Neural Network-Based Adaptive Optimal Controller - A Continuous-Time Formulation -*

D. Vrabie, F.L. Lewis, *Fellow IEEE,* D. Levine

Abstract— **In this paper is presented new online adaptive control scheme, for partially unknown nonlinear systems, which converges to the optimal state-feedback control solution for affine in the input nonlinear systems. The main features of the algorithm map on the characteristics of the rewards-based decision making process in the mammal brain.**

The derivation of the optimal adaptive control algorithm is presented in a continuous-time framework. The optimal control solution will be obtained in a direct fashion, without system identification. The algorithm is an online approach to policy iterations based on an adaptive critic structure to find an approximate solution to the state feedback, infinite-horizon, optimal control problem.

I. Introduction

It is well known that solving the optimal control problem is generally difficult even in the presence of complete and correct knowledge of the system dynamics, as Bellman's dynamic programming approach suffers from the so called "curse of dimensionality" [16]. This motivated several advances in solving the optimal control problem using dual adaptive control techniques [8], surveyed in [9], [29], which would simultaneously improve the estimated system model parameters and improve on the suboptimal controller. Nonetheless, another difficulty appeared, posed by dual control theory, known as the exploration-exploitation dilemma [24].

In order to adaptively solve optimal control problems a new methodology, namely Reinforcement Learning (RL), was developed in the computational intelligence community and then gradually adapted to fit the control engineering requirements. Reinforcement learning means finding a control policy, i.e. learning the parameters of a controller mapping between the system states and the control signal, such that to maximize a numerical reward signal [24]. Reinforcement learning is defined by characterizing a learning problem which is in fact the adaptive optimal control problem. Thus, from a control engineering perspective, RL algorithms can be viewed as a class of adaptive controllers which solve the optimal control problem based on reward information which characterizes the performance of a given controller.

*This paper is a revised version of the paper previously published at the 2008 International Conference on Intelligent Computing. This work was supported by the National Science Foundation ECS-0501451, ECCS-0801330 and the Army Research Office W91NF-05-1-0314.

D. Vrabie and F. Lewis are with the Automation and Robotics Research Institute, University of Texas at Arlington, 7300 Jack Newell Blvd. S. Fort Worth, TX 76118 USA (phone/fax: +817-272-5938; e-mail: dvrabie@uta.edu).

D. Levine is with the Department of Psychology, University of Texas at Arlington, Arlington, TX 76019-0528 USA

In this paper we will focus our attention on a class of reinforcement learning algorithms, namely policy iteration. The goal of the paper is to present a new policy iteration algorithm which, without making use of complete knowledge of a system's dynamics, will learn to approximate, in an online fashion and with arbitrary small accuracy, the optimal control solution for a general nonlinear affine in the input continuous-time system.

In order to solve the optimal control problem, instead of directly solving the Hamilton-Jacobi-Bellman (HJB) equation [16] for the optimal cost and then finding the optimal control policy (*i.e.* the feedback gain for linear systems), the policy iteration method starts with the evaluation of the cost associated with an initial stabilizing control policy and then uses this information to obtain a new policy which will result in improved control performances. The algorithm can be viewed as a directed search for the optimal controller in the space of admissible control policies.

Policy iteration algorithm was first formulated in [13]. For continuous state linear systems policy iteration algorithms were developed in [5], [19] and [25] used to find the optimal Linear Quadratic Regulator (LQR) [16]. Convergence guarantees were given in [11] and [14]. In [5] policy iteration was formulated to solve the discrete-time LQR problem using Q-functions [26], [27], thus the resulting algorithm is model free. For continuous-time systems, in [19], the model free quality of the approach was achieved either by evaluating online the infinite horizon cost associated with an admissible control policy or by using measurements of the state derivatives. The policy iteration algorithm in [25] is an online technique which solves the LQR problem along a single state trajectory, using only partial knowledge about the system dynamics and without requiring measurements of the state derivative.

In the case on nonlinear systems policy iteration is in fact the method of successive approximations developed in [21]. This method iterates on a sequence of Lyapunov equations which are somewhat easier to solve than the HJB equation. In [2], [3] the solution for these Lyapunov equations was obtained using the Galerkin spectral approximation method and in [1] they were solved, in the presence of saturation restrictions on the control input, using neural network approximator structures. Neural network-based structures for learning the optimal control solution via the HJB equation, namely Adaptive Critics, were first proposed in [18]. Adaptive Critics and neural network training algorithms were presented both in discrete-time, [20], and continuous-time, [10], framework.

The policy iteration methods developed in [2], [3] and [1] are generally applied offline as they require complete knowledge on the dynamics of the system to be controlled. Stabilizing adaptive controllers that are inverse optimal, with respect to some relevant cost not specified by the designer,

have also been derived [17]. Due to their offline character imposed by the system model requirement these methods are not sensitive to changes in the system dynamics. The algorithm that we present in this paper is a policy iteration algorithm which uses the Bellman optimality equation as a consistence relation when solving for the value associated with a given policy, and not the regular, Hamiltonian-based, Lyapunov equation. This determines in the model free property of the proposed algorithm and grants its online implementation feature.

In the next section the continuous-time optimal control problem for nonlinear systems is formulated. The new online policy iteration algorithm is then presented followed by its neural network based online implementation, on an Actor-Critic structure. The relation of the algorithm with certain learning mechanisms in the mammal brain is then discussed followed by concluding remarks.

II. The Optimal Control Problem

Consider the time-invariant affine in the input dynamical system given by

$$\dot{x}(t)=f(x(t))+g(x(t))u(x(t));\ x(0)=x_0 \tag{1}$$

with $x(t)\in\mathbb{R}^n$, $f(x(t))\in\mathbb{R}^n$, $g(x(t))\in\mathbb{R}^{n\times m}$ and the input $u(t)\in U\subset\mathbb{R}^m$. We assume that $f(x)+g(x)u$ is Lipschitz continuous on a set $\Omega\subseteq\mathbb{R}^n$ that contains the origin and that the dynamical system is stabilizable on Ω, *i.e.* there exists a continuous control function $u(t)\in U$ such that the system is asymptotically stable on Ω.

Define the infinite horizon integral cost

$$V(x_0)=\int_0^\infty r(x(\tau),u(\tau))d\tau \tag{2}$$

where $r(x,u)=Q(x)+u^TRu$ with $Q(x)$ positive definite, *i.e.* $\forall x\neq 0, Q(x)>0$ and $x=0\Rightarrow Q(x)=0$, and $R\in\mathbb{R}^{m\times m}$ is a positive definite matrix.

Definition 1 (Admissible policy) A control policy $\mu(x)$ is defined as admissible with respect to (2) on Ω, denoted by $\mu\in\Psi(\Omega)$, if $\mu(x)$ is continuous on Ω, $\mu(0)=0$, $\mu(x)$ stabilizes (1) on Ω and $V(x_0)$ is finite $\forall x_0\in\Omega$.

For any admissible control policy $\mu\in\Psi(\Omega)$ if the associated cost function

$$V^\mu(x_0)=\int_0^\infty r(x(\tau),\mu(x(\tau)))d\tau \tag{3}$$

is C^1 then a infinitesimal version of (3) is

$$0=r(x,\mu(x))+V_x^{\mu T}(f(x)+g(x)\mu(x)),\ V^\mu(0)=0 \tag{4}$$

where V_x^μ denotes the partial derivative of the value function V^μ with respect to x, as the value function does not depend explicitly on time. Equation (4) is a Lyapunov equation for nonlinear systems which, given the controller $\mu(x)\in\Psi(\Omega)$, can be solved for the value function $V^\mu(x)$ associated with it. Given that $\mu(x)$ is an admissible control policy, if $V^\mu(x)$ satisfies (4), with $r(x,\mu(x))\geq 0$, then $V^\mu(x)$ is a Lyapunov function for the system (1) with control policy $\mu(x)$.

The optimal control problem can now be formulated: Given the continuous-time system (1), the set $u\in\Psi(\Omega)$ of admissible control policies and the infinite horizon cost functional (2), find an admissible control policy such that the cost index (2) associated with the system (1) is minimized.

Defining the Hamiltonian of the problem

$$H(x,u,V_x^*)=r(x(t),u(t))+V_x^{*T}(f(x(t))+g(x(t))u(t)) \tag{5}$$

the optimal cost function $V^*(x)$ satisfies the HJB equation

$$0=\min_{u\in\Psi(\Omega)}[H(x,u,V_x^*)] \tag{6}$$

Assuming that the minimum on the right hand side of the equation (6) exists and is unique then the optimal control function for the given problem is

$$u^*(x)=-R^{-1}g^T(x)V_x^*(x) \tag{7}$$

Inserting this optimal control in the Hamiltonian we obtain the HJB equation in terms of V_x^*

$$\begin{aligned}&0=Q(x)+V_x^{*T}(x)f(x)-\frac{1}{4}V_x^{*T}(x)g(x)R^{-1}g^T(x)V_x^*(x)\\&V^*(0)=0\end{aligned} \tag{8}$$

This is a necessary and sufficient condition for the optimal value function [16]. For the linear system case, considering a quadratic cost functional, the equivalent of this HJB equation is the well known Riccati equation.

In order to find the optimal control solution for the problem one only needs to solve the HJB equation (8) for the value function and then substitute the solution in (7) to obtain the optimal control. However, solving the HJB equation is generally difficult as it is a nonlinear differential equation, quadratic in the cost function, which also requires complete knowledge of the system dynamics (*i.e.* the system dynamics described by the functions $f(x), g(x)$ need to be known).

III. The Policy Iteration Algorithm

In order to solve the optimal control problem, instead of directly solving the HJB equation (8) for the optimal cost and then finding the optimal control policy given by (7), the policy iteration method starts by evaluating the cost of a given initial admissible policy and then makes use of this information to improve the control policy. The two steps are repeated until the policy improvement step no longer changes the actual policy. The following online reinforcement learning algorithm will solve the infinite horizon optimal control problem without using knowledge regarding the system internal dynamics (*i.e.* the system function $f(x)$).

First note that given an admissible policy for (1), $\mu(x)$, such that the closed loop system is asymptotically stable on Ω, then the infinite horizon cost for any $x(t)\in\Omega$ is given by

(3) and $V^{\mu}(x(t))$ serves as a Lyapunov function for (1). The cost function (3) can thus be written as

$$V^{\mu}(x(t))=\int_{t}^{t+T} r(x(\tau),\mu(x(\tau)))d\tau+V^{\mu}(x(t+T)). \quad (9)$$

Based on (9) and (6), considering an initial admissible control policy $\mu^{(0)}(x)$, the following policy iteration scheme can be derived:

1. solve for $V^{\mu^{(i)}}(x)$ using

$$V^{\mu^{(i)}}(x(t))=\int_{t}^{t+T} r(x(\tau),\mu^{(i)}(x(\tau)))d\tau+V^{\mu^{(i)}}(x(t+T)), \quad (10)$$
$$V^{\mu^{(i)}}(0)=0$$

2. update the control policy using

$$\mu^{(i+1)}(x)=\arg\min_{\mu}\{H(x,\mu,V_x^{\mu^{(i)}})\} \quad (11)$$

which in this case is

$$\mu^{(i+1)}(x)=-R^{-1}g^{T}(x)V_x^{\mu^{(i)}}(x) \quad (12)$$

Equations (10) and (12) formulate a new policy iteration algorithm to solve for the optimal control without making use of any knowledge of the system internal dynamics $f(x)$. The online implementation of the algorithm will be discussed in next section. This algorithm is an online version of the offline algorithms proposed in [2], [3], [1], algorithm inspired by the online adaptive critic techniques proposed by computational intelligence researchers [4], [20], [28].

The convergence of the algorithm is now discussed.

Lemma 1 Solving for $V^{\mu^{(i)}}$ in equation (10) is equivalent with finding the solution of the Lyapunov equation

$$0=r(x,\mu^{(i)}(x))+V_x^{\mu^{(i)}T}(f(x)+g(x)\mu^{(i)}(x)) \quad (13)$$
$$V^{\mu^{(i)}}(0)=0$$

The proof is based on the fact that the solution of the Lyapunov equation (13), $V^{\mu^{(i)}}$, satisfies also equation (10), and that equation (10) has a unique solution.

Remark 1 Note that although the same solution is obtained whether solving the equation (10) or (13), solving equation (10) does not require any knowledge on the system dynamics $f(x)$.

From Lemma 1 it follows that the algorithm (10) and (12) is equivalent to iterating between (13) and (12), without using knowledge of the system internal dynamics.

Theorem 1 *(convergence)* The policy iteration algorithm (10) and (12) converges to the optimal control solution on the trajectories having initial state $x_0\in\Omega$.

Proof: In [2], [3], [1] it was shown that using policy iteration conditioned by an initial admissible policy $\mu^{(0)}(x)$, all the subsequent control policies will be admissible and the iteration (13) and (12) will converge to the solution of the HJB equation. Based on the proven equivalence between the equations (10) and (13) we can conclude that the proposed online adaptive optimal control algorithm will converge to the solution of the optimal control problem (2) without using knowledge on the internal dynamics of the controlled system (1). ▌

IV. Online Neural Network-Based Approximate Optimal Control Solution on an Actor-Critic Structure

For the implementation of the iteration scheme given by (10) and (12) one only needs to have knowledge of the input to state dynamics, *i.e.* the function $g(x)$, which is required for the policy update in equation (12); however no knowledge on the internal state dynamics, described by $f(x)$, is required.

In order to solve for the cost function $V^{\mu^{(i)}}(x)$ in equation (10) we will use a neural network, which is a universal approximator [12], to obtain an approximation of the value function for any given initial state $x\in\Omega$. The cost function $V^{\mu^{(i)}}(x(t))$ will be approximated by

$$V^{\mu^{(i)}}(x)=\sum_{j=1}^{L} w_j^{\mu^{(i)}}\phi_j(x)=(\mathbf{w}_L^{\mu^{(i)}})^{T}\boldsymbol{\varphi}_L(x) \quad (14)$$

a neural network with L neurons on the hidden layer and activation functions $\phi_j(x)\in C^1(\Omega),\phi_j(0)=0$. $w_j^{\mu^{(i)}}$ denote the weights of the neural network, $\boldsymbol{\varphi}_L(x)$ is the vector of activation functions and $\mathbf{w}_L^{\mu^{(i)}}$ is the weight vector. The issues related with the neural network approximation error will be addressed in a future paper while we continue the following derivations assuming that the neural network is an exact description of the cost function.

Using the neural network description for the value function, equation (14), equation (10) can be written as

$$\mathbf{w}_L^{\mu^{(i)}T}\boldsymbol{\varphi}_L(x(t))=\int_{t}^{t+T} r(x,\mu^{(i)}(x))d\tau+\mathbf{w}_L^{\mu^{(i)}T}\boldsymbol{\varphi}_L(x(t+T)). \quad (15)$$

As the cost function was replaced with the neural network approximation, equation (15) will have the residual error

$$\delta_L^i(x(t),T)=\int_{t}^{t+T} r(x,\mu^{(i)}(x))d\tau+ \quad (16)$$
$$+\mathbf{w}_L^{\mu^{(i)}T}[\boldsymbol{\varphi}_L(x(t+T))-\boldsymbol{\varphi}_L(x(t))].$$

From the perspective of temporal difference learning methods, *e.g.* [7], this error can be viewed as temporal difference residual error.

To determine the parameters of the neural network approximating the cost function, in the least-squares sense, we use the method of weighted residuals. Thus we seek to minimize the objective

$$S=\int_{\Omega}\delta_L^i(x,T)\delta_L^i(x,T)dx \quad (17)$$

Using the inner product notation for the Lebesgue integral one can write

$$\left\langle \frac{d\delta_L^i(x,T)}{d\mathbf{w}_L^{\mu^{(i)}}},\delta_L^i(x,T)\right\rangle_{\Omega}=0. \quad (18)$$

Thus, conditioned by

$$\Phi=\left\langle [\boldsymbol{\varphi}_L(x(t+T))-\boldsymbol{\varphi}_L(x(t))],[\boldsymbol{\varphi}_L(x(t+T))-\boldsymbol{\varphi}_L(x(t))]^T \right\rangle_\Omega$$

invertible, we obtain the solution

$$\mathbf{w}_L{}^{\mu^{(i)}}=-\Phi^{-1}\left\langle [\boldsymbol{\varphi}_L(x(t+T))-\boldsymbol{\varphi}_L(x(t))], \int_t^{t+T} r(x(s),\mu^{(i)}(x(s)))ds \right\rangle_\Omega . \quad (18)$$

To show that Φ is invertible the following technical results are needed.

Lemma 2 If the set $\{\phi_j\}_1^N$ is linearly independent and $u\in\Psi(\Omega)$ then the set $\{\nabla\phi_j{}^T(f+gu)\}_1^N$ is also linearly independent.

For the proof see [2].

We now introduce a lemma proving that Φ can be inverted.

Lemma 3 Let $\mu(x)\in\Psi(\Omega)$ such that $f(x)+g(x)\mu(x)$ is asymptotically stable. If the set $\{\phi_j\}_1^N$ is linearly independent then $\exists T>0$ such that $\forall x(t)\in\Omega$ the set $\{\bar{\phi}_j(x(t),T)=\phi_j(x(t+T))-\phi_j(x(t))\}_1^N$ is also linearly independent.

The proof is by contradiction with the result in lemma 2.

Based on the result of Lemma 3, conditioned by an excitation requirement related to the selection of the sample time T, the parameters W_i of the cost function can be calculated using only online measurements of the state vector and the integrated reward over a finite time interval. The control policy is updated at time $t+T$, after observing the state $x(t+T)$ and it will be used for controlling the system during the time interval $[t+T,t+2T]$; thus the algorithm is suitable for online implementation from the control theory point of view. Figure 1 presents the structure of the system with optimal adaptive controller.

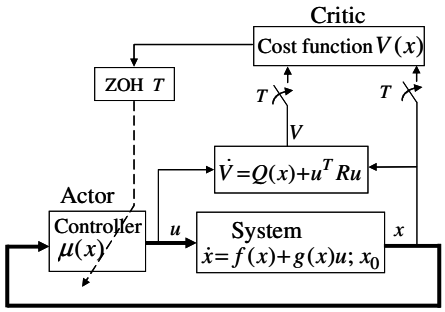

Figure 1. Structure of the system with adaptive controller

It is observed that the update of both the actor and the critic is performed at discrete moments in time. However, the control action is a full fledged continuous-time control, with its constant gain updated at discrete moments in time, since the critic update is based on the observations of the continuous-time cost over a finite sample interval. As a result, the algorithm converges to the solution of the continuous-time optimal control problem.

The flowchart of the online algorithm is presented in Fig. 1.

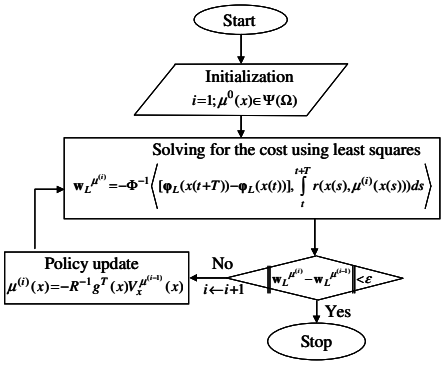

Figure 1. Flowchart of the online algorithm

Subsequent to the calculation of the solution of equation (15), given by (18), the control policy is updated according to the equation (12), written explicitly as

$$\mu^{(i+1)}(x)=-\tfrac{1}{2}R^{-1}g^T(x)(\boldsymbol{\varphi}_L)_x{}^T(x)\mathbf{w}_L{}^{\mu^{(i)}} . \quad (19)$$

V. Relation of the Proposed Algorithm with Reward-Based Learning Mechanisms in the Mammal Brain

The adaptive algorithm based on policy iteration is implemented on an actor-critic structure [18], [28]. The way in which the actor-critic structure performs continuous-time closed loop control while searching for optimal control policies points out the existence of two time scales for the mechanisms involved: a fast time scale which characterizes the continuous time control process, and a slower time scale which characterizes the learning processes at the levels of the critic and the actor.

Thus the actor and critic structures perform tasks at different operation frequencies in relation with the nature of the task to be performed. The fact that is not surprising given that the actor-critic structure was inspired by the way in which the reward based learning takes place in the mammal brain. Different oscillation frequencies are connected with the way in which different areas of the brain perform their functions of processing the information received from the sensors [15]. Low level control structures must quickly react to new information received from the environment while higher level structures slowly evaluate the results associated with the present behavior policy.

Another feature of the online policy iteration algorithm presented in this paper is related with the nature of information, *i.e.* a computed temporal difference (TD) error signal, required for the learning process to take place at the critic level. In relation to this there exist a number of reports, *e.g.* [22], [23], which argue that the dopamine signal produced by basal ganglia structures in the mammal brain encodes the TD error between the received and the expected rewards and the fact that this dopamine signal favors the learning process by increasing the synaptic plasticity of certain groups of neurons.

A third, and most distinctive, attribute of the adaptive optimal control algorithm concerns the value of the sample time used for obtaining the reward information for the Critic learning process. Lemma 3 indicates that the learning

process at the Critic level is conditioned by certain values of the reward signal sampling. Choosing the value of the sample time is generally considered to be a technical requirement of online algorithms and is related with the well known persistency of excitation requirement which grants asymptotic convergence for the learning process. It was thus even more surprising to learn that there exists in the brain a mechanism, described in [6] and verified against experimental data, which supports the existence of a variable sample time for the reward signal.

The connection between the learning mechanisms in the mammal brain and the learning structures and algorithms developed for control engineering purposes provides a strong argument in favor of a desired collaboration between the engineering fields of computational intelligence and control, and cognitive science.

VI. Conclusion

In this paper we presented a new adaptive controller based on a reinforcement learning algorithm, namely policy iteration, to solve on-line the continuous time optimal control problem without using knowledge about the system's internal dynamics. Several remarks relating the proposed algorithm with reinforcement learning mechanisms in the mammal brain have been included.

References

[1] Abu-Khalaf, M., Lewis, F.L.: Nearly Optimal Control Laws for Nonlinear Systems with Saturating Actuators Using a Neural Network HJB Approach , Automatica, 41(5), pp. 779-791, (2005).

[2] Beard, R., Saridis, G., Wen, J.: Galerkin Approximations of the Generalized Hamilton-Jacobi-Bellman Equation, Automatica, 33(12), pp. 2159-2177, (1997)

[3] Beard, R., Saridis, G., Wen, J.: Approximate Solutions to the Time-Invariant Hamilton-Jacobi-Bellman Equation, Journal of Optimization Theory and Application, 96(3), pp. 589-626, (1998)

[4] Bertsekas, D.P., Tsitsiklis, J.N.: Neuro-Dynamic Programming, Athena Scientific, MA, (1996)

[5] Bradtke, S.J., Ydestie, B.E., Barto, A.G.: Adaptive Linear Quadratic Control Using Policy Iteration, Proc. of ACC, pp. 3475-3476, Baltimore, June, (1994)

[6] Brown, J., Bullock, D., Grossberg S.: How the basal ganglia use parallel excitatory and inhibitory learning pathways to selectively respond to unexpected rewarding cues, J. Neuroscience, 19, pp. 10502-10511, (1999)

[7] Doya K.: Reinforcement Learning In Continuous Time and Space, Neural Computation, 12(1), pp. 219-245, (2000)

[8] Feldbaum, A.A.: Dual control theory I-II, Autom. Remote Control, 21, pp. 874-880, 1033-1039, (1960)

[9] Filatov, N.M., Unbehauen, H.: Survey of adaptive dual control methods, IEE Proc. Control Theory and Applications, 147(1), pp. 118-128, (2000)

[10] Hanselmann, T., Noakes, L., Zaknich, A.: Continuous-Time Adaptive Critics, IEEE Trans. on Neural Networks, 18(3), pp. 631-647, (2007)

[11] Hewer, G.: An Iterative Technique for the Computation of the Steady State Gains for the Discrete Optimal Regulator, IEEE Trans. on Automatic Control, vol. 16, pp. 382- 384, Aug. (1971)

[12] Hornik, K., Stinchcombe, M., White, H.: Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, Neural Networks, 3, pp. 551–560, (1990)

[13] Howard, R.A.: Dynamic Programming and Markov Processes, MIT Press, Cambridge, MA, (1960)

[14] Kleinman, D.: On an Iterative Technique for Riccati Equation Computations, IEEE Trans. on Automatic Control, vol. 13, pp. 114-115, Feb, (1968).

[15] Levine, D.S., Brown, V.R., Shirey V.T. eds.: Oscillations in Neural Systems, Lawrence Erlbaum Associates, (2000)

[16] Lewis, F., Syrmos, V.: Optimal Control, New York: Wiley, (1995).

[17] Li, Z.H., Krstic, M.: Optimal design of adaptive tracking controllers for nonlinear systems, Proc. of ACC, pp. 1191-1197, (1997)

[18] Miller, W.T., Sutton, R., Werbos, P.: Neural networks for control, Cambridge, Massachusetts: The MIT Press., (1990)

[19] Murray, J.J., Cox, C.J, Lendaris, G.G., and Saeks, R.: Adaptive Dynamic Programming, IEEE Trans. on Systems, Man and Cybernetics, 32(2), pp 140-153, (2002).

[20] Prokhorov, D., Wunsch, D.: Adaptive critic designs, IEEE Trans. on Neural Networks, 8(5), pp 997-1007, (1997)

[21] Saridis, G., Lee, C.S.: An Approximation Theory of Optimal Control for Trainable Manipulators, IEEE Trans. on Systems, Man and Cybernetics, 9(3), pp. 152-159, (1979)

[22] Schultz, W., Dayan P., Read Montague, P.: A Neural Substrate of Prediction and Reward, Science, 275, pp. 1593-1599, (1997)

[23] Schultz, W.: Neural coding of basic reward terms of animal learning theory, game theory, microeconomics and behavioral ecology, Current Opinion in Neurobiology, 14, pp. 139-147, (2004)

[24] Sutton, R.S., Barto, A.G.: Reinforcement Learning – An introduction, MIT Press, Cambridge MA, (1998)

[25] Vrabie, D., Pastravanu, O., Lewis, F.L.: Policy Iteration for Continuous-time Systems with Unknown Internal Dynamics, Proc. of MED, (2007)

[26] Watkins C.J.C.H.: Learning from delayed rewards. PhD Thesis, University of Cambridge, England, (1989)

[27] Werbos P.: Neural networks for control and system identification, IEEE Proc. CDC'89, (1989)

[28] Werbos P.: Approximate dynamic programming for real-time control and neural modeling, Handbook of Intelligent Control, ed. D.A. White and D.A. Sofge, New York: Van Nostrand Reinhold, (1992)

[29] Wittenmark, B.: Adaptive dual control methods: An overview, 5th IFAC Symp. on Adaptive Systems in Control and Signal Processing, pp. 67-73, (1995)