Automatica 45 (2009) 477–484

Contents lists available at ScienceDirect

Automatica

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

Brief paper

# Adaptive optimal control for continuous-time linear systems based on policy iteration☆

D. Vrabie [a,*], O. Pastravanu [b], M. Abu-Khalaf [c], F.L. Lewis [a]

[a] Automation and Robotics Research Institute, The University of Texas at Arlington, 7300 Jack Newell Blvd. S., Ft. Worth, TX 76118, USA

[b] Technical University "Gh. Asachi" – Automatic Control Department, Blvd. D. Mangeron 53A, 700050 Iasi, Romania

[c] The Mathworks Inc., 3 Apple Hill Drive, Natick, MA 01760, USA

ARTICLE INFO

*Article history:*
Received 1 May 2007
Received in revised form
3 April 2008
Accepted 6 August 2008
Available online 27 December 2008

*Keywords:*
Adaptive critics
Adaptive control
Policy iterations
LQR

ABSTRACT

In this paper we propose a new scheme based on adaptive critics for finding online the state feedback, infinite horizon, optimal control solution of linear continuous-time systems using only partial knowledge regarding the system dynamics. In other words, the algorithm solves online an algebraic Riccati equation without knowing the internal dynamics model of the system. Being based on a policy iteration technique, the algorithm alternates between the policy evaluation and policy update steps until an update of the control policy will no longer improve the system performance. The result is a direct adaptive control algorithm which converges to the optimal control solution without using an explicit, a priori obtained, model of the system internal dynamics. The effectiveness of the algorithm is shown while finding the optimal-load-frequency controller for a power system.



## 1. Introduction

In this paper is presented a new, partially model-free, algorithm based on policy iterations for solving online the optimal control problem for continuous-time, linear, time-invariant systems. It is well known that solving this problem is equivalent to finding the unique positive definite solution of the underlying Algebraic Riccati Equation (ARE). Considerable effort has been made to solve the ARE and the following approaches have been proposed and extended: backwards integration of the Differential Riccati Equation, or Chandrasekhar equations (Kailath, 1973); eigenvector-based algorithms (MacFarlane, 1963; Potter, 1966) and the numerically advantageous Schur-vector-based modification (Laub, 1979); matrix-sign-based algorithms (Balzer, 1980; Byers, 1987; Hasan, Yang, & Hasan, 1999); Newton's method (Banks & Ito, 1991; Guo & Lancaster, 1998; Kleinman, 1968; Moris & Navasca, 2006).

All of these methods and their numerically advantageous variants are offline procedures which have been proven to converge to the desired solution of the ARE. They either operate on the Hamiltonian matrix associated with the ARE (eigenvector and matrix-sign-based algorithms) or require solving Lyapunov equations (Newton's method). In all cases a model of the system is required and a preceding identification procedure is always necessary. Furthermore, even if a model is available the state-feedback controller obtained based on it will only be optimal for the model approximation of the real system dynamics.

The class of techniques called adaptive control (*e.g.* see Ioannou & Fidan, 2006) was developed in order to deal with the problem of designing controllers for systems with unknown or uncertain parameter models (*e.g.* systems for which parameters can drift slowly over time). Unfortunately adaptive control is not optimal in a formal sense, only minimizing a cost function of the output error. Indirect methods have been developed, which require system identification and then the Riccati equation solution.

From a different perspective, adaptive inverse optimization methods, extensively developed for nonlinear control (*e.g.* Freeman & Kokotovic, 1996; Krstic & Deng, 1998; Li & Krstic, 1997), solve for control strategies that optimize a performance index without directly solving the underlying Riccati equation. This methodology restricts the choice of the performance index, which can no longer be freely specified by the designer, while at the same time requires knowledge of a stabilizing control law.

For the purpose of obtaining optimal controllers that minimize a given cost function without making use of a model of the system

☆ This paper was not presented at any IFAC meeting. The material in this paper was partially presented at The 15th Mediterranean Conference on Control and Automation — MED'07 Athens, Greece, June 27–29, 2007. This paper was recommended for publication in revised form by Associate Editor Shuzhi Sam Ge under the direction of Editor Miroslav Krstic.

* Corresponding author. Tel.: +1 817 272 5938; fax: +1 817 272 5938.
*E-mail addresses:* dvrabie@uta.edu (D. Vrabie), opastrav@ac.tuiasi.ro (O. Pastravanu), Murad.Abu-Khalaf@mathworks.com (M. Abu-Khalaf), lewis@uta.edu (F.L. Lewis).

0005-1098/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2008.08.017

to be controlled, a class of reinforcement learning techniques, namely adaptive critics, was developed in the computational intelligence community (Sutton, Barto, & Williams, 1991). These are in effect adaptive control techniques which sequentially update the controller parameters based on a scalar reinforcement signal which measures the controller's performance. These algorithms provide an alternative to solving the optimal control problem by approximately solving Bellman's equation for the optimal cost, and then computing the optimal control policy (*i.e.* the feedback gain for linear systems). Compared with adaptive control, the learning process does not take place at the controller tuning level alone but a new adaptive structure was introduced to learn cost functions like the ones specified in optimal control framework.

Within this body of work, the technique called policy iteration, first formulated in the framework of stochastic decision theory (Howard, 1960), describes the class of algorithms consisting of a two-step iteration: policy evaluation and policy improvement. The policy iteration technique has been extensively studied and employed for finding the optimal control solution for Markov decision problems of all sorts, White and Sofge (1992) and Bertsekas and Tsitsiklis (1996) giving a comprehensive overview of the research status in this field.

For state feedback control of continuous state linear systems the research effort has been mainly focused on the discrete-time type of formulation. Bradtke, Ydestie, and Barto (1994) developed a policy iterations algorithm that converges to the optimal solution of the discrete-time LQR problem using Q-functions. They gave a proof of convergence for Q-learning policy iteration for discrete-time systems, which, by virtue of using the so called Q-functions (Watkins, 1989; Werbos, 1989), does not require any knowledge of the system dynamics. The recursive algorithm requires initialization with a stabilizing controller, the controller remaining stabilizing at every step of the iteration. In the recent works Al-Tamimi, Abu-Khalaf, and Lewis (2007) and Landelius (1997) there have been introduced policy iteration algorithms which converge to the discrete time $H_2$ and H-infinity optimal state feedback control solution without the requirement of a stabilizing controller at each iteration step.

A first reinforcement learning attempt to determine optimal controllers for continuous-time discrete-state systems was the advantage updating algorithm (Baird, 1994) which adapts discrete-time reinforcement learning techniques to the case when the sampling time goes to zero. For continuous-time and continuous-state linear systems, in Murray, Cox, Lendaris, and Saeks (2002) were presented two policy iterations algorithms, mathematically equivalent to the Newton's method, which avoid the necessity of knowing the internal system dynamics either by evaluating the infinite horizon cost associated with a control policy along the entire stable state trajectory, or by using measurements of the state derivatives to form the Lyapunov equations. Offline, model dependent, policy iteration algorithms have also been developed to solve the Hamilton–Jacobi–Bellman and Hamilton–Jacobi–Isaacs equations associated with the continuous-time nonlinear optimal control problem in Abu-Khalaf, Lewis, and Huang (2006) and Abu-Khalaf and Lewis (2005).

In this paper we propose a new policy iteration technique that will solve in an online fashion, along a single state trajectory, the LQR problem for continuous-time systems using only partial knowledge about the system dynamics (*i.e.* the internal dynamics of the system need not be known) and without requiring measurements of the state derivative. This is in effect a direct (no identification procedure is employed) adaptive control scheme for partially unknown linear systems that converges to the optimal control solution. It will be shown that the new adaptive critics based control scheme is in fact a dynamic controller with the state given by the cost or value function.

The continuous-time policy iteration formulation for linear time-invariant systems is given in Section 2. Equivalence with iterating on underlying Lyapunov equations is proven. It is shown that the policy iteration is in fact a Newton method for solving the Riccati equation thus convergence to the optimal control is established. In Section 3 we develop the online algorithm that implements the policy iteration scheme, without knowing the plant matrix, in order to find the optimal controller. To demonstrate the capabilities of the proposed policy iteration scheme we present the simulation results of applying the algorithm to find the optimal-load-frequency controller for a power plant (Wang, Zhou, & Wen, 1993).

## 2. Continuous-time adaptive critic solution for the infinite horizon optimal control problem

In this section we develop the policy iteration algorithm, with the purpose of solving online the LQR problem without using knowledge regarding the system internal dynamics.

Consider the linear time-invariant dynamical system described by

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{1}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $(A, B)$ is stabilizable, subject to the optimal control problem

$$u^*(t) = \arg \min_{\substack{u(t) \\ t_0 \le t \le \infty}} V(t_0, x(t_0), u(t)) \tag{2}$$

where the infinite horizon quadratic cost function to be minimized is expressed as

$$V(x(t_0), t_0) = \int_{t_0}^{\infty} (x^{\mathrm{T}}(\tau)Qx(\tau) + u^{\mathrm{T}}(\tau)Ru(\tau))\mathrm{d}\tau \tag{3}$$

with $Q \ge 0$, $R > 0$ and $(Q^{1/2}, A)$ detectable.

The solution of this optimal control problem, determined by Bellman's optimality principle (Lewis & Syrmos, 1995), is given by $u(t) = -Kx(t)$ with

$$K = R^{-1}B^{\mathrm{T}}P \tag{4}$$

where the matrix $P$ is the unique positive definite solution of the Algebraic Riccati Equation (ARE)

$$A^{\mathrm{T}}P + PA - PBR^{-1}B^{\mathrm{T}}P + Q = 0. \tag{5}$$

Under the detectability condition for $(Q^{1/2}, A)$ the unique positive semidefinite solution of the ARE determines a stabilizing closed loop controller given by (4). It is important to note that, in order to solve Eq. (5), complete knowledge of the model of the system is needed, *i.e.* both the system matrix $A$ and control input matrix $B$ must be known. For this reason, developing algorithms that will converge to the solution of the optimization problem without performing prior system identification and using explicit models of the system dynamics is of particular interest from the control systems point of view.

In the following we propose a new policy iteration algorithm that will solve online for the optimal control gain, the solution of the LQR problem, without using knowledge regarding the system internal dynamics (*i.e.* the system matrix $A$). The result will in fact be an adaptive controller which converges to the state feedback optimal controller. The algorithm is based on an Actor/Critic structure and consists in a two-step iteration namely the Critic update and the Actor update. The update of the Critic structure results in calculating the infinite horizon cost associated with the use of a given stabilizing controller. The Actor parameters (*i.e.* the controller feedback gain) are then updated in the sense of reducing the cost compared to the present control policy. The derivation of the algorithm is given in Section 2.1. An analysis is done and proof of convergence is provided in Section 2.2.

### 2.1. Policy iteration algorithm

Let $K$ be a stabilizing gain for (1), under the assumption that $(A, B)$ is stabilizable, such that $\dot{x} = (A-BK)x$ is a stable closed loop system. Then the corresponding infinite horizon quadratic cost is given by

$$V(x(t)) = \int_t^\infty x^{\mathrm{T}}(\tau)(Q + K^{\mathrm{T}}RK)x(\tau)\mathrm{d}\tau = x^{\mathrm{T}}(t)Px(t) \tag{6}$$

where $P$ is the real symmetric positive definite solution of the Lyapunov matrix equation

$$(A - BK)^{\mathrm{T}}P + P(A - BK) = -(K^{\mathrm{T}}RK + Q) \tag{7}$$

and $V(x(t))$ serves as a Lyapunov function for (1) with controller gain $K$. The cost function (6) can be written as

$$V(x(t)) = \int_t^{t+T} x^{\mathrm{T}}(\tau)(Q + K^{\mathrm{T}}RK)x(\tau)\mathrm{d}\tau + V(x(t+T)). \tag{8}$$

Based on (8), denoting $x(t)$ with $x_t$, with the parameterization $V(x_t) = x_t^{\mathrm{T}}Px_t$ and considering an initial stabilizing control gain $K_1$, the following policy iteration scheme can be implemented online:

$$x_t^{\mathrm{T}}P_ix_t = \int_t^{t+T} x_\tau^{\mathrm{T}}(Q + K_i^{\mathrm{T}}RK_i)x_\tau\mathrm{d}\tau + x_{t+T}^{\mathrm{T}}P_ix_{t+T} \tag{9}$$

$$K_{i+1} = R^{-1}B^{\mathrm{T}}P_i. \tag{10}$$

Eqs. (9) and (10) formulate a new policy iteration algorithm motivated by the work of Murray et al. (2002). Note that implementing this algorithm does not involve the plant matrix $A$.

### 2.2. Proof of convergence

The next results will establish the convergence of the proposed algorithm.

**Lemma 1.** *Assuming that $A_i = A - BK_i$ is stable, solving for $P_i$ in Eq. (9) is equivalent to finding the solution of the underlying Lyapunov equation*

$$A_i^{\mathrm{T}}P_i + P_iA_i = -(K_i^{\mathrm{T}}RK_i + Q). \tag{11}$$

**Proof.** Since $A_i$ is a stable matrix and $K_i^{\mathrm{T}}RK_i + Q > 0$ then there exists a unique solution of the Lyapunov equation (12), $P_i > 0$. Also, since $V_i(x_t) = x_t^{\mathrm{T}}P_ix_t$, $\forall x_t$, is a Lyapunov function for the system $\dot{x} = A_ix$ and

$$\frac{\mathrm{d}(x_t^{\mathrm{T}}P_ix_t)}{\mathrm{d}t} = x_t^{\mathrm{T}}(A_i^{\mathrm{T}}P_i + P_iA_i)x_t = -x_t^{\mathrm{T}}(K_i^{\mathrm{T}}RK_i + Q)x_t \tag{12}$$

then, $\forall T > 0$, the unique solution of the Lyapunov equation satisfies

$$\begin{aligned}\int_t^{t+T} x_\tau^{\mathrm{T}}(Q + K_i^{\mathrm{T}}RK_i)x_\tau\mathrm{d}\tau &= -\int_t^{t+T}\frac{\mathrm{d}(x_\tau^{\mathrm{T}}P_ix_\tau)}{\mathrm{d}\tau}\mathrm{d}\tau \\ &= x_t^{\mathrm{T}}P_ix_t - x_{t+T}^{\mathrm{T}}P_ix_{t+T}\end{aligned} \tag{13}$$

*i.e.*, Eq. (9). That is, provided that $A_i$ is asymptotically stable, the solution of (9) is the unique solution of (11). □

**Remark 1.** Although the same solution is obtained whether solving the Eq. (11) or (9), Eq. (9) can be solved without using any knowledge on the system matrix $A$.

From Lemma 1 it follows that the algorithm (9) and (10) is equivalent to iterating between (11) and (10), without using knowledge of the system internal dynamics, if $A_i$ is stable at each iteration.

**Lemma 2.** *Assuming that the control policy $K_i$ is stabilizing with $V_i(x_t) = x_t^{\mathrm{T}}P_ix_t$ the cost associated with it, if (10) is used for updating the control policy then the new control policy will be stabilizing.*

**Proof.** Take the positive definite cost function $V_i(x_t)$ as a Lyapunov function candidate for the state trajectories generated while using the controller $K_{i+1}$. Taking the derivative of $V_i(x_t)$ along the trajectories generated by $K_{i+1}$ one obtains

$$\begin{aligned}\dot{V}_i(x_t) &= x_t^{\mathrm{T}}[P_i(A - BK_{i+1}) + (A - BK_{i+1})^{\mathrm{T}}P_i]x_t \\ &= x_t^{\mathrm{T}}[P_i(A - BK_i) + (A - BK_i)^{\mathrm{T}}P_i]x_t \\ &\quad + x_t^{\mathrm{T}}[P_iB(K_i - K_{i+1}) + (K_i - K_{i+1})^{\mathrm{T}}B^{\mathrm{T}}P_i]x_t.\end{aligned} \tag{14}$$

The second term, using the update given by (10) and completing the squares, can be written as

$$\begin{aligned}&x_t^{\mathrm{T}}[K_{i+1}^{\mathrm{T}}R(K_i - K_{i+1}) + (K_i - K_{i+1})^{\mathrm{T}}RK_{i+1}]x_t \\ &= x_t^{\mathrm{T}}[-(K_i - K_{i+1})^{\mathrm{T}}R(K_i - K_{i+1}) - K_{i+1}^{\mathrm{T}}RK_{i+1} + K_i^{\mathrm{T}}RK_i]x_t.\end{aligned}$$

Using (11) the first term in the equation can be written as $-x_t^{\mathrm{T}}[K_i^{\mathrm{T}}RK_i + Q]x_t$ and summing up the two terms one obtains

$$\begin{aligned}\dot{V}_i(x_t) &= -x_t^{\mathrm{T}}[(K_i - K_{i+1})^{\mathrm{T}}R(K_i - K_{i+1})]x_t \\ &\quad - x_t^{\mathrm{T}}[Q + K_{i+1}^{\mathrm{T}}RK_{i+1}]x_t.\end{aligned} \tag{15}$$

Thus, under the initial assumptions from the problem setup $Q \geq 0$, $R > 0$, $V_i(x_t)$ is a Lyapunov function proving that the updated control policy $u = -K_{i+1}x$, with $K_{i+1}$ given by Eq. (10), is stabilizing. □

**Remark 2.** Based on Lemma 2 one can conclude that if the initial control policy given by $K_1$ is stabilizing, then all policies obtained using the iteration (9)–(10) will be stabilizing policies.

Let $Ric(P_i)$ be the matrix valued function defined as

$$Ric(P_i) = A^{\mathrm{T}}P_i + P_iA + Q - P_iBR^{-1}B^{\mathrm{T}}P_i \tag{16}$$

and let $Ric'_{P_i}$ denote the Fréchet derivative of $Ric(P_i)$ taken with respect to $P_i$. The matrix function $Ric'_{P_i}$ evaluated at a given matrix $M$ will thus be $Ric'_{P_i}(M) = (A - BR^{-1}B^{\mathrm{T}}P_i)^{\mathrm{T}}M + M(A - BR^{-1}B^{\mathrm{T}}P_i)$.

**Lemma 3.** *The iteration between (9) and (10) is equivalent to Newton's method*

$$P_i = P_{i-1} - (Ric'_{P_{i-1}})^{-1}Ric(P_{i-1}). \tag{17}$$

**Proof.** Eqs. (11) and (10) can be compactly written as

$$A_i^{\mathrm{T}}P_i + P_iA_i = -(P_{i-1}BR^{-1}B^{\mathrm{T}}P_{i-1} + Q). \tag{18}$$

Subtracting $A_i^{\mathrm{T}}P_{i-1} + P_{i-1}A_i$ on both sides gives

$$\begin{aligned}&A_i^{\mathrm{T}}(P_i - P_{i-1}) + (P_i - P_{i-1})A_i \\ &\quad = -(P_{i-1}A + A^{\mathrm{T}}P_{i-1} - P_{i-1}BR^{-1}B^{\mathrm{T}}P_{i-1} + Q)\end{aligned} \tag{19}$$

which, making use of the introduced notations $Ric(P_i)$ and $Ric'_{P_i}$, is the Newton method formulation (17). □

**Theorem 4** (*Convergence*). *Under the assumptions of stabilizability of $(A, B)$ and detectability of $(Q^{1/2}, A)$, with $Q \geq 0$, $R > 0$ in the cost index (2), the policy iteration (9) and (10), conditioned by an initial stabilizing controller, converges to the optimal control solution given by (3) where the matrix $P$ satisfies the ARE (4).*

**Proof.** In Kleinman (1968) it has been shown that Newton's method, *i.e.* the iteration (11) and (10), conditioned by an initial stabilizing policy will converge to the solution of the ARE. Also, if the initial policy is stabilizing, all the subsequent control policies will be stabilizing (as by Lemma 2). The proven equivalence between (11) and (10), and (9) and (10), shows that the proposed new online policy iteration algorithm will converge to the solution of the optimal control problem (2) with the infinite horizon quadratic cost (3) — without using knowledge of the internal dynamics of the controlled system (1). □

Note that the only requirement for convergence to the optimal controller consists in an initial stabilizing policy that will guarantee a finite value for the cost $V_1(x_t) = x_t^{\mathrm{T}} P_1 x_t$. Under the assumption that the system to be controlled is stabilizable and implementation of an optimal state feedback controller is possible and desired, it is reasonable to assume that a stabilizing (though not optimal) state feedback controller is available to begin the iteration (Kleinman, 1968; Moris & Navasca, 2006). In fact in many cases the system to be controlled is itself stable such that the initial controller can be chosen as zero.

## 3. Online implementation of the adaptive optimal control algorithm without using knowledge of the system internal dynamics

For the implementation of the iteration scheme given by (9) and (10) one only needs to have knowledge of the $B$ matrix which explicitly appears in the policy update. The information regarding the system $A$ matrix is embedded in the states $x(t)$ and $x(t+T)$ which are observed online, and thus the system matrix is never required for the computation of either of the two steps of the policy iteration scheme. The details regarding the online implementation of the algorithm are discussed next. Simulation results obtained while finding the optimal controller for a power system are then presented.

### 3.1. Online implementation of the adaptive algorithm based on policy iteration

To find the parameters (matrix $P_i$) of the cost function associated with the policy $K_i$ in (9), the term $x^{\mathrm{T}}(t)P_i x(t)$ is written as

$$x^{\mathrm{T}}(t)P_i x(t) = \bar{p}_i^{\mathrm{T}} \bar{x}(t) \tag{20}$$

where $\bar{x}(t)$ denotes the Kronecker product quadratic polynomial basis vector with the elements $\{x_i(t)x_j(t)\}_{i=1,n;j=i,n}$ and $\bar{p} = v(P)$ with $v(.)$ a vector valued matrix function that acts on symmetric matrices and returns a column vector by stacking the elements of the diagonal and upper triangular part of the symmetric matrix into a vector where the off-diagonal elements are taken as $2P_{ij}$ (Brewer, 1978). Using (20), Eq. (9) is rewritten as

$$\bar{p}_i^{\mathrm{T}}(\bar{x}(t) - \bar{x}(t+T)) = \int_t^{t+T} x^{\mathrm{T}}(\tau)(Q + K_i^{\mathrm{T}} R K_i)x(\tau)\mathrm{d}\tau. \tag{21}$$

In this equation $\bar{p}_i$ is the vector of unknown parameters and $\bar{x}(t) - \bar{x}(t+T)$ acts as a regression vector. The right hand side target function, denoted $d(\bar{x}(t), K_i)$ (also known as the reinforcement on the time interval $[t, t+T]$),

$$d(\bar{x}(t), K_i) \equiv \int_t^{t+T} x^{\mathrm{T}}(\tau)(Q + K_i^{\mathrm{T}} R K_i)x(\tau)\mathrm{d}\tau$$

is measured based on the system states over the time interval $[t, t+T]$. Considering $\dot{V}(t) = x^{\mathrm{T}}(t)Qx(t) + u^{\mathrm{T}}(t)Ru(t)$ as a definition for a new state $V(t)$, augmenting the system (1), the value of $d(\bar{x}(t), K_i)$ can be measured by taking two measurements of this newly introduced system state since $d(\bar{x}(t), K_i) = V(t+T) - V(t)$. This new state signal is the output of an analog integration block having as inputs the quadratic terms $x^{\mathrm{T}}(t)Qx(t)$ and $u^{\mathrm{T}}(t)Ru(t)$ which can also be obtained using an analog processing unit.

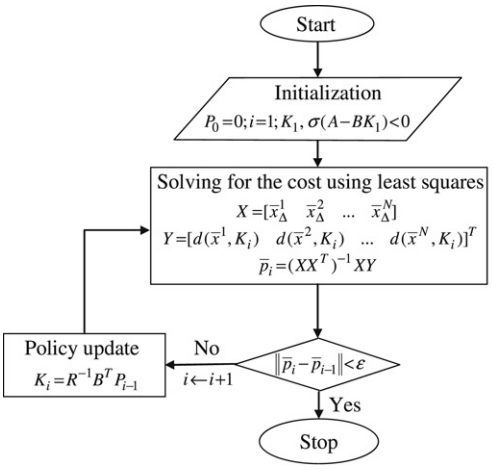


**Fig. 1.** Continuous-time linear policy iteration algorithm.

The parameter vector $\bar{p}_i$ of the function $V_i(x_t)$ (*i.e.* the Critic), which will then yield the matrix $P_i$, is found by minimizing, in the least-squares sense, the error between the target function, $d(\bar{x}(t), K_i)$, and the parameterized left hand side of (21). Evaluating the right hand side of (21) at $N \geq n(n+1)/2$ (the number of independent elements in the matrix $P_i$) points $\bar{x}^i$ in the state space, over the same time interval $T$, the least-squares solution is obtained as

$$\bar{p}_i = (XX^{\mathrm{T}})^{-1}XY \tag{22}$$

where

$$X = [\bar{x}_\Delta^1 \quad \bar{x}_\Delta^2 \quad \dots \quad \bar{x}_\Delta^N]$$
$$\bar{x}_\Delta^i = \bar{x}^i(t) - \bar{x}^i(t+T)$$
$$Y = [d(\bar{x}^1, K_i) \quad d(\bar{x}^2, K_i) \quad \dots \quad d(\bar{x}^N, K_i)]^{\mathrm{T}}.$$

The least-squares problem can be solved in real-time after a sufficient number of data points are collected along a single state trajectory, under the regular presence of an excitation requirement. A flow chart of the algorithm is presented in Fig. 1.

Alternatively, the solution given by (22) can be obtained also using recursive estimation algorithms (*e.g.* gradient descent algorithms or the Recursive Least Squares algorithm) in which case a persistence of excitation condition is required. For this reason there are no real issues related to the algorithm becoming computationally expensive with the increase of the state space dimension.

Relative to the convergence speed of the algorithm, it has been proven in Kleinman (1968) that Newton's method has quadratic convergence; by the proven equivalence (Theorem 4) the online algorithm proposed in this paper has the same property in the case in which the cost function associated with a given control policy (*i.e.* Eq. (9)) is solved for in a single step (*e.g.* using a method such as using the exact least-squares described by Eq. (22)). For the case in which the solution of the Eq. (9) is obtained iteratively, the convergence speed of the online algorithm proposed in this paper will decrease. In this case at each step in the policy iteration algorithm (which involves solving Eqs. (9) and (10)) a recursive gradient descent algorithm, which most often has exponential convergence, will be used

for solving Eq. (10). From this perspective one can resolve that the convergence speed of the online algorithm will depend on the chosen technique for solving Eq. (9); analyses along these lines are presented in details in adaptive control literature (*e.g.* see Ioannou & Fidan, 2006).

Although the value of the sample time $T$ does not affect in any way the convergence property of the online algorithm, it is related to the excitation condition necessary in the setup of a numerically well posed least squares problem and obtaining the least squares solution (22). More precisely, assuming without loss of generality that the matrix $X$ in (22) is square, and letting $\varepsilon > 0$ be a desired lower bound on the determinant of $X$, then the chosen sampling time $T$ must satisfy

$$T > \frac{a\,\varepsilon}{\prod_{l=1}^{n} |\lambda_l(A_c)|},$$

where $\lambda_l$ denotes the eigenvalues of the closed loop system and $a > 0$ is a scaling factor. From this point of view a minimal insight relative to the dynamics of the system would be required for choosing the sampling time $T$.

The proposed online policy iteration procedure requires only measurements of the states at discrete moments in time, $t$ and $t + T$, as well as knowledge of the observed cost over the time interval $[t, t + T]$, which is $d(\bar{x}(t), K_i)$. Therefore there is no required knowledge about the system $A$ matrix for the evaluation of the cost or the update of the control policy. The $B$ matrix is required for the update of the control policy, using (10), and this makes the tuning algorithm only partially model-free.

Compared with the algorithms presented in Murray et al. (2002), the policy iteration algorithm proposed in this paper also avoids the use of $A$ matrix knowledge and at the same time does not require measuring the state derivatives. Moreover, since the control policy evaluation requires measurements of the cost function over finite time intervals, the algorithm can converge (*i.e.* optimal control is obtained) while performing measurements along a single state trajectory, provided that there is enough initial excitation in the system. In this case, the control policy is updated at time $t + T$, after observing the state $x(t + T)$ and it is 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.

The structure of the system with the adaptive controller is presented in Fig. 2. Most important is that the system was augmented with an extra state $V(t)$, defined as $\dot{V} = x^{\mathrm{T}}Qx + u^{\mathrm{T}}Ru$, in order to extract the information regarding the cost associated with the given policy. This newly introduced system dynamics is part of the adaptive critic based controller thus the control scheme is actually a dynamic controller with the state given by the cost function $V$. One can observe that the adaptive optimal controller has a hybrid structure with a continuous-time internal state followed by a sampler and discrete time update rule.

It is shown that having little information about the system states, $x$, and the augmented system state, $V$ (controller dynamics), extracted from the system only at specific time values (*i.e.* the algorithm uses only the data samples $x(t)$, $x(t + T)$ and $V(t + T) - V(t)$ over several time samples), the critic is able to evaluate the performance of the system associated with a given control policy. The control policy is improved after the solution given by (22) is obtained. In this way, over a single state trajectory in which several policy evaluations and updates have taken place, the algorithm can converge to the optimal control policy. Sufficient excitation in the initial state of the system is nonetheless necessary, as the algorithm iterates only on stabilizing policies which will make the states go to zero. In the case that excitation is lost prior to obtaining the convergence (the system reaches the equilibrium point) a new experiment needs to be conducted having as a starting point the last policy from the previous experiment.

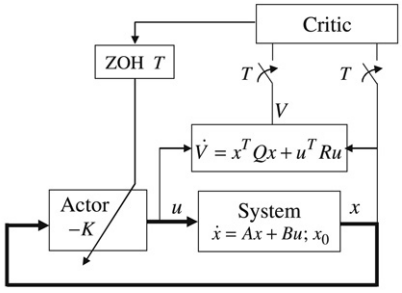


**Fig. 2.** Structure of the system with optimal adaptive controller.

The critic will stop updating the control policy when the difference between the performance of the system evaluated at two consecutive steps will cross below a designer specified threshold, *i.e.* the algorithm has converged to the optimal controller. Also in the case that this error is bigger than the above mentioned threshold the critic will again start tuning the actor parameters to obtain an optimal control policy. In fact, if the dynamics described by the $A$ matrix change suddenly, as long as the current controller is stabilizing for the new $A$ matrix, the algorithm will converge to the solution to the corresponding new ARE. Simulations showing successful performance were presented in Vrabie, Pastravanu, and Lewis (2007), which reported preliminary results to the ones presented here.

It is observed that the update of both the actor and the critic is performed at discrete moments in time. Nevertheless, the control action is a full-fledged continuous-time control, only that its constant gain is updated only at certain points in time. Moreover, 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, as was proven in Section 2.

### 3.2. Online-load-frequency controller design for a power system

In this section we present the results that are obtained in simulation while finding the optimal controller for a power system. Even though power systems are characterized by nonlinearities, linear state feedback control is regularly employed for load-frequency control at a certain nominal operating point. This simplifies the design problem, but a new issue appears as only the range of the plant parameters can be determined. Thus it is particularly advantageous to apply model-free methods to obtain the optimal LQR controller for a given operating point of the power system.

The plant that we consider is the linearized model of the power system presented in Wang et al. (1993).

The matrices of the plant are

$$A_{nom} = \begin{bmatrix} -0.0665 & 8 & 0 & 0 \\ 0 & -3.663 & 3.663 & 0 \\ -6.86 & 0 & -13.736 & -13.736 \\ 0.6 & 0 & 0 & 0 \end{bmatrix} \tag{23}$$

$$B = \begin{bmatrix} 0 & 0 & 13.736 & 0 \end{bmatrix}^{\mathrm{T}}.$$

For this simulation it was considered that the linear model of the real plant internal dynamics is given by

$$A = \begin{bmatrix} -0.0665 & 11.5 & 0 & 0 \\ 0 & -2.5 & 2.5 & 0 \\ -9.5 & 0 & -13.736 & -13.736 \\ 0.6 & 0 & 0 & 0 \end{bmatrix}. \quad (24)$$

The simulation was conducted using data obtained from the system at every 0.05 s. For the purpose of demonstrating the algorithm, the initial state of the system was $x_0 = [\ 0 \quad 0.1 \quad 0 \quad 0\ ]$. The cost function parameters, namely the $Q$ and $R$ matrices, were chosen to be identity matrices of appropriate dimensions. We start the iterative algorithm while using the controller calculated for the nominal model of the plant (23), and the controller parameters will be adapted online to converge to the optimal controller for the real plant.

In order to solve online for the values of the $P$ matrix which parameterizes the cost function, before each iteration step a least-squares problem of the sort described in Section 2.1, with the solution given by (22), was setup. Since there are 10 independent elements in the symmetric matrix $P$, at least 10 measurements of the cost function associated with the given control policy and values of the systems states at the beginning and the end of each time interval are required, provided that there is enough excitation in the system. When the system states are not continuously excited and because resetting the state at each step is not an acceptable solution for online implementation, in order to have consistent data necessary to obtain the solution given by (22), one has to continue reading information from the system until the solution of the least-squares problem is feasible. A least squares problem was solved after 20 sample data were acquired and thus the controller was updated every 1 s.

The result of applying the algorithm for the power system is presented in Fig. 3. It is clear that the cost function (*i.e.* Critic) parameters converged to the optimal ones – indicated on the figure with star shaped points – which were placed for comparison ease at $t = 5$ s. The values of the $P$ matrix parameters at $t = 0$ s correspond to the solution of the Riccati equation that was solved considering the approximate model of the system (23). The values of the cost function parameters, associated with the initial controller, are indicated by the points placed at $t = 1$ s. The optimal controller, close in the range of $10^{-4}$ to the solution of the Riccati equation, was obtained at time $t = 4$ s after four updates of the controller parameters. The $P$ matrix obtained online using the adaptive critic algorithm – without knowing the plant internal dynamics – is

$$P = \begin{bmatrix} 0.4599 & 0.6910 & 0.0518 & 0.4641 \\ 0.6910 & 1.8665 & 0.2000 & 0.5798 \\ 0.0518 & 0.2000 & 0.0532 & 0.0300 \\ 0.4641 & 0.5798 & 0.0300 & 2.2105 \end{bmatrix}. \quad (25)$$

The solution that was obtained by directly solving the algebraic Riccati equation considering the real plant internal dynamics (24) is

$$P = \begin{bmatrix} 0.4600 & 0.6911 & 0.0519 & 0.4642 \\ 0.6911 & 1.8668 & 0.2002 & 0.5800 \\ 0.0519 & 0.2002 & 0.0533 & 0.0302 \\ 0.4642 & 0.5800 & 0.0302 & 2.2106 \end{bmatrix}. \quad (26)$$

In practice, the convergence of the algorithm is considered to be achieved when the difference between the measured cost and the expected cost crosses below a designer specified threshold value. Note that after the convergence to the optimal controller was attained, the algorithm need not continue to be run and subsequent updates of the controller need not be performed.

In Fig. 4 is presented a detail of the system state trajectories for the first 2 seconds of the simulation. The state values that were actually measured and subsequently used for the Critic update computation are represented by the points on the state trajectories. Note that the control policy was updated at time $t = 1$ s.

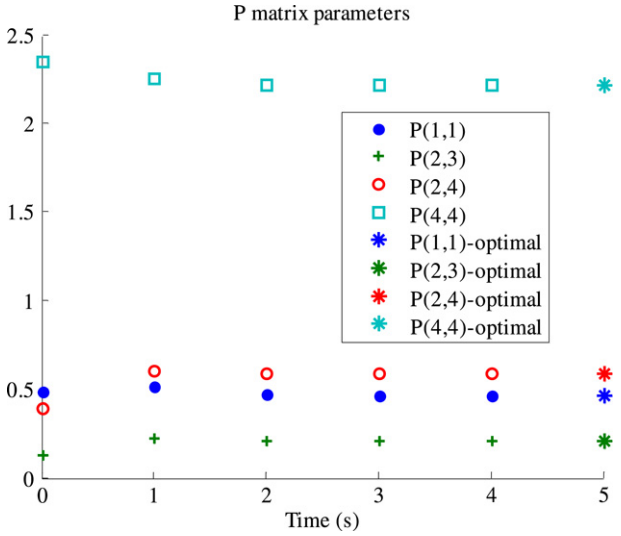


**Fig. 3.** Evolution of the parameters of the $P$ matrix for the duration of the experiment.

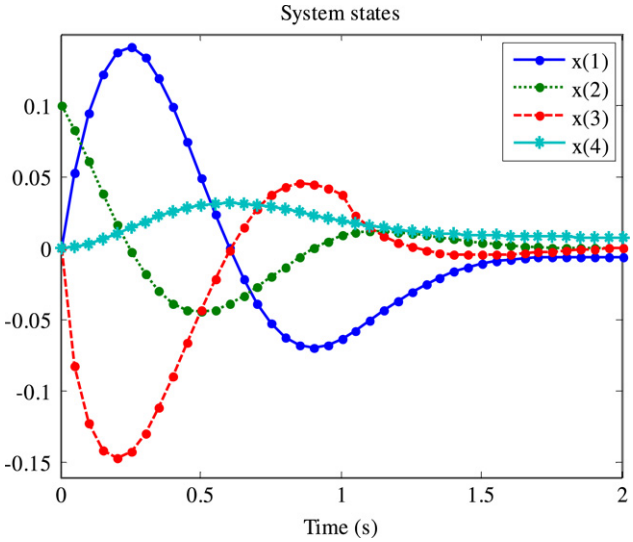


**Fig. 4.** System state trajectories (lines) and state information that was actually used for the Critic update (dots on the state trajectories).

It is important to point out that in the case when the system to be controlled is itself stable this allows starting the iteration while using no controller (*i.e.* the initial controller is zero and no identification procedure needs to be performed). Fig. 5 presents the convergence result for the case the adaptive optimal control algorithm was initialized with no controller.

The Critic parameters converged to the optimal ones at time $t = 7$ s after seven updates of the controller parameters. The $P$ matrix calculated with the adaptive algorithm is

$$P = \begin{bmatrix} 0.4601 & 0.6912 & 0.0519 & 0.4643 \\ 0.6912 & 1.8672 & 0.2003 & 0.5800 \\ 0.0519 & 0.2003 & 0.0533 & 0.0302 \\ 0.4643 & 0.5800 & 0.0302 & 2.2107 \end{bmatrix}, \quad (27)$$

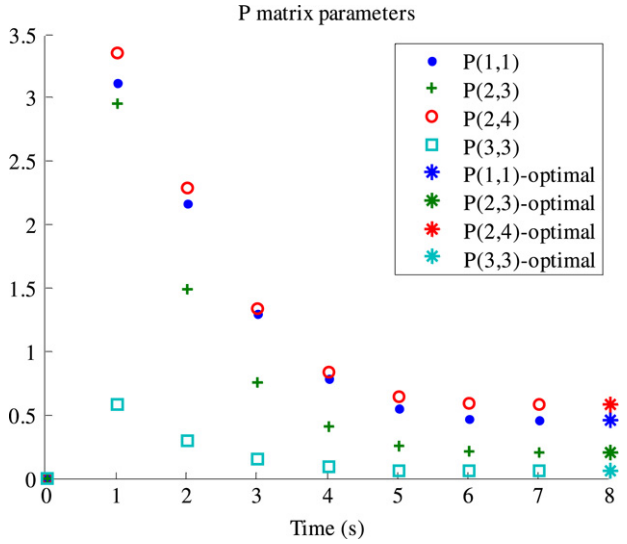


**Fig. 5.** Evolution of the parameters of the $P$ matrix for the duration of the experiment when the adaptive algorithm was started without controller for the power system.

the error difference between the parameters of the solution (27) obtained iteratively and the optimal solution (26) is in the range of $10^{-4}$.

## 4. Conclusions

In this paper we proposed a new policy iteration technique to solve online the continuous time LQR problem without using knowledge about the system's internal dynamics (system matrix $A$). The algorithm is an online adaptive optimal controller based on an adaptive critic scheme in which the actor performs continuous time control while the critic incrementally corrects the actor's behavior at discrete moments in time until best performance is obtained. The critic evaluates the actor performance over a period of time and formulates it in a parameterized form. Based on the critic's evaluation the actor behavior policy is updated for improved control performance.

The result can be summarized as an algorithm which effectively provides solution to the algebraic Riccati equation associated with the optimal control problem without using knowledge of the system matrix $A$. Convergence to the solution of the optimal control problem, under the condition of initial stabilizing controller, has been established by proving equivalence with the algorithm presented in Kleinman (1968). The convergence results obtained in simulation for load-frequency optimal control of a power system generator have also been provided.

## Acknowledgements

This research was supported by the National Science Foundation, ECS-0501451 and the Army Research Office, W91NF-05-1-0314.

## References

Abu-Khalaf, M., Lewis, F. L., & Huang, J. (2006). Policy iterations and the Hamilton–Jacobi–Isaacs equation for H-infinity state feedback control with input saturation. *IEEE Transactions on Automatic Control*, *51*(12), 1989–1995.

Abu-Khalaf, M., & Lewis, F. L. (2005). Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. *Automatica*, *41*(5), 779–791.

Al-Tamimi, A., Abu-Khalaf, M., & Lewis, F. L. (2007). Model-free Q-learning designs for discrete-time zero-sum games with application to H-Infinity control. *Automatica*, *43*(3), 473–482.

Baird, L. C. III (1994). Reinforcement learning in continuous time: Advantage updating. In *Proc. of ICNN*.

Balzer, L. A. (1980). Accelerated convergence of the matrix sign function method of solving Lyapunov, Riccati and other equations. *International Journal of Control*, *32*(6), 1076–1078.

Banks, H. T., & Ito, K. (1991). A numerical algorithm for optimal feedback gains in high dimensional linear quadratic regulator problems. *SIAM Journal on Control and Optimization*, *29*(3), 499–515.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*. MA: Athena Scientific.

Byers, R. (1987). Solving the algebraic Riccati equation with the matrix sign. *Linear Algebra and its Applications*, *85*, 267–279.

Bradtke, S.J., Ydestie, B.E., & Barto, A.G. (1994). Adaptive linear quadratic control using policy iteration. In: *Proc. of ACC* (pp. 3475–3476).

Brewer, J. W. (1978). Kronecker products and matrix calculus in system theory. *IEEE Transactions on Circuits and Systems*, *25*(9), 772–781.

Freeman, R. A., & Kokotovic, P. (1996). *Robust nonlinear control design: State-space and Lyapunov techniques*. Boston, MA: Birkhauser.

Guo, C. H., & Lancaster, P. (1998). Analysis and modification of Newton's method for algebraic Riccati equations. *Mathematics of Computation*, *67*(223), 1089–1105.

Hasan, M.A., Yang, J.S., & Hasan, A.A. (1999). A method for solving the algebraic Riccati and Lyapunov equations using higher order matrix sign function algorithms. In: *Proc. of ACC* (pp. 2345–2349).

Howard, R. A. (1960). *Dynamic programming and Markov processes*. Cambridge, MA: MIT Press.

Ioannou, P., & Fidan, B. (2006). Adaptive control tutorial. In *Advances in design and control*. PA: SIAM.

Kailath, T. (1973). Some new algorithms for recursive estimation in constant linear systems. *IEEE Transactions on Information Theory*, *19*(6), 750–760.

Kleinman, D. (1968). On an iterative technique for Riccati equation computations. *IEEE Transactions on Automatic Control*, *13*(1), 114–115.

Krstic, M., & Deng, H. (1998). *Stabilization of nonlinear uncertain systems*. Springer.

Landelius, T. (1997). *Reinforcement learning and distributed local model synthesis. Ph.D. dissertation*. Sweden: Linkoping University.

Laub, A. J. (1979). A Schur method for solving algebraic Riccati equations. *IEEE Transactions on Automatic Control*, *24*(6), 913–921.

Lewis, F. L., & Syrmos, V. L. (1995). *Optimal control*. John Wiley.

Li, Z.H., & Krstic, M. (1997). Optimal design of adaptive tracking controllers for nonlinear systems. In: *Proc. of ACC* (pp. 1191–1197).

MacFarlane, A. G. J. (1963). An eigenvector solution of the optimal linear regulator problem. *Journal of Electronics and Control*, *14*, 643–654.

Moris, K., & Navasca, C. (2006). Iterative solution of algebraic Riccati equations for damped systems. In: *Proc. of CDC* (pp. 2436–2440).

Murray, J. J., Cox, C. J., Lendaris, G. G., & Saeks, R. (2002). Adaptive dynamic programming. *IEEE Transactions on Systems, Man and Cybernetics*, *32*(2), 140–153.

Potter, J. E. (1966). Matrix quadratic solutions. *SIAM Journal on Applied Mathematics*, *14*, 496–501.

Sutton, R.S., Barto, A.G., & Williams, R.J. (1991). Reinforcement learning is direct adaptive optimal control. In: *Proc. of ACC* (pp. 2143–2146).

Vrabie, D., Pastravanu, O., & Lewis, F. L. (2007). Policy iteration for continuous-time systems with unknown internal dynamics. In *Proceedings of MED*.

Wang, Y., Zhou, R., & Wen, C. (1993). Robust load-frequency controller design for power systems. *IEE Proceedings C*, *140*(1), 11–16.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards. Ph.D. thesis*. England: University of Cambridge.

Werbos, P. (1989). Neural networks for control and system identification. In: *Proc. of CDC* (pp. 260–265).

White, D. A., & Sofge, D. A. (Eds.) (1992). *Handbook of intelligent control*. New York: Van Nostrand Reinhold.

**D. Vrabie** received her B.Sc. in Automatic Control and Industrial Informatics from the Automatic Control and Computer Engineering Dept., "Gh. Asachi" Technical University of Iasi in 2003. She received her M.Sc. degree in Control Engineering from the above mentioned faculty for the work "Neuro-predictive Method for On-line Controller Tuning" in 2004. Since May 2005, she has been pursuing her Ph.D. degree and working as a research assistant at the Automation and Robotics Research Institute, University of Texas at Arlington. Her research interests include Approximate Dynamic Programming for continuous state and action spaces, optimal control, adaptive control, Model Predictive Control, and general theory of nonlinear systems.