*J Control Theory Appl* 2011 9 (3) 353–360
DOI 10.1007/s11768-011-0166-4

# Adaptive dynamic programming for online solution of a zero-sum differential game

Draguna VRABIE [1], Frank LEWIS [2]

1.United Technologies Research Center, East Hartford, CT 06108, U.S.A.;

2.Automation and Robotics Research Institute, University of Texas at Arlington, Fort Worth, TX 76118, U.S.A.

**Abstract:** This paper will present an approximate/adaptive dynamic programming (ADP) algorithm, that uses the idea of integral reinforcement learning (IRL), to determine online the Nash equilibrium solution for the two-player zero-sum differential game with linear dynamics and infinite horizon quadratic cost. The algorithm is built around an iterative method that has been developed in the control engineering community for solving the continuous-time game algebraic Riccati equation (CT-GARE), which underlies the game problem. We here show how the ADP techniques will enhance the capabilities of the offline method allowing an online solution without the requirement of complete knowledge of the system dynamics. The feasibility of the ADP scheme is demonstrated in simulation for a power system control application. The adaptation goal is the best control policy that will face in an optimal manner the highest load disturbance.

**Keywords:** Approximate/Adaptive dynamic programming; Game algebraic Riccati equation; Zero-sum differential game; Nash equilibrium

## 1 Introduction

In this paper, we present a continuous-time adaptive dynamic programming (ADP) procedure that uses the idea of integral reinforcement learning (IRL) to find online the Nash equilibrium solution for the two-player zero-sum differential game with linear dynamics and infinite horizon quadratic cost. In the introduction section we first motivate the interest in zero-sum infinite horizon differential games from the perspective of control systems theory. We then provide a brief overview on the methods that are used to find the Nash equilibrium of the games and outline the characteristics of the new ADP method to be proposed.

There are numerous control applications in which the presence of disturbance signals is certain and will have a negative effect on the performance of the control system. In these cases the optimal control problem is formulated with the purpose of finding all admissible controllers that minimize the $\mathrm{H}_\infty$ norm. These methods are known as $\mathrm{H}_\infty$ controllers. Such control policies counteract, in an optimal sense, the effects of the worst disturbance that might affect the system.

The solution to the $\mathrm{H}_\infty$ problem is the saddle point solution to a two-player zero-sum differential game (see, e.g., [1]). For the case when the system has linear dynamics and the cost index is quadratic and has infinite horizon, finding the Nash equilibrium solution to the game problem is equivalent with solving a Riccati equation with sign indefinite quadratic term (see, e.g., [1–5]), which is the game algebraic Riccati equation (GARE)

$$0=A^{\mathrm{T}}P+PA+C^{\mathrm{T}}C-P(B_2B_2^{\mathrm{T}}-\frac{1}{\gamma^{*2}}DD^{\mathrm{T}})P. \quad (1)$$

Suboptimal $\mathrm{H}_\infty$ controllers can be determined such that the $\mathrm{H}_\infty$ norm is less than a given prescribed bound that is larger than the minimum $\mathrm{H}_\infty$ norm [2]. This means that for any $\gamma > \gamma^*$, one can find a suboptimal $\mathrm{H}_\infty$ state-feedback controller, which admits a performance level of at least $\gamma$, by solving equation (1) where $\gamma^*$ has been replaced with $\gamma$. In the following, to simplify the mathematical notation, for a given $\gamma > \gamma^*$, we will denote $B_1 = \gamma^{-1}D$. Thus, the GARE of our concern will be written as

$$0 = A^{\mathrm{T}}P + PA + C^{\mathrm{T}}C - P(B_2B_2^{\mathrm{T}} - B_1B_1^{\mathrm{T}})P. \quad (2)$$

The solution to the GARE has been approached in [6–8] while for its nonlinear generalization, that is, the Hamilton Jacobi Isaacs (HJI) equation, iterative solutions have been presented in [9–11]. In all cases the solution is determined in an iterative manner by means of a Newton-type of algorithm. These algorithms construct sequences of cost functions that are monotonically convergent to the solution to interest. In all cases exact knowledge of the system dynamics is required and the solution is obtained by means of offline computation.

ADP methods provide an online solution to optimal control problems while making use of measured information from the system and using computation in a forward in time fashion, as opposed to the backward in time procedure that characterizes the classical Dynamic Programming approach. These methods were initially developed for systems with finite state and action spaces and are based on Sutton's temporal difference learning [12], Werbos' heuristic dynamic programming (HDP) [13], and Watkins's $Q$-learning [14].

To our knowledge, there exists a single ADP procedure that provides a solution for the HJI equation [15]. The algorithm presented in [15] involves calculation of two se-

Received 14 July 2010; revised 23 May 2011.
This work was supported by the National Science Foundation (No.ECCS-0801330), and the Army Research Office (No.W91NF-05-1-0314).

quences of cost functions, the upper and lower performance indices, which will converge to the saddle point solution to the game. The adaptive critic structure that is required for learning the saddle point solution is comprised of four action networks and two critic networks. The requirement of full knowledge on the system dynamics is still present.

The result presented in this paper is a reinforcement learning approach to the saddle point solution for a two player zero-sum differential game. Here we use the idea of IRL, introduced in [16], which allows calculation of the value function associated with the pair of behavior policies of the two players. By virtue of the online ADP method exact knowledge of part of the system dynamics is not required. Specifically, the $A$ matrix that is part of the system dynamics, and appears explicitly in (2), need not be known.

The objective of this paper is to present an online algorithm that makes use of ADP techniques to provide a solution to the two-player differential zero-sum game. The main traits of this new online procedure are as follows:

- It is sustained by a mathematical algorithm [8] that has been developed in the control engineering community to solve the underlying equation of the game problem. This supporting algorithm makes use of offline procedures and requires full knowledge of the system dynamics to determine the Nash equilibrium of the game.

- It involves the use of ADP techniques, namely the IRL technique [17], developed in the computational intelligence community. Such ADP techniques will enhance the capabilities of the supporting algorithm transforming it into an online data-based procedure that does not require full knowledge of the system dynamics.

The novel online procedure that will be described here is built on the mathematical result given in [8]. This algorithm involves solving a sequence of Riccati equations, in order to build a monotonically increasing sequence of matrices that converges to the equilibrium solution to the game. Every matrix in this sequence is determined by solving a Riccati equation with sign definite quadratic term of the sort associated with the optimal control problem [18]. In this paper, we use ADP techniques and the idea of IRL for finding the solution to these optimal control problems in an online fashion and using reduced information on the system dynamics.

While working in the framework of control applications we will be referring to the two players as ‘controller’ and ‘disturbance’, in contrast to the regular nomenclature used in game theory – ‘pursuer’ and ‘evader’. We are assuming that both players have perfect instantaneous state information. A player’s policy is a function that allows computation of the player’s output signal based on the state information. Both players are competing and learning in real-time, and the two policies that characterize the Nash-equilibrium solution to the game will be determined based on online data measured from the system.

In the ADP approach described in this paper only one of the two players is actively learning and improving its policy. The algorithm is built on the interplay between a learning phase, performed by the controller that is learning in order to optimize its behavior, and a policy update step, performed by the disturbance that is gradually increasing its detrimental effect. The controller is learning online in order to maximize its performance, while the policy of the disturbance remains constant. The disturbance player will update its action policy only after the controller has learned its optimal behavior in front of the present policy of his opponent. The update of the disturbance policy uses the information on the policy of his opponent and gives way for further improvement for the controller policy. For learning the control policy in this paper we will use the online continuous-time HDP procedure developed in [16].

We begin our investigation by providing the formulation of the two player zero-sum game problem and reviewing the iterative mathematical algorithm that provides a solution for its underlying CT-GARE. Next, we briefly describe the online HDP algorithm that is used to determine the solution to the single player optimal control problem. Section 3 will present and discuss the online algorithm that establishes the solution to the two-player zero-sum game. Here, we will introduce four propositions that provide intuition on the online learning procedure. The adaptive critic structure that arises will be given. Section 4 will provide and discuss the simulation results that were obtained for the control of a power system. The goal is to determine online the best control policy that will face in an optimal manner the highest load disturbance.

## 2 Preliminaries

### 2.1 Problem formulation

Consider the system described by the equation

$$\dot{x} = Ax + B_1 w + B_2 u, \tag{3}$$

where $x \in \mathbb{R}^n, u \in \mathbb{R}^m, w \in \mathbb{R}^q$.

The controller player, which computes the signal $u$, desires to minimize the quadratic performance index

$$V(x_0, u, w) = \int_0^\infty (x^\mathrm{T} C^\mathrm{T} C x + u^\mathrm{T} u - w^\mathrm{T} w)\mathrm{d}t, \tag{4}$$

while the disturbance player, which computes the signal $w$, desires to maximize it. The goal is to determine the saddle point stabilizing equilibrium solution defined by the most effective control policy against the worst case disturbance policy. The motivation for the choice of the performance index (4) is tightly connected to the formulation of the disturbance attenuation problem. For a good intuitive presentation of the topic the reader is referred to [19].

The following assumptions, related to the solvability of the two-player game, are made:

- all unobservable modes of $(C, A)$ are strictly stable,
- $(A, B_2)$ is stabilizable, and
- there exists a unique positive definite stabilizing solution to the GARE (2).

Denoting with $\Pi$ the unique positive definite stabilizing solution to (2) the saddle point of the Nash game is

$$\begin{cases} u = -B_2^\mathrm{T} \Pi x, \\ w = B_1^\mathrm{T} \Pi x, \\ V(x_0, u, w) = x_0^\mathrm{T} \Pi x_0. \end{cases} \tag{5}$$

We shall use the notations $u = Kx$ and $w = Lx$ for the state feedback control and the disturbance policies, respectively. We say that $K$ is the gain of the control policy and $L$ is the gain of the disturbance policy. The meaning of the saddle point solution to the Nash differential game is that for any state feedback control policy $\tilde{u} = \tilde{K}x$ and any state-feedback disturbance policy $\tilde{w} = \tilde{L}x$, different than the ones in (5), the value of the game will satisfy

$$V(x_0, \tilde{u}, w) \geqslant V(x_0, u, w) \geqslant V(x_0, u, \tilde{w}). \tag{6}$$

### 2.2 Offline iterative algorithm that solves the GARE

Given real matrices $A, B_1, B_2, C$ with compatible dimensions, define the map

$$\begin{cases} F : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}, \\ F(P) = A^{\mathrm{T}}P + PA + C^{\mathrm{T}}C - P(B_2B_2^{\mathrm{T}} - B_1B_1^{\mathrm{T}})P. \end{cases} \tag{7}$$

The following iterative method for solving the GARE (2) has been introduced in [8]. We use the notation

$$A_{\mathrm{u}}^{i-1} = A + B_1B_1^{\mathrm{T}}P_{\mathrm{u}}^{i-1} - B_2B_2^{\mathrm{T}}P_{\mathrm{u}}^{i-1}.$$

**Algorithm 1** (Iterations on the control policy)

1) Start with

$$P_{\mathrm{u}}^0 = 0. \tag{8}$$

2) Solve, for the unique positive definite $Z_{\mathrm{u}}^i$,

$$0 = (A_{\mathrm{u}}^{i-1})^{\mathrm{T}}Z_{\mathrm{u}}^i + Z_{\mathrm{u}}^iA_{\mathrm{u}}^{i-1} - Z_{\mathrm{u}}^iB_2B_2^{\mathrm{T}}Z_{\mathrm{u}}^i + F(P_{\mathrm{u}}^{i-1}). \tag{9}$$

3) Update

$$P_{\mathrm{u}}^i = P_{\mathrm{u}}^{i-1} + Z_{\mathrm{u}}^i. \tag{10}$$

The subindex ‘u’ has been used in the algorithm to underline the fact that the controller player is the one that will be learning online to find a solution to the sequence of Riccati equations (9).

The convergence of the algorithm to the unique positive definite solution to the GARE (2) was given by the following result in [8].

**Theorem 1** [8] Given real matrices $A, B_1, B_2, C$ with compatible dimensions, such that all unobservable modes of $(C, A)$ are strictly stable and $(A, B_2)$ stabilizable, define the map $F$ as in (7). Suppose that there exists a stabilizing solution $\mathit{\Pi} > 0$ of (2).

Then,

I) there exist two square matrix series $P_{\mathrm{u}}^i \in \mathbb{R}^{n\times n}$ and $Z_{\mathrm{u}}^i \in \mathbb{R}^{n\times n}$for all $i \in \mathbb{N}$ satisfying Algorithm 1;

II) the elements of the two series, defined recursively, have the following properties:

a) $(A + B_1B_1^{\mathrm{T}}P_{\mathrm{u}}^i, B_2)$ is stabilizable for all $i \in \mathbb{N}$,

b) $Z_{\mathrm{u}}^i \geqslant 0, \forall i \in \mathbb{N}$,

c) $F(P_{\mathrm{u}}^{i+1}) = Z_{\mathrm{u}}^iB_1B_1^{\mathrm{T}}Z_{\mathrm{u}}^i, \forall i \in \mathbb{N}$,

d) $A + B_1B_1^{\mathrm{T}}P_{\mathrm{u}}^i - B_2B_2^{\mathrm{T}}P_{\mathrm{u}}^{i+1}$ is Hurwitz, $\forall i \in \mathbb{N}$,

e) $\mathit{\Pi} \geqslant P_{\mathrm{u}}^{i+1} \geqslant P_{\mathrm{u}}^i \geqslant 0, \forall i \in \mathbb{N}$;

III) let $P_{\mathrm{u}}^{\infty} = \lim\limits_{i\to\infty} P_{\mathrm{u}}^i \geqslant 0$, then $P_{\mathrm{u}}^{\infty} = \mathit{\Pi}$.

We now introduce two propositions that provide equivalent formulations for Algorithm 1. We are introducing them here in order to bring meaning to every step of the iterative algorithm. These propositions will provide a compact mathematical formulation for the iterative algorithm 1. Using these results we will then show, by means of Propositions 3 and 4, that the iterative algorithm that provides a solution for the two-player zero sum game involves solving a sequence of single-player games, namely solving a sequence of optimal control problems.

**Proposition 1** The iteration between (9) and (11) in Algorithm 1 can be written as

$$\begin{aligned} &P_{\mathrm{u}}^i(A + B_1B_1^{\mathrm{T}}P_{\mathrm{u}}^{i-1}) + (A + B_1B_1^{\mathrm{T}}P_{\mathrm{u}}^{i-1})^{\mathrm{T}}P_{\mathrm{u}}^i \\ &-P_{\mathrm{u}}^iB_2B_2^{\mathrm{T}}P_{\mathrm{u}}^i - P_{\mathrm{u}}^{i-1}B_1B_1^{\mathrm{T}}P_{\mathrm{u}}^{i-1} + C^{\mathrm{T}}C = 0. \end{aligned} \tag{11}$$

**Proof** The result is obtained by writing compactly the two equations and making use of the definition of the map $F$.

Explicitly, (11) becomes

$$\begin{aligned} 0 = {} & (A_{\mathrm{u}}^{i-1})^{\mathrm{T}}(P_{\mathrm{u}}^i - P_{\mathrm{u}}^{i-1}) + (P_{\mathrm{u}}^i - P_{\mathrm{u}}^{i-1})A_{\mathrm{u}}^{i-1} \\ & -(P_{\mathrm{u}}^i - P_{\mathrm{u}}^{i-1})B_2B_2^{\mathrm{T}}(P_{\mathrm{u}}^i - P_{\mathrm{u}}^{i-1}) + A^{\mathrm{T}}P_{\mathrm{u}}^{i-1} \\ & +P_{\mathrm{u}}^{i-1}A + C^{\mathrm{T}}C - P_{\mathrm{u}}^{i-1}(B_2B_2^{\mathrm{T}} - B_1B_1^{\mathrm{T}})P_{\mathrm{u}}^{i-1}. \end{aligned}$$

Unfolding the parentheses, using the notation

$$A_{\mathrm{u}}^{i-1} = A + B_1B_1^{\mathrm{T}}P_{\mathrm{u}}^{i-1} - B_2B_2^{\mathrm{T}}P_{\mathrm{u}}^{i-1},$$

and cancelling the equivalent terms we obtain

$$\begin{aligned} 0 = {} & P_{\mathrm{u}}^{i-1}B_1B_1^{\mathrm{T}}(P_{\mathrm{u}}^i - P_{\mathrm{u}}^{i-1}) - P_{\mathrm{u}}^{i-1}B_2B_2^{\mathrm{T}}(P_{\mathrm{u}}^i - P_{\mathrm{u}}^{i-1}) \\ & +(P_{\mathrm{u}}^i - P_{\mathrm{u}}^{i-1})B_1B_1^{\mathrm{T}}P_{\mathrm{u}}^{i-1} - (P_{\mathrm{u}}^i - P_{\mathrm{u}}^{i-1})B_2B_2^{\mathrm{T}}P_{\mathrm{u}}^{i-1} \\ & -P_{\mathrm{u}}^iB_2B_2^{\mathrm{T}}P_{\mathrm{u}}^i + P_{\mathrm{u}}^iB_2B_2^{\mathrm{T}}P_{\mathrm{u}}^{i-1} + P_{\mathrm{u}}^{i-1}B_2B_2^{\mathrm{T}}P_{\mathrm{u}}^i \\ & -P_{\mathrm{u}}^{i-1}B_2B_2^{\mathrm{T}}P_{\mathrm{u}}^{i-1} + A^{\mathrm{T}}P_{\mathrm{u}}^i + P_{\mathrm{u}}^iA + C^{\mathrm{T}}C \\ & -P_{\mathrm{u}}^{i-1}(B_2B_2^{\mathrm{T}} - B_1B_1^{\mathrm{T}})P_{\mathrm{u}}^{i-1}. \end{aligned}$$

After more cancelations and rearranging, we obtain

$$\begin{aligned} 0 = {} & P_{\mathrm{u}}^{i-1}B_1B_1^{\mathrm{T}}P_{\mathrm{u}}^i + P_{\mathrm{u}}^iB_1B_1^{\mathrm{T}}P_{\mathrm{u}}^{i-1} + A^{\mathrm{T}}P_{\mathrm{u}}^i + P_{\mathrm{u}}^iA \\ & -P_{\mathrm{u}}^iB_2B_2^{\mathrm{T}}P_{\mathrm{u}}^i - P_{\mathrm{u}}^{i-1}B_1B_1^{\mathrm{T}}P_{\mathrm{u}}^{i-1} + C^{\mathrm{T}}C. \end{aligned}$$

That is the result in (11).

**Proposition 2** The iteration between (9) and (10) in Algorithm 1 can be written as

$$\begin{aligned} 0 = {} & (P_{\mathrm{u}}^i - P_{\mathrm{u}}^{i-1})A_{\mathrm{u}}^{i-1} + (A_{\mathrm{u}}^{i-1})^{\mathrm{T}}(P_{\mathrm{u}}^i - P_{\mathrm{u}}^{i-1}) \\ & -(P_{\mathrm{u}}^i - P_{\mathrm{u}}^{i-1})B_2B_2^{\mathrm{T}}(P_{\mathrm{u}}^i - P_{\mathrm{u}}^{i-1}) \\ & +(P_{\mathrm{u}}^{i-1} - P_{\mathrm{u}}^{i-2})B_1B_1^{\mathrm{T}}(P_{\mathrm{u}}^{i-1} - P_{\mathrm{u}}^{i-2}). \end{aligned} \tag{12}$$

This results directly from (9), (10) and Theorem 1 II) c).

It is important to notice at this point that the result given in Proposition 2 includes three instances of the index of the sequence $\{P_{\mathrm{u}}^i\}_{i\geqslant 0}$, namely $P_{\mathrm{u}}^{i-2}, P_{\mathrm{u}}^{i-1}, P_{\mathrm{u}}^i$. For this reason it can only be used for calculating the values $\{P_{\mathrm{u}}^i\}_{i\geqslant 2}$ provided that the first two elements in the sequence are available.

The next two propositions that we introduce are formulating optimal control problems that are associated with the Riccati equations (11) and (12). This is important because they attach meaning to the recursive algorithm, and thus are enhancing both the reinforcement learning perspective and the game theoretical reasoning.

**Proposition 3** Solving the Riccati equation (11) is equivalent to finding the solution to the following optimal control problem:

‘For the system

$$\dot{x} = A + B_1w_{i-1} + B_2u_i,$$

let the state-feedback disturbance policy gain be $L_{\mathrm{u}}^{i-1} = B_1^{\mathrm{T}}P_{\mathrm{u}}^{i-1}$ such that $w_{i-1} = B_1^{\mathrm{T}}P_{\mathrm{u}}^{i-1}x$. Determine the state-feedback control policy $u_i$ that minimizes the infinite horizon quadratic cost index

$$\int_0^{\infty} [x^{\mathrm{T}}C^{\mathrm{T}}Cx - w_{i-1}^{\mathrm{T}}w_{i-1} + u_i^{\mathrm{T}}u_i]\mathrm{d}t$$

is minimized.’

Let

$$x_0^{\mathrm{T}} P_{\mathrm{u}}^i x_0 = \min_{u_i} \int_0^\infty [x^{\mathrm{T}}(C^{\mathrm{T}}C - P_{\mathrm{u}}^{i-1}B_1B_1^{\mathrm{T}}P_{\mathrm{u}}^{i-1})x + u_i^{\mathrm{T}}u_i]\mathrm{d}t,$$

then the optimal state-feedback control policy is described by $K_{\mathrm{u}}^i = -B_2^{\mathrm{T}}P_{\mathrm{u}}^i$, such that the optimal state-feedback control is $u_i = -B_2^{\mathrm{T}}P_{\mathrm{u}}^i x$.

**Proposition 4** Solving the Riccati equation (12) is equivalent to finding the solution to the following optimal control problem:

'For the system

$$\dot{x} = A + B_1 w_{i-1} + B_2(u_{i-1} + \hat{u}_i)$$

let the state-feedback disturbance policy be $w_{i-1} = B_1^{\mathrm{T}}P_{\mathrm{u}}^{i-1}x$ and the base state-feedback control policy be $u_{i-1} = -B_2^{\mathrm{T}}P_{\mathrm{u}}^{i-1}x$. Determine the correction for the state-feedback control policy, $\hat{u}_i$, which minimizes the infinite horizon quadratic cost index

$$\hat{J} = \int_0^\infty [\,x^{\mathrm{T}}(P_{\mathrm{u}}^{i-1} - P_{\mathrm{u}}^{i-2})B_1B_1^{\mathrm{T}}(P_{\mathrm{u}}^{i-1} - P_{\mathrm{u}}^{i-2})x + \hat{u}_i^{\mathrm{T}}\hat{u}_i]\mathrm{d}t.\text{'}$$

Let $x_0^{\mathrm{T}}Z_{\mathrm{u}}^i x_0 = \min\limits_{\hat{u}_i} \hat{J}$ then the optimal control policy $u_i = u_{i-1} + \hat{u}_i$ is $u_i = -B_2^{\mathrm{T}}(P_{\mathrm{u}}^{i-1} + Z_{\mathrm{u}}^i)x$.

The iterative algorithm discussed above can be used as the backbone for the online approach to the saddle point solution for the zero-sum differential game. In the next section we will describe the online algorithm.

Because the iterative algorithm reaches the solution to the zero-sum game by means of building a sequence of solutions for Riccati equations, in the next subsection we provide a brief description of the continuous-time heuristic dynamic programming (CT-HDP) method that finds, in an online fashion, the solution to a Riccati equation with a sign definite quadratic term.

### 2.3 CT-HDP method to find the solution to an ARE

The goal of this section is to briefly present the online algorithm that uses reinforcement learning ideas to solve the ARE with sign definite quadratic term

$$A^{\mathrm{T}}P + PA + Q - PBR^{-1}B^{\mathrm{T}}P = 0. \tag{13}$$

We note that the notation used in this subsection is general and not specifically related to the notations used in the previous subsections. In effect here we present an online procedure, introduced in [16], that provides the solution to a standard continuous-time state-feedback optimal problem with quadratic performance cost and that of its associated ARE (13).

The reinforcement learning-based value iteration algorithm that provides the unique positive definite solution to (13) is as follows:

Let $P_0 = 0$ and let $K_0$ be a state-feedback control policy (not necessarily stabilizing). Iterate between

$$x_t^{\mathrm{T}}P_{i+1}x_t = \int_t^{t+T_0} x_\tau^{\mathrm{T}}(Q + K_i^{\mathrm{T}}RK_i)x_\tau \mathrm{d}\tau + x_{t+T_0}^{\mathrm{T}}P_i x_{t+T_0}, \tag{14}$$

$$K_{i+1} = R^{-1}B^{\mathrm{T}}P_{i+1} \tag{15}$$

until convergence, where $x_\tau$ denotes the state of the system described by $\dot{x} = (A + BK_i)x$ with initial condition $x_t$. The value of the state at time $t + T_0$ is denoted with $x_{t+T_0}$. The online implementation of the algorithm is given next.

The solution to (14) consists of the value of the matrix $P_{i+1}$ that is parameterizing the cost function. The two quadratic cost functions will be written as

$$x_t^{\mathrm{T}}P_i x_t = \bar{p}_i^{\mathrm{T}}\bar{x}_t, \tag{16}$$

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 $\bar{p} = \nu(P)$ with $\nu(\,\cdot\,)$ 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 $2P_{ij}$ [20]. Denote the reinforcement over the time interval $[t, t+T_0]$ by

$$d(\bar{x}_t, K_i) \equiv \int_t^{t+T_0} x^{\mathrm{T}}(\tau)(Q + K_i^{\mathrm{T}}RK_i)x(\tau)\mathrm{d}\tau, \tag{17}$$

and we will use the same notation $d(\bar{x}_t, K_i)$ for the reinforcement signal over the interval $[t, t+T_0]$. Based on these notations and structures (14) is rewritten as

$$\bar{p}_{i+1}^{\mathrm{T}}\bar{x}_t = d(\bar{x}_t, K_i) + \bar{p}_i^{\mathrm{T}}\bar{x}_{t+T_0}. \tag{18}$$

The vector of unknown parameters is $\bar{p}_{i+1}$, and $\bar{x}_t$ acts as a regression vector. The right hand side target reinforcement function is measured based on the state trajectories over the time interval $[t, t+T_0]$ and the state value at $t+T_0$, $\bar{x}_{t+T_0}$.

The parameter vector $\bar{p}_{i+1}$ is found by minimizing, in the least-squares sense, the error between the target expected cost over the infinite horizon, which is the sum between the measured reinforcement over the time interval and the expected cost based on the present cost model, $d(\bar{x}_t, K_i) + \bar{p}_i^{\mathrm{T}}\bar{x}_{t+T_0}$, and the parameterized left hand side of (19). The solution can be obtained using the batch least squares or the recursive least squares algorithms.

The online value iteration algorithm is an online data-based approach that uses reinforcement learning ideas to find the solution to the algebraic Riccati equation with sign definite quadratic term. This algorithm does not require explicit knowledge of the model of the controlled system's drift dynamics, i.e., matrix $A$.

In the following section, we formulate the iterative algorithm that provides the saddle point solution to the two-player zero-sum differential game in terms of iterations on Riccati equations with sign definite quadratic term. At every step these Riccati equations can be solved by means of the online value iteration (i.e., CT-HDP) algorithm. The end result is an online algorithm that leads to the saddle point solution to the differential game while neither of the two players uses any knowledge on the drift dynamics of the environment.

## 3 Online approach to solve the differential game

The goal of this section is to describe the online data-based approach to the saddle point equilibrium solution for the two-player zero-sum differential game.

We will name the two players 'controller player' and 'disturbance player'. The solution to the game is found online while the game is played. We shall see that only one of the players is learning and optimizing his behavior strategy while the other is playing based on fixed policies. We shall say that the learning player is leading the adaptive procedure. His opponent is a passive player that decides on a fixed policy and maintains it constant during every learning phase. The passive player will change his behavior policy only based on information regarding his opponent's optimal strategy. In this case, the passive player will simply adopt his opponent's strategy as his own. It is noted that the passive player does not choose the best strategy that he could play to get a leading advantage in the sense of the Stackelberg game solution. Instead he will play the strategy that his opponent, i.e., the learning player, chooses as this will allow the learning process to end with the calculation of the desired Nash equilibrium solution to the game.

In the algorithm presented in this paper the reinforcement learning technique is employed only by the controller while the passive player is the disturbance.

Concisely, the game is played as follows:

1) The game starts while the disturbance player does not play.

2) The controller player plays the game without an opponent and uses reinforcement learning to find the optimal behavior that minimizes his costs, then informs his opponent on his new behavior policy.

3) The disturbance player starts playing using the behavior policy of his opponent.

4) The controller player corrects iteratively his own behavior using reinforcement knowledge such that his costs are again minimized, then informs his opponent on his new behavior policy.

5) Go to step 3) until the two policies are characterized by the same parameter values.

The two players execute successively steps 3) and 4) until the controller player can no longer lower his costs by changing his behavior policy. The saddle point equilibrium has been obtained.

Next, we give the formulation of the algorithm as iteration on Riccati equations. The online form of the algorithm will be presented after that. Throughout, we will make use of the notation $A_{\mathrm u}^{i-1} = A + B_1 B_1^{\mathrm T} P_{\mathrm u}^{i-1} - B_2 B_2^{\mathrm T} P_{\mathrm u}^{i-1}$.

**Algorithm 2**

1) Let $P_{\mathrm u}^0 = 0$.

2) Solve online the Riccati equation

$$P_{\mathrm u}^1 A + A^{\mathrm T} P_{\mathrm u}^1 - P_{\mathrm u}^1 B_2 B_2^{\mathrm T} P_{\mathrm u}^1 + C^{\mathrm T} C = 0. \tag{19}$$

3) Let $Z_{\mathrm u}^1 = P_{\mathrm u}^1$.

4) For $i \geqslant 2$ solve online the Riccati equation

$$\begin{aligned} &Z_{\mathrm u}^i A_{\mathrm u}^{i-1} + (A_{\mathrm u}^{i-1})^{\mathrm T} Z_{\mathrm u}^i - Z_{\mathrm u}^i B_2 B_2^{\mathrm T} Z_{\mathrm u}^i \\ &+ Z_{\mathrm u}^{i-1} B_1 B_1^{\mathrm T} Z_{\mathrm u}^{i-1} = 0. \end{aligned} \tag{20}$$

5) $P_{\mathrm u}^i = P_{\mathrm u}^{i-1} + Z_{\mathrm u}^i$.

At every step, the Riccati equations can be solved using the online data-based approach reviewed in Section 2.3 without using the exact knowledge on the drift term in the system dynamics.

**Algorithm 3** (Online form)

1) Let $P_{\mathrm u}^0 = 0$.

2) Let $K_{\mathrm u}^{0(0)}$ be zero, let $k = 0$.

a) Solve online

$$\begin{aligned} &x_t^{\mathrm T} P_{\mathrm u}^{0(k+1)} x_t \\ &= \int_t^{t+T_0} x_\tau^{\mathrm T} (C^{\mathrm T} C + (K_{\mathrm u}^{0(k)})^{\mathrm T} R K_{\mathrm u}^{0(k)}) x_\tau \mathrm d\tau \\ &\quad + x_{t+T_0}^{\mathrm T} P_{\mathrm u}^{0(k)} x_{t+T_0}. \end{aligned}$$

b) Update $K_{\mathrm u}^{0(k+1)} = -B_2^{\mathrm T} P_{\mathrm u}^{0(k+1)}$, $k = k + 1$.

c) Until $\|P_{\mathrm u}^{0(k)} - P_{\mathrm u}^{0(k-1)}\| < \varepsilon$.

3) $P_{\mathrm u}^1 = P_{\mathrm u}^{0(k)}$, $Z_{\mathrm u}^1 = P_{\mathrm u}^1$.

4) For $i \geqslant 2$,

a) Let $K_{\mathrm u}^{i-1(0)}$ be zero, let $k = 0$.

b) Solve online for the value of $Z_{\mathrm u}^{i(k+1)}$ using value iteration (i.e., CT-HDP),

$$\begin{aligned} &x_t^{\mathrm T} Z_{\mathrm u}^{i(k+1)} x_t \\ &= \int_t^{t+T_0} x_\tau^{\mathrm T} (Z_{\mathrm u}^{i-1} B_1 B_1^{\mathrm T} (Z_{\mathrm u}^{i-1})^{\mathrm T} + K_{\mathrm u}^{i-1(k)} (K_{\mathrm u}^{i-1(k)})^{\mathrm T}) x_\tau \mathrm d\tau \\ &\quad + x_{t+T_0}^{\mathrm T} Z_{\mathrm u}^{i(k)} x_{t+T_0}, \end{aligned}$$

where $x_\tau$ denotes the solution over the interval $[t, t+T_0]$ of the system $\dot x = (A + B_1 B_1^{\mathrm T} P_{\mathrm u}^{i-1} + B_2 K_{\mathrm u}^{i-1(k)}) x$ with initial condition $x_t$ (note that the integral term is not calculated by using numerical integration but is measured directly from the system by subtracting two measurements of the value associated with the game at time moments $t + T_0$ and $t$).

c) Update

$$K_{\mathrm u}^{i-1(k+1)} = K_{\mathrm u}^{i-1(k)} - B_2^{\mathrm T} Z_{\mathrm u}^{i(k+1)}, k = k + 1.$$

d) Until $\|Z_{\mathrm u}^{i(k)} - Z_{\mathrm u}^{i(k-1)}\| < \varepsilon$.

5) $Z_{\mathrm u}^i = Z_{\mathrm u}^{i(k)}$, $P_{\mathrm u}^i = P_{\mathrm u}^{i-1} + Z_{\mathrm u}^i$.

6) Until $\|P_{\mathrm u}^i - P_{\mathrm u}^{i-1}\| < \varepsilon_{\mathrm P}$.

From the perspective of two-player zero-sum games, the algorithm translates as follows:

1) Let the initial disturbance policy be zero, $w = 0$.

2) Let $K_{\mathrm u}^{0(0)}$ be an initial control policy for system (3) with zero disturbance $w = 0$, and let $k = 0$,

a) Update the value associated with the controller $K_{\mathrm u}^{0(k)}$.

b) Update the control policy, $k = k + 1$.

c) Until the controller with the highest value (i.e., minimum cost) has been obtained.

3) Update the disturbance policy using the gain of the control policy.

4) For $i \geqslant 2$,

a) Let $K_{\mathrm u}^{i-1(0)}$ be a control policy for system (3) with disturbance policy $w = B_1^{\mathrm T} P_{\mathrm u}^{i-1} x$, and let $k = 0$.

i) Find the added value associated with the change in the control policy.

ii) Update the control policy, $k = k + 1$.

iii) Repeat steps i) and ii) until the controller with the highest value has been obtained.

5) Go to step 3) until the control policy and disturbance policy have the same gain.

The adaptive critic structure that represents the implementation of this algorithm is given in Fig. 1.

An important aspect that is revealed by the adaptive critic structure is the fact that this ADP algorithm is now using three time scales:

- the continuous-time scale, represented by the full lines, which is connected with the continuous-time dynamics of the system, and the continuous-time computation performed by the two players;
- a discrete time scale given by $\Delta_1$, which is a multiple of the sample time $T$. This time scale is connected with the online learning procedure that is based on discrete time measured data;
- a slower discrete time scale characterized by the period $\Delta_2$, which is a multiple of $\Delta_1$. This time scale is connected with the update procedure of the disturbance policy, procedure that is performed once the controller policy has converged.

The values of the time periods $\Delta_1$ and $\Delta_2$ can be variable and are controlled by the critic that outputs the matrices $Z_{\mathrm{u}}^{i(k)}$ after every $\Delta_1$ interval, and $Z_{\mathrm{u}}^{i}$ after every $\Delta_2$ time interval.

One notes that both the control and the disturbance signals are obtained as sums of two signals: a base signal and a correction signal.

$$\begin{cases} u = u_{\mathrm{b}} + \hat{u} = -B_2^{\mathrm{T}} P_{\mathrm{u}}^{i-1} x - B_2^{\mathrm{T}} Z_{\mathrm{u}}^{i(k)} x \\ w = w_{\mathrm{b}} + \hat{w} = B_1^{\mathrm{T}} P_{\mathrm{u}}^{i-2} x + B_1^{\mathrm{T}} Z_{\mathrm{u}}^{i-1} x \end{cases} \tag{21}$$

The disturbance policy is constant over the time intervals $\Delta_2$, and is equal with $B_1^{\mathrm{T}} P_{\mathrm{u}}^{i-1}$, being described by the parameters of the base policy of the controller given by the matrix $P_{\mathrm{u}}^{i-1}$. The control policy is constant over the shorter time intervals $\Delta_1$, and is equal with $-B_2^{\mathrm{T}}(P_{\mathrm{u}}^{i-1} + Z_{\mathrm{u}}^{i(k)})$. The correction policy of the controller is the one that changes at every time interval $\Delta_1$ while the base policy remains constant over the larger interval $\Delta_2$.

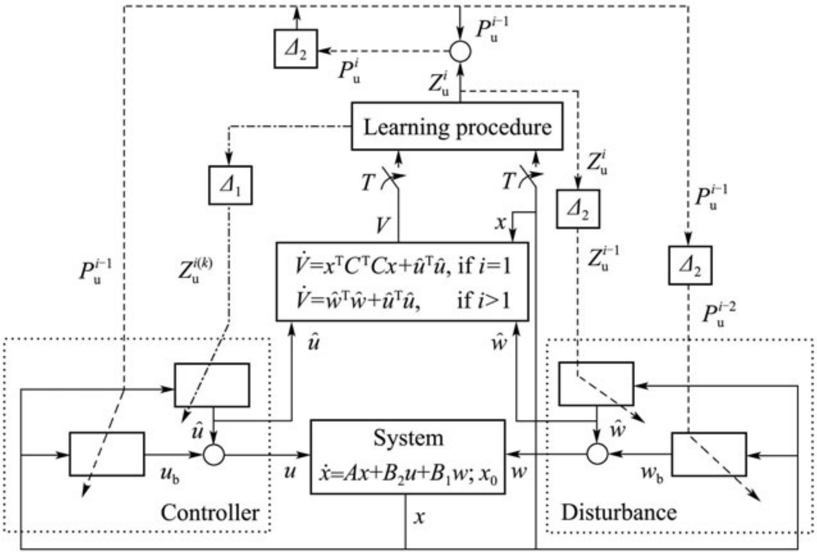

Fig. 1 Adaptive critic structure for the ADP game.

## 4 Online load-frequency controller design for a power system

This section presents the results that were 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 certain nominal operating points that are characterized by variations of the system load in a given range around a constant value. Although this assumption seems to have simplified the design problem of a load-frequency controller, a new problem appears from the fact that the parameters of the actual plant are not precisely known and only the range of these parameters can be determined. For this reason it is particularly advantageous to apply model free methods to obtain the optimal $\mathrm{H}_\infty$ controller for a given operating point of the power system.

The plant that was considered is the linear model of the power system presented in [21]. The purpose of the design method is to determine the control strategy that results in minimal control cost for maintaining the specified frequency set point in front of a maximal demand load change.

The state vector of the system is

$$x = [\Delta f \ \ \Delta P_{\mathrm{g}} \ \ \Delta X_{\mathrm{g}} \ \ \Delta E]^{\mathrm{T}}, \tag{22}$$

where the state components are the incremental frequency deviation $\Delta f$ (Hz), incremental change in generator output $\Delta P_{\mathrm{g}}$ (p.u. MW), incremental change in governor value position $\Delta X_{\mathrm{g}}$ (p.u. MW) and the incremental change in integral control $\Delta E$. The matrices of the model of the plant that are used in this simulation are

$$\begin{cases} A_{\mathrm{nom}} = \begin{bmatrix} -0.0665 & 8 & 0 & 0 \\ 0 & -3.663 & 3.663 & 0 \\ -6.86 & 0 & -13.736 & -13.736 \\ 0.6 & 0 & 0 & 0 \end{bmatrix}, \\ B_2 = [0 \ \ 0 \ \ 13.736 \ \ 0]^{\mathrm{T}}, \\ B_1 = [-8 \ \ 0 \ \ 0 \ \ 0]^{\mathrm{T}}. \end{cases} \tag{23}$$

The cost function parameter, i.e., the $C$ matrix, was chosen such that $Q = C^{\mathrm{T}}C$ is the identity matrix of appropriate dimensions.

Having the model of the system matrices one can easily determine the saddle point of the zero-sum game as

$$P_{\mathrm{u}}^{\infty} = \varPi = \begin{bmatrix} 0.6036 & 0.7398 & 0.0609 & 0.5877 \\ 0.7398 & 1.5438 & 0.1702 & 0.5978 \\ 0.0609 & 0.1702 & 0.0502 & 0.0357 \\ 0.5877 & 0.5978 & 0.0357 & 2.3307 \end{bmatrix}. \quad (24)$$

For the purpose of demonstrating the algorithm the closed loop system was excited with an initial condition of 1 incremental change in integral control $\Delta E$, the initial state of the system being $x_0 = [0 \;\; 0 \;\; 0 \;\; 1]$. The simulation was conducted using data obtained from the system at every 0.1 s. The value of the stop criterion $\varepsilon$ was $10^{-7}$.

In order to solve online for the values of the $P_{\mathrm{u}}^{i}$ matrix that parameterizes the cost function of the game, a least-squares problem of the sort described in Section 2.3 was setup before each iteration step 2) a) or 4) a) i) in Algorithm 3. Because there are 10 independent elements in the symmetric matrix $P$ the setup of the least-squares problem requires at least 10 measurements of the cost function associated with the given control policy and measurements of the system's states at the beginning and the end of each time interval, provided that there is enough excitation in the system. A least squares problem was solved after 25 data samples were acquired and thus the policy of the controller was updated every 2.5 s.

Fig. 2 presents the evolution of the parameters of the value of the game. It is clear that the cost function (i.e., critic) parameters converged to the optimal ones – indicated on the figure with star-shaped points. The high jumps in the values of the parameters presented in the figure are associated with the time moments when the disturbance player updates its policy.

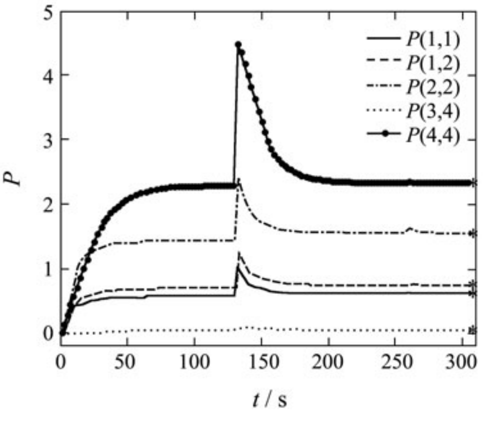

Fig. 2 Convergence of the cost function of the game using the ADP method.

The value of the game after 5 updates of the control policy is given by the matrix

$$P_{\mathrm{u}}^{5} = \begin{bmatrix} 0.6036 & 0.7399 & 0.0609 & 0.5877 \\ 0.7399 & 1.5440 & 0.1702 & 0.5979 \\ 0.0609 & 0.1702 & 0.0502 & 0.0357 \\ 0.5877 & 0.5979 & 0.0357 & 2.3307 \end{bmatrix}. \quad (25)$$

Comparing (25) with (24) one can see that the solution that was obtained using the ADP gaming method described in Section 3, which uses measurements from the system and does not require any knowledge on the matrix $A$, is very close to the exact solution that was obtained offline via numerical methods that require the exact model of the system.

The number of iterations until convergence of the $Z_{\mathrm{u}}^{i}$ for $i = \overline{1,5}$ are given in the following vector $\mathrm{iter} = [52 \;\; 51 \;\; 15 \;\; 3 \;\; 1]$. The large change in the parameters of the cost function, indicated in the figure at time 132.5 s, is determined by the first update of the policy of the disturbance player, i.e., after the sequence $Z_{\mathrm{u}}^{1(k)}$ has converged to $Z_{\mathrm{u}}^{1}$, and $P_{\mathrm{u}}^{2}$ was calculated.

It is important to note that the convergence to the equilibrium of the game is strongly connected with the convergence of the algorithm at steps 2) and 4) of Algorithm 3. If convergence at these steps is not obtained, and the disturbance policy is yet updated, then there are no guarantees that the saddle point solution to the game will still be obtained.

As discussed in [8], there are no guarantees that the closed loop dynamics of the game, characterized by $A_{\mathrm{u}}^{i-1} = A + B_1 B_1^{\mathrm{T}} P_{\mathrm{u}}^{i-1} - B_2 B_2^{\mathrm{T}} P_{\mathrm{u}}^{i-1}$, obtained after the disturbance has updated its policy will be stable. For this reason, the iteration that was employed at steps 2) and 4) of Algorithm 3 was chosen to be the CT-HDP algorithm described in [16], as this CT-HDP procedure does not require initialization with a stabilizing controller.

## 5 Conclusions

This paper introduced an online data-based approach that makes use of reinforcement learning techniques, namely the IRL method, to determine in an online fashion the solution for the two-player zero-sum differential game with linear dynamics. The result is based on a mathematical algorithm that solves offline the GARE and involves iterations on Riccati equations to build a sequence of controllers (and respectively disturbance policies). The sequence converges monotonically to the state-feedback saddle point solution to the two-player zero-sum differential game.

The Riccati equations that appear at each step of the iteration are solved online using measured data by means of a continuous-time value iteration (i.e., CT-HDP) algorithm. In this way the $\mathrm{H}_{\infty}$ state-feedback optimal controller, or the solution to the differential game, can be obtained online without using exact knowledge of the drift dynamics of the system, i.e., matrix $A$ in the system dynamics (3).

In order to give a clear presentation, here we considered the infinite horizon, state-feedback, linear-quadratic case of the problem. Ideas related with the extension of this result to the more general case of a game with nonlinear dynamics will be pursued in a future paper.

### References

[1] T. Basar, P. Bernhard. $\mathrm{H}_{\infty}$ *Optimal Control and Related Minimax Design Problems*. Boston: Birkhuser, 1995.

[2] T. Basar, G. J. Olsder. *Dynamic Noncooperative Game Theory (Classics in Applied Mathematics 23)*. 2nd ed. Philadelphia: SIAM, 1999.

[3] J. Doyle, K. Glover, P. Khargonekar, et al. State-space solutions to standard $\mathrm{H}_2$ and $\mathrm{H}_{\infty}$ control problems. *IEEE Transactions on Automatic Control*, 1989, 34(8): 831 – 847

[4] A. A. Stoorvogel. *The* $\mathrm{H}_{\infty}$ *Control Problem: A State Space Approach*. New York: Prentice Hall, 1992.

[5] K. Zhou, P. P. Khargonekar. An algebraic Riccati equation approach

to $H_\infty$ optimization. *Systems & Control Letters*, 1988, 11(2): 85 – 91.

[6] L. Cherfi, H. Abou-Kandil, H. Bourles. Iterative method for general algebraic Riccati equation. *Proceedings of International Conference on Automatic Control and System Engineering*, Cairo, Egypt, 2005: 85 – 88.

[7] T. Damm. *Rational Matrix Equations in Stochastic Control*. Berlin: Springer-Verlag, 2004.

[8] A. Lanzon, Y. Feng, B. D. O. Anderson, et al. Computing the positive stabilizing solution to algebraic Riccati equations with an indefinite quadratic term via a recursive method. *IEEE Transactions on Automation Control*, 2008, 53(10): 2280 – 2291.

[9] M. Abu-Khalaf, F. L. Lewis, J. Huang. Policy iterations and the Hamilton-Jacobi-Isaacs equation for $H_\infty$ state feedback control with input saturation. *IEEE Transactions on Automatic Control*, 2006, 51(12): 1989 – 1995.

[10] Y. Feng, B. D. O. Anderson, M. Rotkowitz. A game theoretic algorithm to compute local stabilizing solutions to HJBI equations in nonlinear $H_\infty$ control. *Automatica*, 2009, 45(4): 881 – 888.

[11] A. J. van der Schaft. $L_2$-gain analysis of nonlinear systems and nonlinear state feedback $H_\infty$ control. *IEEE Transactions on Automatic Control*, 1992, 37(6): 770 – 784.

[12] R. Sutton. Learning to predict by the method of temporal differences. *Machine Learning*, 1988, 3(1): 9 – 44.

[13] P. J. Werbos. Approximate dynamic programming for real-time control and neural modeling. D. White, D. Sofge, eds. *Handbook of Intelligent Control, Neural, Fuzzy, and, Adaptive Approaches*, New York: Van Nostrand, 1992: 493 – 525.

[14] C. Watkins. *Learning from Delayed Rewards*. Ph.D. thesis. Cambridge, U.K.: Cambridge University, 1989.

[15] Q. Wei, H. Zhang. A new approach to solve a class of continuous-time nonlinear quadratic zero-sum game using ADP. *Proceedings of IEEE International Conference on Networking, Sensing and Control*, New York: IEEE, 2008: 507 – 512.

[16] D. Vrabie, M. Abu-Khalaf, F. L. Lewis, et al. Continuous-time ADP for linear systems with partially unknown dynamics. *Proceedings of Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL)*, New York: IEEE, 2007: 247 – 253.

[17] D. Vrabie, O. Pastravanu, F. L. Lewis, et al. Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica*, 2009, 45(2): 477 – 484.

[18] F. L. Lewis, V. L. Syrmos. *Optimal Control*. New York: John Wiley & Sons, 1995.

[19] J. Speyer, D. Jacobson. *Primer on Optimal Control Theory*. Philadelphia: SIAM, 2010.

[20] J. W. Brewer. Kronecker products and matrix calculus in system theory. *IEEE Transactions on Circuit and System*, 1978, 25(9): 772 – 781.

[21] Y. Wang, R. Zhou, C. Wen. Robust load-frequency controller design for power systems. *IEE Proceedings – C: Generation, Transmission, and Distribution*, 1993, 140(1): 11 – 16.

**Draguna VRABIE** is a senior research scientist at the United Technologies Research Center, East Hartford, Connecticut. She received her B.S. in 2003 and M.S. degrees in 2004 from the Automatic Control and Computer Engineering Department, 'Gh. Asachi' Technical University of Iasi, and her Ph.D. in Electrical Engineering in 2009 from the University of Texas at Arlington. She is coauthor of the book 'Automatic Systems with PID Controllers', 3 book chapters, and 25 technical publications. She received the Best Paper award at the International Joint Conference on Neural Networks (IJCNN'10), Barcelona, Spain, 2010, and the Best Student award from the Automation & Robotics Research Institute, University of Texas at Arlington, in 2009. She serves as Associate Editor for the IEEE Transactions on Neural Networks, and the Transaction of the Institute of Measurement and Control. She serves on the Technical Program Committee for several international conferences. E-mail: vrabiedl@utrc.utc.com.

**Frank LEWIS** was born in Würzburg, Germany, subsequently studying in Chile and Gordonstoun School in Scotland. He obtained his Bachelor's degree in Physics/Electrical Engineering and Master's of Electrical Engineering degree at Rice University in 1971. He spent six years in the U.S. Navy, serving as Navigator aboard the frigate USS Trippe (FF-1075), and Executive Officer and Acting Commanding Officer aboard USS Salinan (ATF-161). In 1977, he received his Master's of Science degree in Aeronautical Engineering from the University of West Florida. In 1981, he obtained his Ph.D. degree at the Georgia Institute of Technology in Atlanta, where he was employed as a professor from 1981 to 1990. He is a professor of Electrical Engineering at the University of Texas at Arlington, where he was awarded the Moncrief-O'Donnell Endowed Chair in 1990 at the Automation & Robotics Research Institute. He is Fellow of the IEEE, Fellow of IFAC, Fellow of the U.K. Institute of Measurement & Control, and Member of the New York Academy of Sciences. Registered Professional Engineer in the State of Texas and Chartered Engineer, U.K. Engineering Council. Charter Member (2004) of the UTA Academy of Distinguished Scholars and Senior Research Fellow of the Automation & Robotics Research Institute. Founding Member of the Board of Governors of the Mediterranean Control Association. Has served as Visiting Professor at Democritus University in Greece, Hong Kong University of Science and Technology, Chinese University of Hong Kong, City University of Hong Kong, National University of Singapore, and Nanyang Technological University Singapore. Elected Guest Consulting Professor at Shanghai Jiao Tong University and South China University of Technology.

Current interests include intelligent control, distributed control on graphs, neural and fuzzy systems, wireless sensor networks, nonlinear systems, robotics, condition-based maintenance, microelectromechanical systems (MEMS) control, and manufacturing process control. Author of 6 U.S. patents, 222 journal papers, 47 chapters and encyclopedia articles, 333 refereed conference papers, and 14 books including 'Optimal Control, Optimal Estimation, Applied Optimal Control and Estimation, Aircraft Control and Simulation, Control of Robot Manipulators', 'Neural Network Control, High-Level Feedback Control with Neural Networks' and the IEEE reprint volume 'Robot Control'. Editor of Taylor & Francis Book Series on Automation & Control Engineering. Served/serves on many Editorial Boards including International Journal of Control, Neural Computing and Applications, Optimal Control & Methods, and International Journal of Intelligent Control Systems. Served as Editor for the flagship journal Automatica. Recipient of NSF Research Initiation Grant and continuously funded by NSF since 1982. Since 1991, he has received $7 million in funding from NSF, ARO, AFOSR and other government agencies, including significant DoD SBIR and industry funding. His SBIR program was instrumental in ARRI's receipt of the US SBA Tibbets Award in 1996. Received Fulbright Research Award 1988, American Society of Engineering Education F.E. Terman Award 1989, International Neural Network Society Gabor Award 2009, U.K. Inst Measurement & Control Honeywell Field Engineering Medal 2009, three Sigma Xi Research Awards, UTA Halliburton Engineering Research Award, UTA Distinguished Research Award, ARRI Patent Awards, various Best Paper Awards, IEEE Control Systems Society Best Chapter Award (as Founding Chairman of DFW Chapter), and National Sigma Xi Award for Outstanding Chapter (as President of UTA Chapter). Received Outstanding Service Award from the Dallas IEEE Section and selected as Engineer of the year by Ft. Worth IEEE Section. Listed in Ft. Worth Business Press Top 200 Leaders in Manufacturing. Appointed to NAE Committee on Space Station in 1995 and IEEE Control Systems Society Board of Governors in 1996. Selected in 1998 as an IEEE Control Systems Society Distinguished Lecturer. Received the 2010 IEEE Region 5 Outstanding Engineering Educator Award and the 2010 UTA Graduate Dean's Excellence in Doctoral Mentoring Award. E-mail: lewis@arri.uta.edu.