Neural Networks 22 (2009) 237–246

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2009 Special Issue

Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems

Draguna Vrabie *, Frank Lewis

Automation and Robotics Research Institute, University of Texas at Arlington, 7300 Jack Newell Blvd. S., Fort Worth, TX 76118, USA

ARTICLE INFO

Article history:
Received 11 January 2009
Received in revised form 16 March 2009
Accepted 19 March 2009

Keywords:
Direct adaptive optimal control
Policy iteration
Neural networks
Online control

ABSTRACT

In this paper we present in a continuous-time framework an online approach to direct adaptive optimal control with infinite horizon cost for nonlinear systems. The algorithm converges online to the optimal control solution without knowledge of the internal system dynamics. Closed-loop dynamic stability is guaranteed throughout. The algorithm is based on a reinforcement learning scheme, namely Policy Iterations, and makes use of neural networks, in an Actor/Critic structure, to parametrically represent the control policy and the performance of the control system. The two neural networks are trained to express the optimal controller and optimal cost function which describes the infinite horizon control performance. Convergence of the algorithm is proven under the realistic assumption that the two neural networks do not provide perfect representations for the nonlinear control and cost functions. The result is a hybrid control structure which involves a continuous-time controller and a supervisory adaptation structure which operates based on data sampled from the plant and from the continuous-time performance dynamics. Such control structure is unlike any standard form of controllers previously seen in the literature. Simulation results, obtained considering two second-order nonlinear systems, are provided.

1. Introduction

In an environment in which a number of players compete for a limited resource, optimal behavior with respect to desired long term goals leads to long term advantages. In a control engineering framework the role of the environment is played by a system to be controlled (this ranges from industrial processes such as distillation columns and power systems, to airplanes, medical equipment and mobile robots), while the controller, equipped with sensors and actuators, plays the role of the agent which is able to regulate the state of the environment such that desired performances are obtained. An intelligent controller is able to adapt its actions in front of unforeseen changes in the system dynamics. In the case in which the controller has a fixed parametric structure, the adaptation of controller behavior is equivalent to changing the values of the controller parameters. From a control engineering perspective, not every automatic control loop needs to be designed to exhibit intelligent behavior. In fact in industrial process control there exists a hierarchy of control loops which has at the lowest level the simplest and most robust regulation, which provides fast reaction in front of parametric and non-parametric disturbances without controller adaptation, while at the topmost end are placed the so-called money-making loops, whose operation close to optimality has the greatest impact on maximization of income. In the latter case the control performance is not explicitly defined in terms of desired trajectories for the states and/or outputs of the system; instead it is implicitly expressed through a functional that captures the nature of the desired performance in a more general sense. Such an optimality criterion characterizes the system's performance in terms of the control inputs and system states; it is in fact an implicit representation of a desired balance between the amount of effort invested in the control process and the resulting outputs.

Optimal control refers to a class of methods that can be used to synthesize a control policy which results in best possible behavior with respect to the prescribed criterion (i.e. control policy which leads to maximization of performance). The solutions of optimal control problems can be obtained either by using Pontryagin's minimum principle, which provides a necessary condition for optimality, or by solving the Hamilton–Jacobi–Bellman (HJB) equation, which is a sufficient condition (see e.g. Kirk (2004) and Lewis and Syrmos (1995)). Although mathematically elegant, both approaches present a major disadvantage posed by the requirement of complete knowledge of the system dynamics. In the case when only an approximate model of the system is available, the optimal controller derived with respect to the system's model will not perform optimally when applied for the control of the real process. Thus, adaptation of the controller

* Corresponding author. Tel.: +1 817 272 5938; fax: +1 817 272 5938.
E-mail addresses: dvrabie@uta.edu (D. Vrabie), lewis@uta.edu (F. Lewis).

0893-6080/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2009.03.008

parameters such that operation becomes optimal with respect to the behavior of the real plant is highly desired.

Adaptive optimal controllers have been developed either by adding optimality features to an adaptive controller (e.g. the adaptation of the controller parameters is driven by desired performance improvement reflected by an optimality criterion functional) or by adding adaptive features to an optimal controller (e.g. the optimal control policy is improved relative to the adaptation of the parameters of the model of the system). A third approach to adaptive optimal control (Sutton, Barto, & Williams, 1992), namely reinforcement learning (RL) (Sutton & Barto, 1998), was introduced and extensively developed in the computational intelligence and machine learning societies, generally to find optimal control policies for Markovian systems with discrete state and action spaces (Howard, 1960). RL-based solutions to the continuous-time optimal control problem have been given in Baird (1994) and Doya (2000). The RL algorithms are constructed on the idea that successful control decisions should be remembered, by means of a reinforcement signal, such that they become more likely to be used a second time. Although the idea originates from experimental animal learning, where it has been observed that the dopamine neurotransmitter acts as a reinforcement informational signal which favors learning at the level of the neuron (see e.g. Doya, Kimura, and Kawato (2001) and Schultz, Tremblay, and Hollerman (2000)), RL is strongly connected from a theoretical point of view with direct and indirect adaptive optimal control methods. In the present issue, Werbos (2009) reviews four generations of general-purpose learning designs in Adaptive, Approximate Dynamic Programming, which provide approximate solutions to optimal control problems and include reinforcement learning as a special case. He argues the relevance of such methods not only for the general goal of replicating human intelligence but also for bringing a solution of efficient regulation in electrical power systems.

The main advantage of using RL for solving optimal control problems comes from the fact that a number of RL algorithms, e.g. Q-learning (Watkins, 1989) (also known as Action Dependent Heuristic Dynamic Programming (Werbos, 1989, 1992)), do not require knowledge or identification/learning of the system dynamics. This is important since it is well known that modeling and identification procedures for the dynamics of a given nonlinear system are most often time consuming iterative approaches which require model design, parameter identification and model validation at each step of the iteration. The identification procedure is even more difficult when the system has hidden nonlinear dynamics which manifest only in certain operating regions. In the RL algorithms' case the learning process is moved at a higher level having no longer as an object of interest the system's dynamics but a performance index which quantifies how close to optimality the closed-loop control system operates. In other words, instead of identifying a model of the plant dynamics, that will later be used for the controller design, the RL algorithms require identification of the static map which describes the system performance associated with a given control policy. One sees now that, as long as enough information is available to describe the performance associated with a given control policy at all significant operating points of the control system, the system performance map can be easily learned, conditioned by the fact that the control system maintains stability properties. This is again advantageous compared with an open-loop identification procedure which, due to the excitatory inputs required for making the natural modes of the system visible in the measured system states, could have as a result the instability of the system.

Even in the case when complete knowledge of the system dynamics is available, a second difficulty appears from the fact that the HJB equation, underlying the optimal control problem, is generally nonlinear and most often does not possess an analytical solution; thus the optimal control solution is regularly addressed by numerical methods (Huang & Lin, 1995). Also from this point of view, RL algorithms provide a natural approach to solve the optimal control problem, as they can be implemented my means of function approximation structures, such as neural networks, which can be trained to learn the solution of the HJB equation. RL algorithms, such as the one that we will present in Section 3, and develop for online implementation in Section 4, can easily be incorporated in higher-level decision making structures of the sort presented in (Brannon, Seiffertt, Draelos, & Wunch, 2009).

RL algorithms can be implemented on Actor/Critic structures which involve two function approximators, namely the Actor, which parameterizes the control policy, and the Critic, a parametric representation for the cost function which describes the performance of the control system. In this case the solution of the optimal control problem will be provided in the form of the Actor neural network for which the associated cost, i.e. the output of the Critic neural network, has an extremal value.

In this paper we present an adaptive method, which uses neural-network-type structures in an Actor/Critic configuration, for solving online the optimal control problem for the case of nonlinear systems, in a continuous-time framework, without making use of explicit knowledge on the internal dynamics of the nonlinear system. The method is based on Policy Iteration (PI), an RL algorithm which iterates between the steps of policy evaluation and policy improvement. The PI method starts by evaluating the cost of a given admissible initial policy and then uses this information to obtain a new control policy, which is improved in the sense of having a smaller associated cost compared with the previous policy, over the domain of interest in the state space. The two steps are repeated until the policy improvement step no longer changes the present policy, this indicating that the optimal control behavior is obtained.

In the case of continuous-time systems with linear dynamics, PI was employed for finding the solution of the state feedback optimal control problem (i.e. LQR) in Murray, Cox, Lendaris, and Saeks (2002), while the convergence guarantee to the LQR solution was given in Kleinman (1968). The PI algorithm, as used by Kleinman (1968), requires repetitive solution of Lyapunov equations, which involve complete knowledge of the system dynamics (i.e. both the input-to-state and internal system dynamics specified by the plant input and system matrices). For nonlinear systems, the PI algorithm was first developed by Leake and Liu (1967). Three decades later it was introduced in Beard, Saridis, and Wen (1997) as a feasible adaptive solution to the CT optimal control problem. In Beard et al. (1997) the Generalized HJB equations (a sort of nonlinear Lyapunov equations), which appear in the PI algorithm, were solved using successive Galerkin approximation algorithms. A neural-network-based approach was developed and extended to the cases of H2 and H-infinity with constrained control in Abu-Khalaf and Lewis (2005) and Abu-Khalaf, Lewis, and Huang (2006). Neural-network-based Actor/Critic structures, in a continuous-time framework, with neural network tuning laws have been given in Hanselmann, Noakes, and Zaknich (2007). All of the above-mentioned methods require complete knowledge of the system dynamics.

In Vrabie, Pastravanu, and Lewis (2007) and Vrabie and Lewis (2008) the authors gave a new formulation of the PI algorithm for linear and nonlinear continuous-time systems. This new formulation allows online adaptation (i.e. learning) of the continuous-time operating controller to the optimal state feedback control policy, without requiring knowledge of the system internal dynamics (knowledge regarding the input-to-state dynamics is still required, but from a system identification point of view this knowledge is relatively easier to obtain).

The continuous-time online approach to RL, which we describe in Section 3, is the one introduced in Vrabie and Lewis (2008). In that paper the convergence of the algorithm was proved under the assumption that the two function approximators in the Actor/Critic structure can provide exact representations of the control and cost functions. While the development in Vrabie and Lewis (2008) showed the validity of the approach to online learning, the assumption of exact representation of the cost functions which have to be learned was not realistic. In this paper we revisit the online algorithm presented in Vrabie and Lewis (2008) and we extend those results providing convergence proofs for the function-approximator-based algorithm, taking into account the existing approximation errors between the Actor/Critic structures and the control and cost functions, respectively. The algorithm converges online to the optimal control solution without knowledge of the internal system dynamics. Closed-loop dynamic stability is guaranteed throughout.

The end result is a control and adaptation structure which is a hybrid combination between a continuous-time controller and a learning structure which operates based on discrete sampled data from the system and from the continuous-time dynamics reflecting the performance of the system. Such structure is unlike any of the standard forms of controllers appearing in the literature.

In the next section we give an overview of the optimal control problem for nonlinear systems. The proposed Policy Iteration algorithm which solves the HJB equation without requiring knowledge of the internal dynamics of the system is presented in Section 3. Convergence of the algorithm is proven by showing equivalence with the general PI algorithm for nonlinear systems. The formulation of the introduced algorithm using neural networks approximation structures is discussed in Section 4. Convergence of the neural-network-based algorithm while considering the error between the cost function and its neural network approximation is then provided. Section 5 gives a flowchart of the online algorithm and discusses the online implementation on an Actor/Critic structure, while commenting also on the relations between the proposed online algorithm and certain learning mechanisms in the mammal brain. Section 6 presents simulation results considering two nonlinear systems with quadratic and quartic cost functions.

2. Background in nonlinear optimal control

In this section we give the formulation of the nonlinear optimal control problem.

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

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

with $x(t) \in R^n$, $f(x(t)) \in R^n$, $g(x(t)) \in R^{n\times m}$ and the input $u(t) \in U \subset \mathbb{R}^m$. We assume that $f(0) = 0$, that $f(x) + g(x)u$ is Lipschitz continuous on a set $\Omega \subseteq \mathbb{R}^n$ which 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.

We note here that although global asymptotic stability is guaranteed in a linear system case, it is generally difficult to guarantee in a general continuous-time nonlinear system problem setting. This is due to the non-smooth nature of a nonlinear system dynamics; at the points in which there exist discontinuities of $\dot{x}$, there will also exist discontinuities of the gradient of the cost function. For this reason we have to restrict our discussion to the case in which asymptotic stability is desired and sought for only in a region $\Omega \subseteq \mathbb{R}^n$ in which the cost function is continuously differentiable.

The infinite horizon integral cost associated with the control input $\{u(\tau);\ \tau \geq t\}$ is defined as

$$V^u(x(t)) = \int_t^\infty r(x(\tau), u(\tau))\mathrm{d}\tau \tag{2}$$

where $x(\tau)$ denotes the solution of (1) for initial condition $x(t) \in \Omega$ and input $\{u(\tau);\ \tau \geq t\}$, $r(x, u) = Q(x) + u^{\mathrm{T}}Ru$ with $Q(x)$ positive definite, i.e. $\forall x \neq 0, Q(x) > 0$ and $x = 0 \Rightarrow Q(x) = 0$, and $R \in \mathbb{R}^{m\times m}$ a positive definite matrix.

Definition 1 (*Beard et al., 1997 (Admissible (Stabilizing) 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 $\forall x_0 \in \Omega$.

The cost function associated with any admissible control policy $\mu \in \Psi(\Omega)$ is

$$V^\mu(x(t)) = \int_t^\infty r(x(\tau), \mu(x(\tau)))\mathrm{d}\tau. \tag{3}$$

$V^\mu(x)$ is C^1. The infinitesimal version of (3) is

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

where ∇V_x^μ (a column vector) denotes the gradient of the cost function V^μ with respect to x, as the cost function does not depend explicitly on time. Eq. (4) is a Lyapunov equation for nonlinear systems which, given the controller $\mu(x) \in \Psi(\Omega)$, can be solved for the cost 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

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

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

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

This Hamiltonian, which does not depend explicitly on time and corresponds to the infinite horizon optimal control problem, must be identically zero when evaluated on an extremal trajectory. Also every admissible policy, $\mu(x)$, and associated cost, $V^\mu(x)$, with $V^\mu(0) = 0$, satisfy $H(x, \mu, V_x^\mu) = 0$. Thus this equation can be used to solve for the cost associated with any given admissible policy.

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

$$u^*(x) = -\frac{1}{2}R^{-1}g^{\mathrm{T}}(x)\nabla V_x^*. \tag{7}$$

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

$$0 = Q(x) + (\nabla V_x^*)^{\mathrm{T}}f(x) - \frac{1}{4}(\nabla V_x^*)^{\mathrm{T}}g(x)R^{-1}g^{\mathrm{T}}(x)\nabla V_x^*, \quad V^*(0) = 0. \tag{8}$$

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 cost function and then substitute the solution in (7) to obtain the optimal control. However, solving the HJB equation is generally difficult. It also requires complete knowledge of the system dynamics (i.e. the functions $f(x)$, $g(x)$ need to be known).

3. Policy iteration algorithm for solving the HJB equation

In the following we give a new online iterative algorithm which will adapt to solve the infinite horizon optimal control problem without using knowledge regarding the system internal dynamics (i.e. the system function $f(x)$). Convergence of the algorithm to the optimal control function is then provided.

3.1. Policy iteration algorithm

Let $\mu^{(0)}(x(t)) \in \Psi(\Omega)$ be an admissible policy, and $T > 0$ such that as $x(t) \in \Omega$ also $x(t+T) \in \Omega$ (the existence of such $T > 0$ is guaranteed by the admissibility of $\mu^{(0)}(.)$ on Ω), then the iteration between

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

$$V^{\mu^{(i)}}(x(t)) = \int_t^{t+T} r(x(s), \mu^{(i)}(x(s)))\mathrm{d}s + V^{\mu^{(i)}}(x(t+T))$$

$$\text{with } V^{\mu^{(i)}}(0) = 0 \tag{9}$$

and

2. (policy improvement) update the control policy using

$$\mu^{(i+1)} = \arg\min_{u \in \Psi(\Omega)}[H(x, u, \nabla V_x^{\mu^{(i)}})], \tag{10}$$

which explicitly is

$$\mu^{(i+1)}(x) = -\frac{1}{2}R^{-1}g^{\mathrm{T}}(x)\nabla V_x^{\mu^{(i)}}, \tag{11}$$

converges to the optimal control policy $\mu^* \in \Psi(\Omega)$ with corresponding cost $V^*(x_0) = \min_\mu(\int_0^\infty r(x(\tau), \mu(x(\tau)))\mathrm{d}\tau)$.

Eqs. (9) and (11) give a new formulation for the Policy Iteration algorithm which allows solving the optimal control problem without making use of any knowledge of the internal dynamics of the system, $f(x)$. This algorithm is an online version of the offline algorithms proposed in Abu-Khalaf and Lewis (2005) and Beard et al. (1997) motivated by the success of the online adaptive critic techniques proposed by computational intelligence researchers (Bertsekas and Tsitsiklis (1996); Murray et al. (2002); Prokhorov and Wunsch (1997)). In the spirit of reinforcement learning algorithms, the integral term in (9) can be addressed as the *reinforcement* over the time interval $[t, t+T)$.

Eq. (9) is a discretized version of $V^{\mu^{(i)}}(x(t)) = \int_t^\infty r(x(\tau), \mu^{(i)}(x(\tau)))\mathrm{d}\tau$ and it can be viewed as a Lyapunov equation for nonlinear systems. In this paper we shall refer to it also as

$$LE(V^{\mu^{(i)}}(x(t))) \triangleq \int_t^{t+T} r(x(s), \mu^{(i)}(x(s)))\mathrm{d}s + V^{\mu^{(i)}}(x(t+T)) - V^{\mu^{(i)}}(x(t)) \quad \text{with } V^{\mu^{(i)}}(0) = 0.$$

The convergence of the new PI algorithm is given in the next subsection. The implementation of the algorithm using neural network structures will be discussed in Section 4.

3.2. Convergence of the policy iteration algorithm

If $\mu^{(i)} \in \Psi(\Omega)$ and $V^{\mu^{(i)}}(x(t)) \in C^1(\Omega)$ satisfy Eq. (9), then one can show the new control policy $\mu^{(i+1)}$, determined based on Eq. (10), is admissible for the system (1) (for proof see Abu-Khalaf and Lewis (2005) and Beard et al. (1997)).

The following result is required in order to prove the convergence of the proposed policy iteration algorithm.

Lemma 1. *Solving for $V^{\mu^{(i)}}$ in Eq. (9) is equivalent to finding the solution of*

$$0 = r(x, \mu^{(i)}(x)) + (\nabla V_x^{\mu^{(i)}})^{\mathrm{T}}(f(x) + g(x)\mu^{(i)}(x)), \quad V^{\mu^{(i)}}(0) = 0. \tag{12}$$

The proof is presented in the Appendix.

Remark 1. Although the same solution is obtained solving either Eq. (9) or (12), solving Eq. (9) does not require any knowledge of the system dynamics $f(x)$, which in turn appears explicitly in (12).

From Lemma 1 it follows that the algorithm (9) and (11) is equivalent to iterating between (12) and (11), without using knowledge of the system internal dynamics $f(x)$.

Theorem 1 (*Convergence*). *The policy iteration (9) and (11) converges uniformly to the optimal control solution on the trajectories originating in Ω, i.e.*

$$\forall\, \varepsilon > 0\ \exists i_0 : \forall i \geq i_0 \quad \sup_{x\in\Omega}\left|V^{\mu^{(i)}}(x) - V^*(x)\right| < \varepsilon, \qquad \sup_{x\in\Omega}\left|\mu^{(i)}(x) - u^*(x)\right| < \varepsilon. \tag{13}$$

Proof. In Abu-Khalaf and Lewis (2005) and Beard et al. (1997), it was shown that iterating on Eq. (12) and (11), conditioned by an initial admissible policy $\mu^{(0)}(x)$, all the subsequent control policies will be admissible and the iteration (12) and (11) will converge to the solution of the HJB equation, i.e. Eq. (13) is satisfied.

Based on the proven equivalence between the Eqs. (9) and (12) we can conclude that the proposed online adaptive optimal control algorithm will converge to the solution of the optimal control problem (2), on Ω, without using knowledge of the internal dynamics of the controlled system (1). □

4. Neural-network-based approximation of the HJB solution

For the implementation of the iteration scheme given by (9) and (11) 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 Eq. (11). One can see that knowledge of the internal state dynamics, described by $f(x)$, is not required. The information regarding the system $f(x)$ matrix is embedded in the states $x(t)$ and $x(t+T)$ which are sampled online.

4.1. Neural network approximation of the cost function

In order to solve Eq. (9) we will use a neural-network-type structure to obtain an approximation of the cost function for any $x \in \Omega$. Due to the universal approximation property (Hornik, Stinchcombe, & White, 1990), neural networks are natural candidates to approximate smooth functions on compact sets. Thus we consider that the cost function $V^{\mu^{(i)}}(x)$ can be approximated, for $x \in \Omega$, by

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

This can be seen as a neural network with L neurons on the hidden layer and activation functions $\phi_j(x) \in C^1(\Omega)$, $\phi_j(0) = 0$. $\boldsymbol{\varphi}_L(x)$ is the vector of activation functions, $w_j^{\mu^{(i)}}$ denote the weights of the output layer and $\mathbf{w}_L^{\mu^{(i)}}$ is the weight vector. The output layer neuron has a linear activation function. The weights of the hidden layer are all equal to one and will not be changed during the training procedure.

Note that, given an infinite set of linearly independent activation functions $\{\phi_j(x)\}_1^\infty$, such that $\phi_j(x) \in C^1(\Omega)$, $\phi_j(0) = 0$, $j = \overline{1,\infty}$, which satisfy the completeness property (i.e. any function $f(x) \in C^1(\Omega), f(0) = 0$ can be represented as a linear combination of a subset of $\{\phi_j(x)\}_1^\infty$), then the exact solution of Eq. (9) can be expressed as

$$V^{\mu^{(i)}}(x) = \sum_{j=1}^{\infty} c_j^{\mu^{(i)}}\phi_j(x) = (\mathbf{c}_\infty^{\mu^{(i)}})^{\mathrm{T}}\boldsymbol{\varphi}_\infty(x), \tag{15}$$

where $\boldsymbol{\varphi}_\infty(x)$ is the vector of activation functions and $\mathbf{c}_\infty^{\mu^{(i)}}$ denotes the weight vector.

Using the neural network approximate description for the cost function, Eq. (14) and (9) can be written as

$$\mathbf{w}_L^{\mu^{(i)}T}\boldsymbol{\varphi}_L(x(t)) = \int_t^{t+T} r(x,\mu^{(i)}(x))\mathrm{d}\tau + \mathbf{w}_L^{\mu^{(i)}T}\boldsymbol{\varphi}_L(x(t+T)). \quad (16)$$

As the cost function was replaced with the neural network approximation, (16) will have the residual error

$$\delta_L^{\mu^{(i)}}(x(t),T) = \int_t^{t+T} r(x,\mu^{(i)}(x))\mathrm{d}\tau + \mathbf{w}_L^{\mu^{(i)}T}[\boldsymbol{\varphi}_L(x(t+T)) - \boldsymbol{\varphi}_L(x(t))]. \quad (17)$$

From the perspective of temporal difference learning methods (e.g., Baird (1994) and Doya (2000)) this error can be viewed as a temporal difference residual error for continuous-time systems.

To determine the parameters of the neural network approximating the cost function $V_L^{\mu^{(i)}}$, in the least-squares sense, we use the method of weighted residuals. Thus we tune the parameters $\mathbf{w}_L^{\mu^{(i)}}$ of the cost function approximation $V_L^{\mu^{(i)}}$ in order to minimize the objective

$$S = \int_\Omega \delta_L^{\mu^{(i)}}(x,T)\delta_L^{\mu^{(i)}}(x,T)\mathrm{d}x. \quad (18)$$

This amounts to $\int_\Omega \frac{\mathrm{d}\delta_L^{\mu^{(i)}}(x,T)}{\mathrm{d}\mathbf{w}_L^{\mu^{(i)}}}\delta_L^{\mu^{(i)}}(x,T)\mathrm{d}x = 0$. Using the inner product notation for the Lebesgue integral, one can write

$$\left\langle \frac{\mathrm{d}\delta_L^{\mu^{(i)}}(x,T)}{\mathrm{d}\mathbf{w}_L^{\mu^{(i)}}}, \delta_L^{i}(x,T)\right\rangle_\Omega = 0, \quad (19)$$

which is

$$\left\langle [\boldsymbol{\varphi}_L(x(t+T)) - \boldsymbol{\varphi}_L(x(t))], [\boldsymbol{\varphi}_L(x(t+T)) - \boldsymbol{\varphi}_L(x(t))]^{\mathrm{T}}\right\rangle_\Omega \mathbf{w}_L^{\mu^{(i)}} + \left\langle [\boldsymbol{\varphi}_L(x(t+T)) - \boldsymbol{\varphi}_L(x(t))], \int_t^{t+T} r(x(s),\mu^{(i)}(x(s)))\mathrm{d}s\right\rangle_\Omega = 0. \quad (20)$$

Conditioned by $\Phi = \langle [\boldsymbol{\varphi}_L(x(t+T)) - \boldsymbol{\varphi}_L(x(t))], [\boldsymbol{\varphi}_L(x(t+T)) - \boldsymbol{\varphi}_L(x(t))]^{\mathrm{T}}\rangle_\Omega$ being invertible, then we obtain

$$\mathbf{w}_L^{\mu^{(i)}} = -\Phi^{-1}\left\langle [\boldsymbol{\varphi}_L(x(t+T)) - \boldsymbol{\varphi}_L(x(t))], \int_t^{t+T} r(x(s),\mu^{(i)}(x(s)))\mathrm{d}s\right\rangle_\Omega. \quad (21)$$

To show that Φ can be inverted the following technical results are needed.

Definition 2 (*(Kolmogorov & Fomin, 1999) (Linearly independent set of functions)*). A set of functions $\{\phi_j\}_1^N$ is said to be linearly independent on a set Ω if $\sum_{j=1}^N c_j\phi_j(x) = 0$ a.e. on Ω implies that $c_1 = \cdots = c_N = 0$.

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

For proof see Beard et al. (1997).

We now give a technical 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. Given that the set $\{\phi_j\}_1^N$ is linearly independent then $\exists T > 0$ such that $\forall x(t) \in \Omega - \{0\}$, 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 presented in the Appendix section.

Based on the result of Lemma 3, there exist values of T such that Φ is invertible and the parameters $\mathbf{w}_L^{\mu^{(i)}}$ of the cost function $V_L^{\mu^{(i)}}$ can be calculated. Having solved for the cost function $V_L^{\mu^{(i)}}$ associated with the control policy $\mu^{(i)}$, the policy update step can be executed. The new control policy will thus be

$$\mu^{(i+1)}(x) = -\frac{1}{2}R^{-1}g^{\mathrm{T}}(x)\nabla\boldsymbol{\varphi}_L^{\mathrm{T}}(x)\mathbf{w}_L^{\mu^{(i)}}. \quad (22)$$

Eq. (22) gives the output of the Actor neural network. Note that in this implementation the controller (Actor) can be seen as a neural network, similar to the one approximating the Critic, which has the same weight parameters as the Critic neural network, but whose activation functions are the gradients of those in the Critic.

4.2. Convergence of $V_L^{\mu^{(i)}}(x)$ to the exact solution of the Lyapunov equation $V^{\mu^{(i)}}(x)$

We now discuss the convergence of the method of least squares when a neural network is used to approximate the cost function in Eq. (9).

Definition 3 (*Convergence in the Mean*). A sequence of Lebesgue integrable functions on a set Ω, $\{f_n(x)\} \in L_2(\Omega)$, is said to converge in the mean to $f(x) \in L_2(\Omega)$ if $\forall\varepsilon > 0$, $\exists N(\varepsilon)$ such that $\forall n > N(\varepsilon)$, $\|f_n(x) - f(x)\|_{L_2(\Omega)} < \varepsilon$, where $\|f(x)\|_{L_2(\Omega)}^2 = \langle f, f\rangle$.

Eq. (9) can be written using a linear operator A defined on the Hilbert space of continuous and differentiable functionals on Ω:

$$\overbrace{V^\mu(x(t)) - V^\mu(x(t+T))}^{AV^\mu} = \overbrace{\int_t^{t+T} r(x(s),\mu(x(s)))\mathrm{d}s}^{d(x,\mu(x),T)}. \quad (23)$$

Neural networks which are defined such that the activation functions are power series of order m are differentiable and can uniformly approximate a continuous function with all its partial derivatives, up to order m, by differentiating the series termwise. This type of series is m-uniformly dense, as shown in Lemma 4.

Lemma 4 (*Higher Order Weierstrass Approximation Theorem*). *Let $f(x) \in C^m(\Omega)$; then there exists a polynomial $P(x)$ such that it converges uniformly to $f(x)$, and all its partial derivatives up to order m converge uniformly (Hornik et al., 1990).*

The following facts hold under the stated standard conditions in optimal control.

Fact 1. The solution of (9) is positive definite. This is guaranteed when the system has stabilizable dynamics and when the performance functional satisfies zero state observability (i.e. observability of the system state through the cost function) (Van Der Schaft, 1992).

Fact 2. The system dynamics and the performance integrand $r(x(s),\mu(x(s)))$ are such that the solution of (9) is continuous and differentiable on Ω.

Fact 3. We can choose a complete set $\{\phi_j\}_1^\infty \in C^1(\Omega)$ such that the solution $V \in C^1(\Omega)$ and ∇V can be uniformly approximated by the infinite series built based on $\{\phi_j\}_1^\infty$.

Fact 4. The sequence $\{\bar{\phi}_j(x(t),T) = \phi_j(x(t+T)) - \phi_j(x(t))\}_1^\infty$ is linearly independent and complete. The linear independence results from Lemma 3, being conditioned by certain values of the sample time T. The completeness relies on the high-order Weierstrass approximation theorem.

$\forall V, \varepsilon \exists L, \mathbf{w}_L$ such that $|V_L - V| < \varepsilon$. This implies that, as $L \to \infty$, $\sup_{x\in\Omega} |AV_L - AV| \to 0 \Rightarrow \|AV_L - AV\|_{L_2(\Omega)} \to 0$, which proves the completeness of $\{\bar{\phi}_j(x(t),T) = A\phi_j\}_1^\infty$.

The first three assumptions are standard in optimal control, and we have proven the fourth herein. The next result is required.

Lemma 5. *Given a set of N linearly independent functions $\{f_j(x)\}_1^N$ defined on Ω then*

$$\left\|\boldsymbol{\alpha}_N^{\mathrm{T}}\mathbf{f}_N\right\|_{L_2(\Omega)}^2 \to 0 \Leftrightarrow \|\boldsymbol{\alpha}_N\|_{l_2}^2 \to 0. \tag{24}$$

For proof see Abu-Khalaf and Lewis (2005).

The next main result shows convergence in the mean.

Theorem 2. *Given that the Facts 1–4 hold, then approximate solutions exist for (9) using the method of least squares and they are unique for each L. In addition, as $L \to \infty$,*

R1. $\left\|LE(V_L^{\mu^{(i)}}(x)) - LE(V^{\mu^{(i)}}(x))\right\|_{L_2(\Omega)} \to 0$

R2. $\left\|V_L^{\mu^{(i)}}(x) - V^{\mu^{(i)}}(x)\right\|_{L_2(\Omega)} \to 0$

R3. $\left\|\nabla V_L^{\mu^{(i)}}(x) - \nabla V^{\mu^{(i)}}(x)\right\|_{L_2(\Omega)} \to 0$

R4. $\left\|\mu_L^{(i)}(x) - \mu^{(i)}(x)\right\|_{L_2(\Omega)} \to 0.$

Proof. The solution $V_L^{\mu^{(i)}}$ of (9) in the least-squares sense is the solution of the minimization problem

$$\left\|AV_L^{\mu^{(i)}} - d(x,\mu^{(i)},T)\right\|^2 = \min_{\mathbf{w}_L} \left\|\mathbf{w}_L^{\mathrm{T}}\bar{\boldsymbol{\varphi}}(x) - d(x,\mu^{(i)},T)\right\|^2. \tag{25}$$

The uniqueness of the solution follows directly from the linear independence of $\{\bar{\phi}_j(x(t),T)\}_1^L$ and R1 follows from the completeness of $\{\bar{\phi}_j(x(t),T) = A\phi_j\}_1^\infty$.

To show R2 we write

$$\begin{aligned} LE(V_L^{\mu^{(i)}}(x)) - LE(V^{\mu^{(i)}}(x)) &= \mathbf{w}_L^{\mathrm{T}}\bar{\boldsymbol{\varphi}}_L(x,T) - \mathbf{c}_\infty^{\mathrm{T}}\bar{\boldsymbol{\varphi}}_\infty(x,T) \\ &= \varepsilon_L(x,T), \end{aligned} \tag{26}$$

$$\begin{aligned} (\mathbf{w}_L - \mathbf{c}_L)^{\mathrm{T}}\bar{\boldsymbol{\varphi}}_L(x,T) &= \varepsilon_L(x,T) + \sum_{j=L+1}^{\infty} c_j\bar{\phi}_j(x,T) \\ &= \varepsilon_L(x,T) + e_L(x,T). \end{aligned} \tag{27}$$

$e_L(x,T)$ converges uniformly to zero due to the high-order Weierstrass approximation theorem (this implies convergence in the mean) and $\varepsilon_L(x,T)$ converges in the mean to zero due to R1. Then

$$\begin{aligned} &\left\|(\mathbf{w}_L - \mathbf{c}_L)^{\mathrm{T}}\bar{\boldsymbol{\varphi}}_L(x,T)\right\|_{L_2(\Omega)}^2 = \|\varepsilon_L(x,T) + e_L(x,T)\|_{L_2(\Omega)}^2 \\ &\quad \le 2\|\varepsilon_L(x,T)\|_{L_2(\Omega)}^2 + 2\|e_L(x,T)\|_{L_2(\Omega)}^2 \to 0. \end{aligned} \tag{28}$$

Since $\bar{\boldsymbol{\varphi}}_L(x,T)$ is linearly independent, then, based on Lemma 5, one sees that $\|(\mathbf{w}_L - \mathbf{c}_L)\|_{l_2}^2 \to 0$. As the set $\{\phi_j\}_1^L$ is linearly independent, from Lemma 5 it results that $\left\|(\mathbf{w}_L - \mathbf{c}_L)^{\mathrm{T}}\boldsymbol{\varphi}_L(x)\right\|_{L_2(\Omega)}^2 \to 0$. It thus follows that, as $L \to \infty$, $\left\|V_L^{\mu^{(i)}} - V^{\mu^{(i)}}\right\|_{L_2(\Omega)}^2 \to 0$.

Similarly, since $\left\{\frac{\mathrm{d}\phi_j}{\mathrm{d}x}\right\}_1^L$ is linearly independent, from Lemma 5 it results that $\left\|(\mathbf{w}_L - \mathbf{c}_L)^{\mathrm{T}}\nabla\boldsymbol{\varphi}_L(x)\right\|_{L_2(\Omega)}^2 \to 0$, from which it follows that $\left\|\nabla V_L^{\mu^{(i)}} - \nabla V^{\mu^{(i)}}\right\|_{L_2(\Omega)}^2 \to 0$.

R4 follows immediately from R3 given that

$$\begin{aligned} &\left\|\mu_L^{(i)}(x) - \mu^{(i)}(x)\right\|_{L_2(\Omega)}^2 \\ &= \left\|-R^{-1}g^{\mathrm{T}}(x)(\nabla V_L^{\mu^{(i-1)}}(x) - \nabla V^{\mu^{(i-1)}}(x))\right\|_{L_2(\Omega)}^2 \\ &\le \left\|-R^{-1}g^{\mathrm{T}}(x)\right\|_{L_2(\Omega)}^2 \left\|\nabla V_L^{\mu^{(i-1)}}(x) - \nabla V^{\mu^{(i-1)}}(x)\right\|_{L_2(\Omega)}^2 \\ &\to 0. \quad \square \end{aligned} \tag{29}$$

Based on the result in Theorem 2 the following stronger result of uniform convergence can be shown.

Corollary 1. *If the results from Theorem 2 hold then*

$$\sup_{x\in\Omega}\left|V_L^{\mu^{(i)}}(x) - V^{\mu^{(i)}}(x)\right| \to 0,$$

$$\sup_{x\in\Omega}\left|\nabla V_L^{\mu^{(i)}}(x) - \nabla V^{\mu^{(i)}}(x)\right| \to 0 \quad \textit{and}$$

$$\sup_{x\in\Omega}\left|\mu_L^{(i)}(x) - \mu^{(i)}(x)\right| \to 0.$$

For proof see Abu-Khalaf and Lewis (2005).

The next result shows that, given an initial admissible control policy, $\mu^{(0)}(x)$, the control policy at each step i of the Policy Iteration algorithm, with value function approximation, $\mu_L^{(i)}(x)$ is admissible provided that the number of neurons in the hidden layer is sufficiently large.

Corollary 2. *(Admissibility of $\mu_L^{(i)}(x)$) $\exists L_0$ such that $\forall L > L_0$, $\mu_L^{(i)} \in \Psi(\Omega)$.*

The proof is given in the Appendix.

Corollary 3. *$\sup_{x\in\Omega}|\mu_L^{(i)}(x) - \mu^{(i)}(x)| \to 0$ implies that $\sup_{x\in\Omega}\left|V_L^{(i)}(x) - V^{(i)}(x)\right| \to 0$.*

4.3. Convergence of the method of least squares to the solution of the HJB equation

In this section we show that the successive least-squares solution using neural networks converges to the solution of the HJB equation (8).

Theorem 3. *Under the assumptions of Theorem 2 the following is satisfied $\forall i \ge 0$:*

i. $\sup_{x\in\Omega}\left|V_L^{(i)}(x) - V^{(i)}(x)\right| \to 0$

ii. $\sup_{x\in\Omega}\left|\mu_L^{(i+1)}(x) - \mu^{(i+1)}(x)\right| \to 0$

iii. $\exists L_0 : \forall L \ge L_0 \quad \mu_L^{(i)}(x) \in \Psi(\Omega)$.

The proof is by induction and is presented in Abu-Khalaf and Lewis (2005).

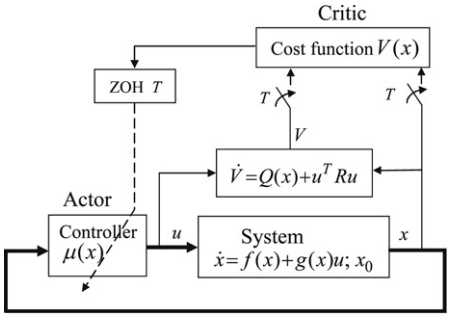

Fig. 1. Structure of the system with adaptive controller.

Theorem 4. $\forall \varepsilon \geq 0,\ \exists i_0, L_0 : \forall i \geq i_0, L \geq L_0$

i. $\sup_{x\in\Omega} \left| V_L^{(i)}(x) - V^*(x) \right| < \varepsilon,$

ii. $\sup_{x\in\Omega} \left| \mu_L^{(i+1)}(x) - u^*(x) \right| < \varepsilon,$

iii. $\mu_L^{(i)}(x) \in \Psi(\Omega).$

The proof follows directly from Theorems 1 and 3.

5. Online algorithm on an Actor/Critic structure

In this section we discuss the implementation of the adaptive algorithm on the Actor/Critic structure. We present the main features of the online adaptive critic while noting similarities with learning mechanisms in the mammal brain.

The structure of the system with the adaptive controller is presented in Fig. 1.

It is important to note that the adaptation structure has dynamics consisting of the state $V(t)$, i.e. the value, which evolves based on $\dot{V} = Q(x) + u^{\mathrm{T}}Ru$. This provides a dynamic memory that enables one to extract the information regarding the cost associated with the given policy. If one resets $V(t)$ to zero at the beginning of each sample interval $[t, t+T)$, then the measurement $V(t+T)$ gives the reinforcement over time interval $[t, t+T)$ required to implement the policy evaluation step in (9), i.e. $V(t+T)$ gives the integral reinforcement term in (9). Thus, this a dynamical controller whose memory is exactly the value $V(t)$ of using the current policy.

The policy iteration technique in this paper has led us to a control system structure that allows one to perform optimal control in an adaptive fashion online without knowing the internal dynamics of the system. We term this optimal adaptive control. This structure is not a standard one in the control systems literature. It is a hybrid continuous-time/discrete-time adaptive control structure which has continuous-time dynamics, and a discrete-time sampled data portion for policy evaluation.

The algorithm is suitable for online implementation from the control theory point of view since the control policy $\mu_L^{(i+1)}(x)$, updated at time t_{i+1} after observing the state $x(t_{i+1})$, will be used for controlling the system during the time interval $(t_{i+1}, t_{i+1}+T]$.

The flowchart of the online algorithm is presented in Fig. 2.

All the calculations involved are performed at a supervisory level which operates based on discrete-time data measured from the system. This high-level intelligent control structure implements the Policy Iteration algorithm and uses the Critic neural network to parameterize the performance of the continuous-time control system associated with a certain control policy. The high-level supervisory structure makes the decisions relative to the discrete-time moments at which both the Actor and the Critic parameters will be updated. The Actor neural network is part of the

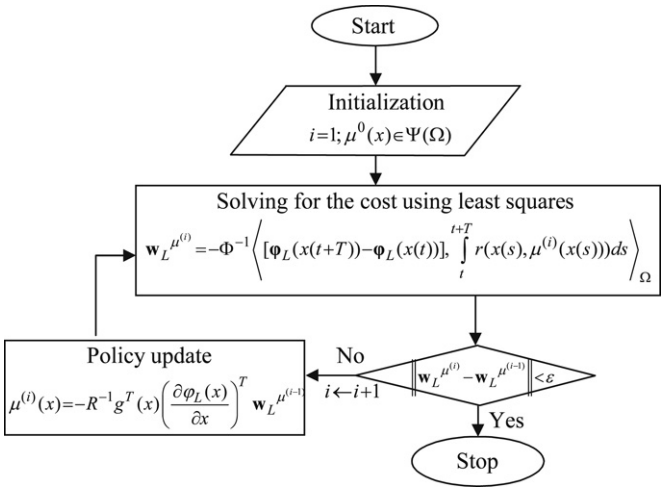

Fig. 2. Flowchart of the online algorithm.

control system structure and performs continuous-time control, while its constant gain is updated at discrete moments in time. The algorithm converges to the solution of the continuous-time optimal control problem, as proved in Section 4, since the Critic update is based on the observations of the continuous-time cost over a finite sample interval. The net result is a continuous-time controller incorporated in a continuous-time/discrete-time adaptive structure, which includes the continuous-time dynamics of the cost function and operates based on sampled data, to perform the policy evaluation and policy update steps at discrete moments in time.

It is interesting to note the rough similarity between the above-mentioned adaptive controller structure and learning mechanisms in the mammal brain. The Critic structure learns, in an episodic manner and based on samples of the reward signal from the environment, the parameters of a function which describes the Actor performance. Once a performance evaluation episode is completed, the Critic passes this information to the Actor structure which will use it to adapt for improved performance. At all times the Actor must perform continuous-time control for the system (the environment in which optimal behavior is sought). This description of the way in which the Actor/Critic structure works while searching for continuous-time 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 (i.e. learning or control). Evidence regarding the oscillatory behavior naturally characterizing biological neural systems is presented in a comprehensive manner in Levine, Brown, and Shirey (2000). 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. 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.

In Section 4 it was shown that having little information about the system states measured from the sensors, x, and the augmented system state, i.e. V, extracted from the system only at specific time values (i.e. $x(t)$, $x(t+T)$ and $V(t+T) - V(t)$), the Critic is able to

evaluate the infinite horizon continuous-time performance of the system associated with a given control policy described in terms of the Actor parameters. The Critic learns the cost function associated with a certain control behavior based on a computed temporal difference (TD) error signal, given by $V(t+T)-V(t)$.

It is interesting to mention here that in a number of reports, e.g. Schultz (2004) and Schultz, Dayan and Montague (1997), it is argued that the temporal difference error between the received and the expected rewards is in physically encoded in the dopamine signal produced by basal ganglia structures in the mammal brain. At the same time, it is known that the dopamine signal encoding the temporal error difference favors the learning process by increasing the synaptic plasticity of certain groups of neurons. We also note an interesting point presented in this issue (Perlovsky, 2009) associating certain emotions with the need for cognition, i.e. emotions play the role of reinforcement signals which drive the need for cognition. This kind of reinforcement learning is located at a higher level than the dopamine driven learning, this suggesting that there exists a hierarchy of reinforcement-based learning mechanisms in the mammal brain.

The cost function solution, given by (21), can be obtained in real time, after a sufficient number of data points are collected along state trajectories in the region of interest Ω. In practice, the matrix inversion in (21) is not performed, the solution of the equation being obtained using algorithms that involve techniques such as Gaussian elimination, backsubstitution, and Householder reflections. One should note that the least-squares method for finding the parameters of the cost function can be replaced with any other suitable, recursive or not recursive, method of parameter identification.

The iterations will be stopped (i.e. the Critic will stop updating the control policy) when the error between the system performance evaluated at two consecutive steps will cross below a designer specified threshold. Also, when this error becomes bigger than the above-mentioned threshold, indicating a change in the system dynamics, the Critic will take again the decision to start tuning the Actor parameters.

We note again that there is no required knowledge about the system dynamics $f(x)$ for the evaluation of the cost or the update of the control policy. However, knowledge on the $g(x)$ function is required for the update of the control policy, using (22), and this makes the online tuning algorithm only partially model free.

The next section presents simulation results which were obtained considering two second-order nonlinear systems.

6. Simulation examples

In this section we illustrate the results of the adaptive optimal control algorithm considering two nonlinear systems for which the optimal cost function and optimal controller are known. The nonlinear system examples were developed using the converse HJB approach, (Nevistic & Primbs, 1996), which allows construction of nonlinear systems, specified initially in a general form, starting from the known optimal cost function. In effect it solves conversely the HJB equation, given the optimal cost function, for the dynamics of the nonlinear system.

6.1. *Example* 1

We consider the nonlinear system given by the equations

$$\begin{cases} \dot{x}_1 = -x_1 + x_2 \\ \dot{x}_2 = f(x) + g(x)u \end{cases} \tag{30}$$

with $f(x) = -\frac{1}{2}(x_1 + x_2) + \frac{1}{2}x_2\sin^2(x_1)$, $g(x) = \sin(x_1)$.

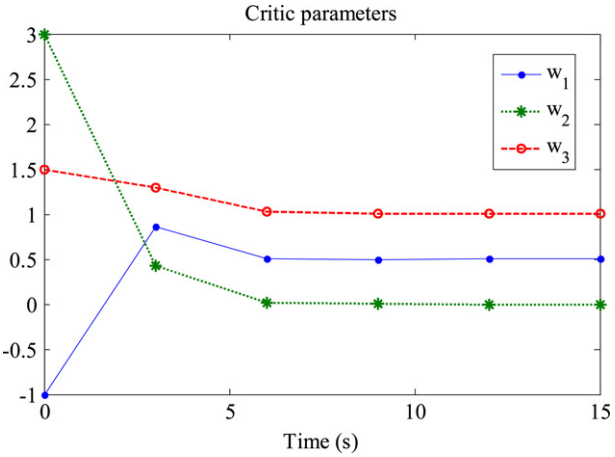

Fig. 3. Convergence of the Critic parameters.

If we define the infinite horizon cost function to be minimized as $V^u(x(t)) = \int_t^\infty (Q(x) + u^2)\mathrm{d}\tau$ with $Q(x) = x_1^2 + x_2^2$, then the optimal cost function for this system is $V^*(x) = \frac{1}{2}x_1^2 + x_2^2$ and the optimal controller is $u^*(x) = -\sin(x_1)x_2$.

The simulation was conducted using data obtained from the system at every 0.1 s. We note here that the value of this sample time is not relevant for the cost function identification procedure. In fact data does not have to be measured with a fixed sample time, as long as it is suitable for learning (i.e. carries new information on the cost function to be identified). In this sense, as long as the measured signals did not reach steady state values, meaning that the measured data is not redundant, the sampling could be executed as quickly as the hardware permits it. At the same time, as we used a batch method for identifying the cost function parameters in the least-squares sense a larger number of samples will lead to better approximation of the cost function. However, this batch procedure can be replaced with a recursive one, or a recursive procedure on time windows, such that the parameters of the cost function will be adapted over time as more data is acquired.

For the purpose of demonstrating the algorithm the initial state of the system is taken to be different than zero. For each iteration we considered data measured along five trajectories defined by five different initial conditions chosen randomly in $\Omega = \{-1 \le x_i \le 1;\ i = 1, 2\}$. The initial stabilizing controller was taken as $\mu^{(0)}(x) = -\frac{3}{2}\sin(x_1)(x_1 + x_2)$. The cost function $V^{\mu^{(i)}}(x)$ was approximated by the following smooth function, for $x \in \Omega$, $V_L^{\mu^{(i)}}(x) = (\mathbf{w}_L^{\mu^{(i)}})^{\mathrm{T}}\boldsymbol{\varphi}_L(x)$ with $L = 3$, $\mathbf{w}_3^{\mu^{(i)}} = \left[w_1^{\mu^{(i)}} \quad w_2^{\mu^{(i)}} \quad w_3^{\mu^{(i)}}\right]^{\mathrm{T}}$ and $\boldsymbol{\varphi}_3(x) = \left[x_1^2 \quad x_1x_2 \quad x_2^2\right]^{\mathrm{T}}$.

In order to solve online for the neural network weights $\mathbf{w}_3^{\mu^{(i)}}$ which parameterize the cost function, at each iteration step we set up a least-squares problem with the solution given by (21). At each iteration step we solved for $\mathbf{w}_3^{\mu^{(i)}}$ using 30 data points consisting of the measured the cost function associated with a given control policy over 30 time intervals $T = 0.1$ s, the initial state and the system state at the end of each time interval, six points measured over each of the five trajectories in the state space. In this way, every 3 s, the cost function was solved for and a policy update was performed. The result of applying the algorithm is presented in Fig. 3.

One can see from the figure that the parameters of the Critic converged to the coefficients of the optimal cost function $V^*(x) = \frac{1}{2}x_1^2 + x_2^2$, i.e. $\mathbf{w}_3^{u^*} = \left[0.5 \quad 0 \quad 1\right]^{\mathrm{T}}$.

6.2. Example 2

In this example we present the results obtained for a system which has stronger nonlinearities and quartic cost. We consider the nonlinear system given by the equations

$$\begin{cases} \dot{x}_1 = -x_1 + x_2 + 2x_2^3 \\ \dot{x}_2 = f(x) + g(x)u \end{cases} \tag{31}$$

with $f(x) = -\frac{1}{2}(x_1 + x_2) + \frac{1}{2}x_2(1 + 2x_2^2)\sin^2(x_1)$, $g(x) = \sin(x_1)$. If we define $Q(x) = x_1^2 + x_2^2 + 2x_2^4$ the infinite horizon cost function to be minimized then the optimal cost function for this system is $V^*(x) = \frac{1}{2}x_1^2 + x_2^2 + x_2^4$ and the optimal controller is $u^*(x) = -\sin(x_1)(x_2 + 2x_2^3)$.

The simulation was conducted using data obtained from the system at every 0.1 s. For each iteration we considered data measured along five trajectories defined by five different initial conditions chosen randomly in $\Omega = \{-1 \le x_i \le 1;\ i = 1, 2\}$. The initial stabilizing controller was taken as $\mu^{(0)}(x) = -\frac{1}{2}\sin(x_1)(3x_2 - 0.2x_1^2x_2 + 12x_2^3)$. The cost function $V^{\mu^{(i)}}(x)$ was approximated on Ω as $V_L^{\mu^{(i)}}(x) = (\mathbf{w}_L^{\mu^{(i)}})^{\mathrm{T}}\boldsymbol{\varphi}_L(x)$ with $L = 8$, $\mathbf{w}_8^{\mu^{(i)}} = \left[w_1^{\mu^{(i)}} \ \dots \ w_8^{\mu^{(i)}}\right]^{\mathrm{T}}$ and $\boldsymbol{\varphi}_8(x) = \left[x_1^2 \ \ x_1x_2 \ \ x_2^2 \ \ x_1^4 \ \ x_1^3x_2 \ \ x_1^2x_2^2 \ \ x_1x_2^3 \ \ x_2^4\right]^{\mathrm{T}}$.

At each iteration step we solved for $\mathbf{w}_8^{\mu^{(i)}}$ using 40 data points, i.e. eight points measured on each of the five trajectories in Ω. Each data point consists of the measured the cost function associated with the present control policy, over a time interval $T = 0.1$ s, the system state at the ends of this interval. In this way, at every 4 s, the cost function was solved for and a policy update was performed. One notes that each data point set measured on each trajectory is sufficient to identify the parameters of the cost function corresponding to that given trajectory. However, it is often not the case that cost function parameters associated with one trajectory are equal to the cost function parameters associated with another trajectory. The result of applying the algorithm is presented in Fig. 4.

The figure clearly shows that the parameters of the Critic neural network converged to the coefficients of the optimal cost function $V^*(x) = \frac{1}{2}x_1^2 + x_2^2 + x_2^4$, i.e. $\mathbf{w}_8^{u^*} = \left[0.5 \ \ 0 \ \ 1 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 1\right]^{\mathrm{T}}$. One observes that after three iteration steps the parameters of the controller, obtained based on the update Eq. (22), are very close to the parameters of the optimal controller $u^*(x) = -\sin(x_1)(x_2 + 2x_2^3)$.

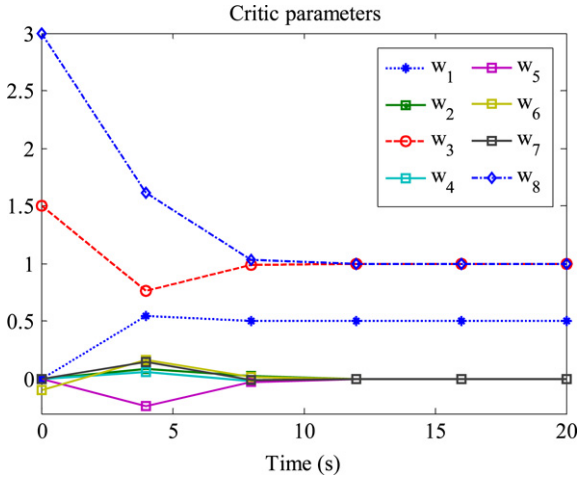

Fig. 4. Convergence of the Critic parameters.

7. Conclusions

In this paper we have presented a continuous-time adaptive controller, based on Policy Iteration, which adapts online to learn the continuous-time optimal control policy without using knowledge of the internal dynamics of the nonlinear system. Convergence of the proposed algorithm, under the condition of initial stabilizing controller, to the solution of the optimal control problem has been established. This paper also presents proofs of convergence for the online neural-network-based version of the algorithm while taking into account the neural network approximation error. Simulation results support the effectiveness of the online adaptive optimal controller.

Acknowledgements

This work was supported by the National Science Foundation ECCS-0801330 and the Army Research Office W91NF-05-1-0314.

Appendix

Proof of Lemma 1. Since $\mu^{(i)} \in \Psi(\Omega)$ then $V^{\mu^{(i)}} \in C^1(\Omega)$, defined as $V^{\mu^{(i)}}(x(t)) = \int_t^\infty r(x(s), \mu^{(i)}(x(s)))\mathrm{d}s$, is a Lyapunov function for the system $\dot{x}(t) = f(x(t)) + g(x(t))\mu^{(i)}(x(t))$. $V^{\mu^{(i)}} \in C^1(\Omega)$ satisfies

$$(\nabla V_x^{\mu^{(i)}})^{\mathrm{T}}(f(x) + g(x)\mu^{(i)}(x)) = -r(x(t), \mu^{(i)}(x(t))) \tag{32}$$

with $r(x(t), \mu^{(i)}(x(t))) > 0;\ x(t) \neq 0$. Integrating (32) over the time interval $[t, t + T]$, one obtains

$$V^{\mu^{(i)}}(x(t)) = \int_t^{t+T} r(x(s), \mu^{(i)}(x(s)))\mathrm{d}s + V^{\mu^{(i)}}(x(t + T)). \tag{33}$$

This means that the unique solution of (12), $V^{\mu^{(i)}}$, also satisfies (33). To complete the proof we have to show that Eq. (33) has a unique solution. The proof is by contradiction.

Thus, assume that there exists another cost function $V \in C^1(\Omega)$ which satisfies (14) with the end condition $V(0) = 0$. This cost function also satisfies $\dot{V}(x(t)) = -r(x(t), \mu^{(i)}(x(t)))$. Subtracting this from (33) we obtain

$$\left(\frac{\mathrm{d}[V(x(t)) - V^{\mu^{(i)}}(x(t))]}{\mathrm{d}x}^{\mathrm{T}}\right)\dot{x} = \left(\frac{\mathrm{d}[V(x(t)) - V^{\mu^{(i)}}(x(t))]}{\mathrm{d}x}^{\mathrm{T}}\right) \times (f(x(t)) + g(x(t))\mu^{(i)}(x(t))) = 0, \tag{34}$$

which must hold for any x on the system trajectories generated by the stabilizing policy $\mu^{(i)}$. Thus $V(x(t)) = V^{\mu^{(i)}}(x(t)) + c$. As this relation must hold also for $x(t) = 0$ then $V(0) = V^{\mu^{(i)}}(0) + c \Rightarrow 0 = c$ and thus $V(x(t)) = V^{\mu^{(i)}}(x(t))$, i.e. Eq. (9) has a unique solution which is equal to the unique solution of (12). □

Proof of Lemma 3. The proof is by contradiction.

The vector field $\dot{x} = f(x) + g(x)\mu(x)$ is asymptotically stable. Denote with $\eta(\tau; x(t), \mu)$, $x(t) \in \Omega$ the system trajectories obtained using the policy $\mu(x)$ for any $x(t) \in \Omega$. Then along the system trajectories we have that

$$\phi(x(t + T)) - \phi(x(t)) = \int_t^{t+T} \nabla\phi_x^{\mathrm{T}}(f + g\mu)(\eta(\tau; x(t), \mu))\mathrm{d}\tau. \tag{35}$$