67–95.

[95] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches,* D. A. White and D. A. Sofge, Eds. New York: Van Nostrand, 1992, ch. 13.

[96] P. J. Werbos, "Using ADP to understand and replicate brain intelligence: the next level design," in *Proc. IEEE Symp. Approximate Dynamic Programming and Reinforcement Learning,* Honolulu, HI, Apr. 2007, pp. 209–216.

[97] B. Widrow, N. Gupta, and S. Maitra, "Punish/reward: Learning with a critic in adaptive threshold systems," *IEEE Trans. Syst., Man., Cybern.*, vol. 3, no. 5, pp. 455–465, Sept. 1973.

[98] M. A. Wiering and H. V. Hasselt, "Ensemble algorithms in reinforcement learning," *IEEE Trans. Syst., Man., Cybern. B*, vol. 38, no. 4, pp. 930–936, Aug. 2008.

[99] V. Yadav, R. Padhi, and S. N. Balakrishnan, "Robust/optimal temperature profile control of a high-speed aerospace vehicle using neural networks," *IEEE Trans. Neural Networks*, vol. 18, no. 4, pp. 1115–1128, July 2007.

[100] Q. Yang and S. Jagannathan, "Online reinforcement learning neural network controller design for nanomanipulation," in *Proc. IEEE Symp. Approximate Dynamic Programming and Reinforcement Learning,* Honolulu, HI, Apr. 2007, pp. 225–232.

[101] L. Yang, R. Enns, Y. T. Wang, and J. Si, "Direct neural dynamic programming," in *Stability and Control of Dynamical Systems with Applications,* D. Liu and P. J. Antsaklis, Eds. Boston, MA: Birkhauser, 2003.

[102] G. G. Yen and P. G. DeLima, "Improving the performance of globalized dual heuristic programming for fault tolerant control through an online learning supervisor," *IEEE Trans. Automat. Sci. Eng.*, vol. 2, no. 2, pp. 121–131, Apr. 2005.

[103] G. G. Yen and T. W. Hickey, "Reinforcement learning algorithms for robotic navigation in dynamic environments," *ISA Trans.*, vol. 43, pp. 217–230, 2004.

[104] H. Zhang, Q. Wei, and Y. Luo, "A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear system based on greedy HDP iteration algorithm," *IEEE Trans. Syst., Man, Cybern. B*, vol. 38, no. 4, pp. 937–942, Aug. 2008.

[105] H. Zhang and Y. Luo, "RBF neural network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints," *IEEE Trans. Neural Networks*, to be published.

[106] Y. Zhao, S. D. Patek, and P. A. Beling, "Decentralized Bayesian search using approximate dynamic programming methods," *IEEE Trans. Syst., Man., Cybern. B*, vol. 38, no. 4, pp. 970–975, Aug. 2008.