Available online at www.sciencedirect.com
ScienceDirect
automatica
Automatica 43 (2007) 473–481
www.elsevier.com/locate/automatica

Brief paper

Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control ☆

Asma Al-Tamimi*, Frank L. Lewis, Murad Abu-Khalaf

Automation and Robotics Research Institute, The University of Texas at Arlington, Texas 76118, USA

Received 20 October 2005; received in revised form 12 September 2006; accepted 15 September 2006
Available online 24 January 2007

Abstract

In this paper, the optimal strategies for discrete-time linear system quadratic zero-sum games related to the H-infinity optimal control problem are solved in forward time without knowing the system dynamical matrices. The idea is to solve for an action dependent value function $Q(x, u, w)$ of the zero-sum game instead of solving for the state dependent value function $V(x)$ which satisfies a corresponding game algebraic Riccati equation (GARE). Since the state and actions spaces are continuous, two action networks and one critic network are used that are adaptively tuned in forward time using adaptive critic methods. The result is a Q-learning approximate dynamic programming (ADP) model-free approach that solves the zero-sum game forward in time. It is shown that the critic converges to the game value function and the action networks converge to the Nash equilibrium of the game. Proofs of convergence of the algorithm are shown. It is proven that the algorithm ends up to be a model-free iterative algorithm to solve the GARE of the linear quadratic discrete-time zero-sum game. The effectiveness of this method is shown by performing an H-infinity control autopilot design for an F-16 aircraft.

Keywords: Adaptive critics; Approximate dynamic programming; Zero-sum games; Policy iterations; H_∞ optimal control; Q-function; Q-learning; Adaptive control

1. Introduction

This paper is concerned with the application of approximate dynamic programming techniques (ADP) to the discrete-time linear quadratic zero-sum game that appearing in the H_∞ optimal control problem (Başar & Bernhard, 1995), where is the disturbance has finite energy. Approximate dynamic programming is an approach to solve dynamical programming problems. Approximate dynamic programming was proposed by Werbos (1991), Barto, Sutton, and Anderson (1983), Howard (1960), Watkins (1989), Bertsekas and Tsitsiklis (1996), and others to solve optimal control problems forward-in-time. In ADP, one combines adaptive critics, a reinforcement learning technique, with dynamic programming.

Several approximate dynamic programming schemes appear in literature. Howard (1960) proposed iterations in the policy space in the framework of stochastic decision theory. Bradtke, Ydestie, and Barto (1994), implemented a Q-learning policy iteration method for the discrete-time linear quadratic optimal control problem, while our is concerned with zero-sum games. In addition, the way we handle exploration noise is different in order to obtain convergence results of the associated Riccati equation (GARE). Hagen and Krose (1998) discussed the relation between the Q-learning policy iteration method and model-based adaptive control with system identification. Werbos (1992) classified approximate dynamic programming approaches into four main schemes: heuristic dynamic programming (HDP), dual heuristic dynamic programming (DHP), action dependent heuristic dynamic programming (ADHDP), also known as Q-learning Watkins (1989), and action dependent dual heuristic dynamic

☆ This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by associate Editor Derong Liu under the direction of Editor M. Krstic. This research was supported by the National Science Foundation ECS-0501451, the Army Research Office W91NF-05-1-0314.

* Corresponding author. Tel./fax: +817 272 5938.
E-mail addresses: altamimi@arri.uta.edu (A. Al-Tamimi), lewis@uta.edu (F.L. Lewis), mabukhalaf@gmail.com (M. Abu-Khalaf).

0005-1098/$ - see front matter
doi:10.1016/j.automatica.2006.09.019

programming (ADDHP). Prokhorov and Wunsch (1997) developed new approximate dynamic programming schemes known as globalized-DHP (GDHP) and ADGDHP. Landelius (1997) applied HDP, DHP, ADHDP and ADDHP techniques to the discrete-time linear quadratic optimal control problem. The current status of work on approximate dynamic programming is given in Si, Barto, Powell, and Wunsch (2004). See also Bertsekas and Tsitsiklis (1996); He and Jagannathan (2005), Si and Wang (2001) and Cao (2002).

In this paper, Q-learning is used since as will be seen in the paper, it allows model-free tuning of the action and critic networks. That is, this method does not require knowledge of the plant model. In Landelius (1997), no initial stable control policy for the optimal control problem is required, however the requirement of exploration noise is not studied.

This problem has been solved off-line using the dynamic programming principle Başar and Bernhard (1995); Başar and Olsder (1999), Lewis (1995). An off-line neural net policy iterations solution was given by Abu-Khalaf, Lewis, and Huang (2004) for the continuous-time case.

The importance of this paper stems from the fact that we propose game-theoretic adaptive critics that create controllers that learn to co-exist with an L_2-gain disturbance signal Başar and Bernhard (1995); Başar and Olsder (1999). In control system design, this is a two-player zero-sum game problem that corresponds to the well-known H_∞ control problem.

An H_∞ control F-16 aircraft autopilot design example is given to show the practical effectiveness of the ADP techniques.

2. Q-function setup for discrete-time linear quadratic zero-sum games installation

In this section, we formulate Bellman's optimality principle for the zero-sum-game using the concept of Q-functions (Watkins, 1989; Werbos, 1990) instead of the standard value functions used elsewhere. Consider the following discrete-time linear system

$$x_{k+1} = Ax_k + Bu_k + Ew_k, \quad y_k = x_k, \tag{1}$$

where $x \in R^n$, $y \in R^p$, $u_k \in R^{m_1}$ is the control input and $w_k \in R^{m_2}$ is the disturbance input. Also consider the infinite-horizon value function

$$V^*(x_k) = \min_{u_i} \max_{w_i} \sum_{i=k}^{\infty} [x_i^{\mathrm{T}} R x_i + u_i^{\mathrm{T}} u_i - \gamma^2 w_i^{\mathrm{T}} w_i] \tag{2}$$

for a prescribed fixed value of γ. In the H-infinity control problem, γ is an upper bound on the desired L_2 gain disturbance attenuation (Başar & Bernhard, 1995; Lin & Byrnes, 1996).

It is desired to find the optimal control u_k^* and the worst case disturbance w_k^*. Here the class of strictly feedback stabilizing policies is considered (Başar & Olsder (1999)). Using the dynamic programming principle, the optimization problem in Eqs. (1) and (2) can be written as

$$\begin{aligned} V^*(x_k) &= \min_{u_k} \max_{w_k} (r(x_k, u_k, w_k) + V(x_{k+1})) \\ &= \max_{w_k} \min_{u_k} (r(x_k, u_k, w_k) + V^*(x_{k+1})). \end{aligned} \tag{3}$$

If we assume that there exists a solution to the GARE that is strictly feedback stabilizing, then it can be shown, see Jacobson (1977), that the policies are in saddle-point equilibrium, i.e. *minimax* is equal to *maximin*, in the restricted class of feedback stabilizing policies under which $x_k \to 0$ as $k \to \infty$ for all $x_0 \in R^n$. (See Başar & Bernhard, 1995, p. 138; Başar & Olsder, 1999, p. 340; Jacobson, 1977; Mageirou, 1976.)

Assuming that the game has a value and is solvable, then it is known that the value function is quadratic in the state and is given as

$$V^*(x_k) = x_k^{\mathrm{T}} P x_k, \tag{4}$$

where $P \geqslant 0$ and satisfies the GARE (Lin & Byrnes, 1996; Stoorvogle & Weeren, 1994), which is given as

$$\begin{aligned} P = {} & A^{\mathrm{T}}PA + R - [A^{\mathrm{T}}PB \;\; A^{\mathrm{T}}PE] \\ & \times \begin{bmatrix} I + B^{\mathrm{T}}PB & B^{\mathrm{T}}PE \\ E^{\mathrm{T}}PB & E^{\mathrm{T}}PE - \gamma^2 I \end{bmatrix}^{-1} \begin{bmatrix} B^{\mathrm{T}}PA \\ E^{\mathrm{T}}PA \end{bmatrix}. \end{aligned} \tag{5}$$

Note that the GARE in Eq. (5) will be the algebraic Riccati equation (ARE) if $E = 0$. The optimal policies are $u_k^* = Lx_k$ and $w_k^* = Kx_k$ where

$$\begin{aligned} L = {} & (I + B^{\mathrm{T}}PB - B^{\mathrm{T}}PE(E^{\mathrm{T}}PE - \gamma^2 I)^{-1}E^{\mathrm{T}}PB)^{-1} \\ & \times (B^{\mathrm{T}}PE(E^{\mathrm{T}}PE - \gamma^2 I)^{-1}E^{\mathrm{T}}PA - B^{\mathrm{T}}PA), \end{aligned} \tag{6}$$

$$\begin{aligned} K = {} & (E^{\mathrm{T}}PE - \gamma^2 I - E^{\mathrm{T}}PB(I + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PE)^{-1} \\ & \times (E^{\mathrm{T}}PB(I + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA - E^{\mathrm{T}}PA). \end{aligned} \tag{7}$$

Note that if P is known, then one still requires the system model to compute the controller gains.

In order to have a unique feedback saddle-point in the class of strictly feedback stabilizing policies, the inequalities in (8) and (9) should be satisfied, (Başar & Bernhard, 1995),

$$I - \gamma^{-2}E^{\mathrm{T}}PE > 0, \tag{8}$$

$$I + B^{\mathrm{T}}PB > 0. \tag{9}$$

Note that the inverse matrices in (6) and (7) exist due to (8) and (9).

In this paper, we extend the concept of Q-functions to zero-sum games that are continuous in the state and action space as in (3). The optimal action dependent value function Q^* of the zero-sum game is then defined to be

$$\begin{aligned} Q^*(x_k, u_k, w_k) &= r(x_k, u_k, w_k) + V^*(x_{k+1}) \\ &= [x_k^{\mathrm{T}} \;\; u_k^{\mathrm{T}} \;\; w_k^{\mathrm{T}}] H [x_k^{\mathrm{T}} \;\; u_k^{\mathrm{T}} \;\; w_k^{\mathrm{T}}]^{\mathrm{T}}, \end{aligned} \tag{10}$$

where H is the matrix associated with P that solves GARE, and is derived as

$$\begin{bmatrix} x_k \\ u_k \\ w_k \end{bmatrix}^{\mathrm{T}} H \begin{bmatrix} x_k \\ u_k \\ w_k \end{bmatrix} = r(x_k, u_k, w_k) + V^*(x_{k+1})$$
$$= x_k^{\mathrm{T}} R x_k + u_k^{\mathrm{T}} u_k - \gamma^2 w_k^{\mathrm{T}} w_k + x_{k+1}^{\mathrm{T}} P x_{k+1}$$
$$= \begin{bmatrix} x_k \\ u_k \\ w_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} R & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} x_k \\ u_k \\ w_k \end{bmatrix} + \begin{bmatrix} x_k \\ u_k \\ w_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} A^{\mathrm{T}} \\ B^{\mathrm{T}} \\ E^{\mathrm{T}} \end{bmatrix} P \begin{bmatrix} A^{\mathrm{T}} \\ B^{\mathrm{T}} \\ E^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} x_k \\ u_k \\ w_k \end{bmatrix} \quad (11)$$

so H can be written as

$$\begin{pmatrix} H_{xx} & H_{xu} & H_{xw} \\ H_{ux} & H_{uu} & H_{uw} \\ H_{wx} & H_{wu} & H_{ww} \end{pmatrix} = \begin{bmatrix} A^{\mathrm{T}}PA + R & A^{\mathrm{T}}PB & A^{\mathrm{T}}PE \\ B^{\mathrm{T}}PA & B^{\mathrm{T}}PB + I & B^{\mathrm{T}}PE \\ E^{\mathrm{T}}PA & E^{\mathrm{T}}PB & E^{\mathrm{T}}PE - \gamma^2 I \end{bmatrix}. \quad (12)$$

The optimal action dependent game value function $Q^*(x_k, u_k, w_k)$ is equal to the game value function $V^*(x_k)$ when the policies u_k, w_k are optimal. Then one has

$$V^*(x_k) = \min_{u_k} \max_{w_k} Q^*(x_k, u_k, w_k)$$
$$= \min_{u_k} \max_{w_k} [x_k^{\mathrm{T}} \; u_k^{\mathrm{T}} \; w_k^{\mathrm{T}}] H [x_k^{\mathrm{T}} \; u_k^{\mathrm{T}} \; w_k^{\mathrm{T}}]^{\mathrm{T}}$$
$$= Q^*(x_k, u_k^*, w_k^*) \quad (13)$$

therefore the relation between P and H can be obtained by equating (13) and (4)

$$P = [I \;\; L^{\mathrm{T}} \;\; K^{\mathrm{T}}] H [I \;\; L^{\mathrm{T}} \;\; K^{\mathrm{T}}]^{\mathrm{T}}. \quad (14)$$

Substituting (14) in (11), H also can be written as

$$H = G + \begin{bmatrix} A & B & E \\ LA & LB & LE \\ KA & KB & KE \end{bmatrix}^{\mathrm{T}} H \begin{bmatrix} A & B & E \\ LA & LB & LE \\ KA & KB & KE \end{bmatrix}, \quad (15)$$

$$G = \begin{bmatrix} R & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & -\gamma^2 \end{bmatrix}$$

which can be related to

$$Q^*(x_k, u_k, w_k) = r(x_k, u_k, w_k) + Q^*(x_{k+1}, u_{k+1}^*, w_{k+1}^*). \quad (16)$$

Eqs. (15) and (16) are an action dependent version of (3) and (5) in terms of the H. Similarly using (12), the gains of the optimal strategies can be written in terms of H as

$$L = (H_{uu} - H_{uw} H_{ww}^{-1} H_{wu})^{-1} (H_{uw} H_{ww}^{-1} H_{wx} - H_{ux}), \quad (17)$$

$$K = (H_{ww} - H_{wu} H_{uu}^{-1} H_{uw})^{-1} (H_{wu} H_{uu}^{-1} H_{ux} - H_{wx}). \quad (18)$$

Eqs. (17) and (18) depend only on the H matrix, and they are the main equations needed in the algorithm to be proposed to find the control and disturbance gains. Note that if H is known, then the system model is not needed to compute the controller gains.

In the next section, we show how to develop an algorithm to learn the Q-functions (i.e. the H matrix) of a given zero-sum game. This model-free Q-learning algorithm allows for solving the GARE equation online without requiring the knowledge of the plant model.

3. Model-free online tuning based on the Q-learning algorithm

In this section, we use the Q-function of Section 2 to develop a Q-learning algorithm to solve for the DT zero-sum game H matrix that does not require the system dynamical matrices. In the Q-learning approach, a parametric structure is used to approximate Q-function of the current control policy. Then the certainty equivalent principle is used to improve the policy of the action network.

3.1. Derivation of Q-learning for zero-sum games

In the Q-learning, one starts with an initial Q-function $Q_0(x, u, w) \geqslant 0$ that is not necessarily optimal, and then finds $Q_1(x, u, w)$ by solving Eq. (19) with $i = 0$ as

$$Q_{i+1}(x_k, u_k, w_k) = \left\{ x_k^{\mathrm{T}} R x_k + u_k^{\mathrm{T}} u_k - \gamma^2 w_k^{\mathrm{T}} w_k + \min_{u_{k+1}} \max_{w_{k+1}} Q_i(x_{k+1}, u_{k+1}, w_{k+1}) \right\}$$
$$= \{x_k^{\mathrm{T}} R x_k + u_k^{\mathrm{T}} u_k - \gamma^2 w_k^{\mathrm{T}} w_k + V_i(x_{k+1})\}$$
$$= \{x_k^{\mathrm{T}} R x_k + u_k^{\mathrm{T}} u_k - \gamma^2 w_k^{\mathrm{T}} w_k + V_i(A x_k + B u_k + E w_k)\} \quad (19)$$

then applying the following incremental optimization on the Q function as

$$\min_{u_k} \max_{w_k} Q_{i+1}(x_k, u_k, w_k) = \min_{u_k} \max_{w_k} [x_k^{\mathrm{T}} \; u_k^{\mathrm{T}} \; w_k^{\mathrm{T}}] H_{i+1} [x_k^{\mathrm{T}} \; u_k^{\mathrm{T}} \; w_k^{\mathrm{T}}]^{\mathrm{T}}.$$

Note that in Eq. (19), the Q-function is given for the any policy u and w. According to (17) and (18) the corresponding state feedback policy updates are given by

$$L_i = (H_{uu}^i - H_{uw}^i {H_{ww}^i}^{-1} H_{wu}^i)^{-1} (H_{uw}^i {H_{ww}^i}^{-1} H_{wx}^i - H_{ux}^i),$$
$$K_i = (H_{ww}^i - H_{wu}^i {H_{uu}^i}^{-1} H_{uw}^i)^{-1} \times (H_{wu}^i {H_{uu}^i}^{-1} H_{ux}^i - H_{wx}^i), \quad (20)$$

$$u_i(x_k) = L_i x_k, \quad w_i(x_k) = K_i x_k. \quad (21)$$

Note that since $Q_i(x, u, w)$ is not initially optimal, the improved policies $u_i(x_k)$ and $w_i(x_k)$ use the certainty equivalence principle. Note that to update the action networks, the plant model A, B and E matrices are not needed since only the H matrix is required. To develop solutions to (19) forward in time that do not need the system matrices, one can substitute (21) in (19) to obtain the following recurrence relation on i

$$\begin{aligned} &Q_{i+1}(x_k, u_i(x_k), w_i(x_k)) \\ &\quad = x_k^{\mathrm{T}} R x_k + u_i^{\mathrm{T}}(x_k) u_i(x_k) - \gamma^2 w_i^{\mathrm{T}}(x_k) w_i(x_k) \\ &\qquad + [x_{k+1}^{\mathrm{T}} \;\; u_i^{\mathrm{T}}(x_{k+1}) \;\; w_i^{\mathrm{T}}(x_{k+1})] \\ &\qquad \times H_i [x_{k+1}^{\mathrm{T}} \;\; u_i^{\mathrm{T}}(x_{k+1}) \;\; w_i^{\mathrm{T}}(x_{k+1})]^{\mathrm{T}} \end{aligned} \tag{22}$$

that is used to solve for the optimal Q-function forward in time.

The idea to solve for Q_{i+1}, then once determined, one repeats the same process for $i = 0, 1, 2, \ldots$. In this paper, it is shown that $Q_{i+1}(x_k, u_i(x_k), w_i(x_k)) \to Q^*(x_k, u_k, w_k)$ as $i \to \infty$, which means $H_i \to H$, $L_i \to L$ and $K_i \to K$.

A parametric structure is used to approximate the actual $Q_i(x, u, w)$. Similarly, parametric structures are used to obtain approximate closed-form representations of the two action networks $\hat{u}(x, L)$ and $\hat{w}(x, K)$. Since in this paper linear quadratic zero-sum games are considered, the Q-function is quadratic in the state and the policies. Moreover, the two action networks are linear in the state. Therefore, a natural choice of these parameter structures is given as

$$\hat{u}_i(x) = L_i x, \tag{23}$$

$$\hat{w}_i(x) = K_i x, \tag{24}$$

$$\hat{Q}(\bar{z}, h_i) = z^{\mathrm{T}} H_i z = h_i^{\mathrm{T}} \bar{z}, \tag{25}$$

where $z = [x^{\mathrm{T}} \;\; u^{\mathrm{T}} \;\; w^{\mathrm{T}}]^{\mathrm{T}}$ $z \in R^{n+m_1+m_2=q}$, $\bar{z} = (z_1^2, \ldots, z_1 z_q, z_2^2, z_2 z_3, \ldots, z_{q-1} z_q, z_q^2)$ is the Kronecker product quadratic polynomial basis vector (Brewer, 1978), and $h = v(H)$ with $v(\cdot)$ a vector function that acts on $q \times q$ matrices and gives a $q(q+1)/2 \times 1$ column vector. The output of $v(\cdot)$ is constructed by stacking the columns of the squared matrix into a one-column vector with the off-diagonal elements summed as $H_{ij} + H_{ji}$. In the linear case, the parametric structures ((23)–(25)) give an exact closed-form representation of the functions in (22). Note that (23) and (24) are updated using (20). To solve for Q_{i+1} in (22), the right-hand side of (22) is written as

$$\begin{aligned} d(z_k(x_k), H_i) = {} & x_k^{\mathrm{T}} R x_k + \hat{u}_i(x_k)^{\mathrm{T}} \hat{u}_i(x_k) - \gamma^2 \hat{w}_i(x_k)^{\mathrm{T}} \hat{w}_i(x_k) \\ & + Q_i(x_{k+1}, \hat{u}_i(x_{k+1}), \hat{w}_i(x_{k+1})) \end{aligned} \tag{26}$$

which can be thought of as the desired target function to which one needs to fit $\hat{Q}(z, h_{i+1})$ in least-squares sense to find h_{i+1} such that

$$h_{i+1}^{\mathrm{T}} \bar{z}(x_k) = d(\bar{z}(x_k), h_i). \tag{27}$$

The parameter vector h_{i+1} is found by minimizing the error between the target value function (26) and (25) in a least-squares sense over a compact set Ω,

$$h_{i+1} = \arg\min_{h_{i+1}} \left\{ \int_{\Omega} |h_{i+1}^{\mathrm{T}} \bar{z}(x_k) - d(\bar{z}(x_k), h_i)|^2 \, \mathrm{d}x_k \right\}. \tag{28}$$

Solving the least-squares problem one obtains

$$h_{i+1} = \left(\int_{\Omega} \bar{z}(x_k) \bar{z}(x_k)^{\mathrm{T}} \, \mathrm{d}x \right)^{-1} \int_{\Omega} \bar{z}(x_k) \, \mathrm{d}(\bar{z}(x_k), h_i) \, \mathrm{d}x, \tag{29}$$

$$\begin{aligned} z(x_k) &= [x_k^{\mathrm{T}} \;\; (\hat{u}_i(x_k))^{\mathrm{T}} \;\; (\hat{w}_i(x_k))^{\mathrm{T}}]^{\mathrm{T}} \\ &= (x_k^{\mathrm{T}} [I \;\; L_i^{\mathrm{T}} \;\; K_i^{\mathrm{T}}]^{\mathrm{T}})^{\mathrm{T}}. \end{aligned} \tag{30}$$

Note that $\hat{u}_i$ and $\hat{w}_i$ are linearly dependent on x_k, see (23) and (24), therefore $\int_{\Omega} \bar{z}(x_k) \bar{z}(x_k)^{\mathrm{T}} \, \mathrm{d}x_k$ is never invertible, which means that the least-squares problem (28), (29) will never be solvable. To overcome this problem one, exploration noise is added to both inputs in (21) to obtain

$$\hat{u}_{ei}(x_k) = L_i x_k + n_{1k}, \quad \hat{w}_{ei}(x_k) = K_i x_k + n_{2k}, \tag{31}$$

where $n_1(0, \sigma_1)$ and $n_2(0, \sigma_2)$ are zero-mean exploration noise with variances σ_1^2 and σ_2^2, respectively, therefore $z(x_k)$ in (30) becomes

$$z(x_k) = \begin{bmatrix} x_k \\ \hat{u}_{ei}(x_k) \\ \hat{w}_{ei}(x_k) \end{bmatrix} = \begin{bmatrix} x_k \\ L_i x_k + n_{1k} \\ K_i x_k + n_{2k} \end{bmatrix} = \begin{bmatrix} x_k \\ L_i x_k \\ K_i x_k \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ n_{1k} \\ n_{2k} \end{bmatrix}.$$

Evaluating (27) at enough points $p1, p2, p3, \ldots \in \Omega$, one has

$$h_{i+1} = (Z Z^{\mathrm{T}})^{-1} Z Y, \tag{32}$$

$$Z = [\bar{z}(p1) \;\; \bar{z}(p2) \;\; \ldots \;\; \bar{z}(pN)],$$

$$Y = [d(\bar{z}(p1), h_i) \;\; d(\bar{z}(p2), h_i) \;\; \ldots \;\; d(\bar{z}(pN), h_i)]^{\mathrm{T}}.$$

Here the target in Eq. (26) becomes

$$\begin{aligned} \mathrm{d}(z_k(x_k), H_i) = {} & x_k^{\mathrm{T}} R x_k + \hat{u}_{ei}(x_k)^{\mathrm{T}} \hat{u}_{ei}(x_k) - \gamma^2 \hat{w}_{ei}(x_k)^{\mathrm{T}} \hat{w}_{ei}(x_k) \\ & + Q_i(x_{k+1}, \hat{u}_i(x_{k+1}), \hat{w}_i(x_{k+1})). \end{aligned} \tag{33}$$

with $\hat{u}_i$ and $\hat{w}_i$ used for Q_i instead of $\hat{u}_{ei}$ and $\hat{w}_{ei}$. The invertibility of the matrix in (32) is therefore guaranteed by the excitation condition.

3.2. Online implementation of the Q-learning algorithm

The least-squares problem in (32) can be solved in real-time by collecting enough data points generated from $d(z_k, h_i)$ in (33). This requires one to have knowledge of the state information x_k, x_{k+1} as the dynamics evolve in time, and also of the reward function $r(z_k) = x_k^{\mathrm{T}} R x_k + \hat{u}_{ei}(x_k)^{\mathrm{T}} \hat{u}_{ei}(x_k) - \gamma^2 \hat{w}_{ei}(x_k)^{\mathrm{T}} \hat{w}_{ei}(x_k)$ and Q_i. This can be determined by simulation, or in real-time applications, by observing the states on-line.

To satisfy the excitation condition of the least-squares problem, one needs to have the number of collected points N at least $N \geqslant q(q+1)/2$, where $q = n + m_1 + m_2$ is the number of states and both policies, control and disturbance. In online implementation of the least-squares problem, Y and Z matrices

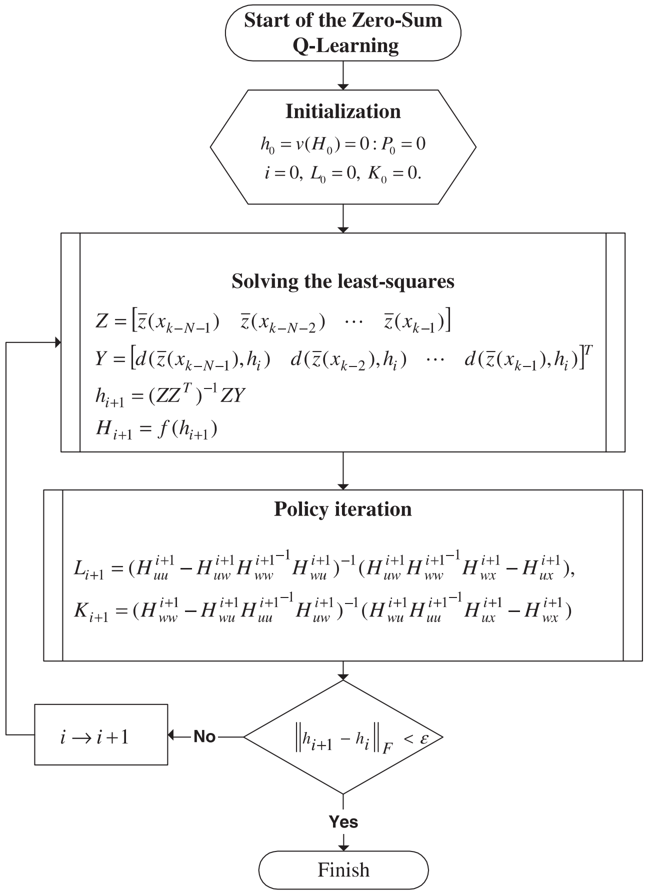


Fig. 1. Zero-sum games Q-learning.

are obtained in real time as

$$Z = [\bar{z}(x_{k-N-1})\ \ \bar{z}(x_{k-N-2})\ \ \ldots\ \ \bar{z}(x_{k-1})],$$

$$Y = [d(\bar{z}(x_{k-N-1}), h_i)\ \ d(\bar{z}(x_{k-2}), h_i)\ \ \ldots\ \ d(\bar{z}(x_{k-1}), h_i)]^{\mathrm{T}}. \tag{34}$$

One can also solve (34) recursively using the well-known recursive least-squares technique. In that case, the excitation condition is replaced by the persistency of excitation condition,

$$\varepsilon_0 I \leqslant \frac{1}{\alpha} \sum_{k=1}^{\alpha} \bar{z}_{k-t} \bar{z}_{k-t}^{\mathrm{T}} \leqslant \varepsilon_1 I,$$

for all $k > \alpha_0$, $\alpha > \alpha_0$, with $\varepsilon_0 \leqslant \varepsilon_1$, ε_0 and ε_1 positive integers and $\varepsilon_0 \leqslant \varepsilon_1$. The on-line Q-learning algorithm developed in this paper is summarized in the flowchart shown in Fig. 1.

This algorithm for zero-sum games follows by iterating between (20) and (34). In the remainder of this section, it will be shown that this policy iteration technique will cause Q_i to converge to the optimal Q^*.

3.3. Convergence of the zero-sum game Q-learning

We now prove that the proposed Q-learning algorithm for zero-sum games converges to the optimal policies. Some preliminary Lemmas are needed.

Lemma 1. *Iterating on Eqs.* (20), *and* (34) *is equivalent to*

$$H_{i+1} = G + \begin{bmatrix} A & B & E \\ L_i A & L_i B & L_i E \\ K_i A & K_i B & K_i E \end{bmatrix}^{\mathrm{T}} \times H_i \begin{bmatrix} A & B & E \\ L_i A & L_i B & L_i E \\ K_i A & K_i B & K_i E \end{bmatrix}. \tag{35}$$

Proof. Since Eq. (33) is equivalent to

$$d(\bar{z}_k(x_k), h_i) = \bar{z}_k^{\mathrm{T}} v \left(G + \begin{bmatrix} A & B & E \\ L_i A & L_i B & L_i E \\ K_i A & K_i B & K_i E \end{bmatrix}^{\mathrm{T}} \times H_i \begin{bmatrix} A & B & E \\ L_i A & L_i B & L_i E \\ K_i A & K_i B & K_i E \end{bmatrix} \right)$$

then using the Kronecker products, the least-squares (34) becomes

$$h_{i+1} = \underbrace{(ZZ^{\mathrm{T}})^{-1}(ZZ)}_{I}\, v \left(G + \begin{bmatrix} A & B & E \\ L_i A & L_i B & L_i E \\ K_i A & K_i B & K_i E \end{bmatrix}^{\mathrm{T}} \times H_i \begin{bmatrix} A & B & E \\ L_i A & L_i B & L_i E \\ K_i A & K_i B & K_i E \end{bmatrix} \right),$$

where v is the vectorized function in Kronecker products.

Since the matrix H_{i+1} reconstructed from h_{i+1} is symmetric, iterating on h_i is equivalent to

$$H_{i+1} = G + \begin{bmatrix} A & B & E \\ L_i A & L_i B & L_i E \\ K_i A & K_i B & K_i E \end{bmatrix}^{\mathrm{T}} \times H_i \begin{bmatrix} A & B & E \\ L_i A & L_i B & L_i E \\ K_i A & K_i B & K_i E \end{bmatrix}. \qquad \square$$

Lemma 2. *The matrices H_{i+1}, L_{i+1} and K_{i+1} can be written*

$$H_{i+1} = \begin{bmatrix} A^{\mathrm{T}} P_i A + R & A^{\mathrm{T}} P_i B & A^{\mathrm{T}} P_i E \\ B^{\mathrm{T}} P_i A & B^{\mathrm{T}} P_i B + I & B^{\mathrm{T}} P_i E \\ E^{\mathrm{T}} P_i A & E^{\mathrm{T}} P_i B & E^{\mathrm{T}} P_i E - \gamma^2 I \end{bmatrix}, \tag{36}$$

$$L_{i+1} = (I + B^{\mathrm{T}} P_i B - B^{\mathrm{T}} P_i E (E^{\mathrm{T}} P_i E - \gamma^2 I)^{-1} E^{\mathrm{T}} P_i B)^{-1} \times (B^{\mathrm{T}} P_i E (E^{\mathrm{T}} P_i E - \gamma^2 I)^{-1} E^{\mathrm{T}} P_i A - B^{\mathrm{T}} P_i A), \tag{37}$$

$$K_{i+1} = (E^{\mathrm{T}} P_i E - \gamma^2 I - E^{\mathrm{T}} P_i B (I + B^{\mathrm{T}} P_i B)^{-1} B^{\mathrm{T}} P_i E)^{-1} \times (E^{\mathrm{T}} P_i B (I + B^{\mathrm{T}} P_i B)^{-1} B^{\mathrm{T}} P_i A - E^{\mathrm{T}} P_i A), \tag{38}$$

where P_i is given as

$$P_i = [I\ \ L_i^{\mathrm{T}}\ \ K_i^{\mathrm{T}}] H_i [I\ \ L_i^{\mathrm{T}}\ \ K_i^{\mathrm{T}}]^{\mathrm{T}}. \tag{39}$$

Proof. Eq. (35) in Lemma 1 can be written as

$$H_{i+1}=G+\begin{bmatrix} A & B & E \\ L_iA & L_iB & L_iE \\ K_iA & K_iB & K_iE \end{bmatrix}^{\mathrm{T}} H_i \begin{bmatrix} A & B & E \\ L_iA & L_iB & L_iE \\ K_iA & K_iB & K_iE \end{bmatrix}$$
$$=G+[A\ \ B\ \ E]^{\mathrm{T}}[I\ \ L_i^{\mathrm{T}}\ \ K_i^{\mathrm{T}}]H_i[I\ \ L_i^{\mathrm{T}}\ \ K_i^{\mathrm{T}}]^{\mathrm{T}}[A\ \ B\ \ E].$$

Since P_i is described as in (39) then it follows that

$$H_{i+1}=\begin{bmatrix} A^{\mathrm{T}}P_iA+R & A^{\mathrm{T}}P_iB & A^{\mathrm{T}}P_iE \\ B^{\mathrm{T}}P_iA & B^{\mathrm{T}}P_iB+I & B^{\mathrm{T}}P_iE \\ E^{\mathrm{T}}P_iA & E^{\mathrm{T}}P_iB & E^{\mathrm{T}}P_iE-\gamma^2 I \end{bmatrix}.$$

Using Eqs. (20), (36), one obtains (37), (38). □

Lemma 3. *Iterating on H_i is similar to iterating on P_i as*

$$P_{i+1}=A^{\mathrm{T}}P_iA+R-[A^{\mathrm{T}}P_iB\ \ A^{\mathrm{T}}P_iE] \times \begin{bmatrix} I+B^{\mathrm{T}}P_iB & B^{\mathrm{T}}P_iE \\ E^{\mathrm{T}}P_iB & E^{\mathrm{T}}P_iE-\gamma^2 I \end{bmatrix}^{-1} \begin{bmatrix} B^{\mathrm{T}}P_iA \\ E^{\mathrm{T}}P_iA \end{bmatrix} \tag{40}$$

with P_i defined as in (39).

Proof. From (39) in Lemma 2, one has

$$P_{i+1}=[I\ \ L_{i+1}^{\mathrm{T}}\ \ K_{i+1}^{\mathrm{T}}]H_{i+1}[I\ \ L_{i+1}^{\mathrm{T}}\ \ K_{i+1}^{\mathrm{T}}]^{\mathrm{T}}$$

and using (36) in Lemma 2, one obtains

$$P_{i+1}=\begin{bmatrix} I \\ L_{i+1} \\ K_{i+1} \end{bmatrix}^{\mathrm{T}} \times \begin{bmatrix} A^{\mathrm{T}}P_iA+R & A^{\mathrm{T}}P_iB & A^{\mathrm{T}}P_iE \\ B^{\mathrm{T}}P_iA & B^{\mathrm{T}}P_iB+I & B^{\mathrm{T}}P_iE \\ E^{\mathrm{T}}P_iA & E^{\mathrm{T}}P_iB & E^{\mathrm{T}}P_iE-\gamma^2 I \end{bmatrix} \times \begin{bmatrix} I \\ L_{i+1} \\ K_{i+1} \end{bmatrix}$$
$$=R+L_{i+1}^{\mathrm{T}}L_{i+1}-\gamma^2K_{i+1}^{\mathrm{T}}K_{i+1} + (A^{\mathrm{T}}+L_{i+1}^{\mathrm{T}}B^{\mathrm{T}}+K_{i+1}^{\mathrm{T}}E^{\mathrm{T}}) \times P_i(A+BL_{i+1}+EK_{i+1}). \tag{41}$$

Substituting (37), and (38) in (41), one has (40). □

The next result is our main theorem and shows convergence of the Q-learning algorithm.

Theorem 1. *Assume that the linear quadratic zero-sum game is solvable and has a value under the state feedback information structure. Then, iterating on Eq.* (35) *in Lemma 1, with $H_0=0$, $L_0=0$ and $K_0=0$ converges with $H_i \to H$, where H corresponds to $Q^*(x_k,u_k,w_k)$ as in* (10) *and* (12) *with corresponding P solving the GARE* (5).

Proof. In Stoorvogle and Weeren (1994) it is shown that iterating on the GARE (40) with $P_0=0$ converges to P that solves (5). Since Lemma 3 shows that iterating on H_i matrix is equivalent to iterating on P_i, then as $i\to\infty$

$$H_i \to \begin{bmatrix} A^{\mathrm{T}}PA+R & A^{\mathrm{T}}PB & A^{\mathrm{T}}PE \\ B^{\mathrm{T}}PA & B^{\mathrm{T}}PB+I & B^{\mathrm{T}}PE \\ E^{\mathrm{T}}PA & E^{\mathrm{T}}PB & E^{\mathrm{T}}PE-\gamma^2 I \end{bmatrix}$$

hence from (12), and since from (39) $H_0=0$, $L_0=0$ and $K_0=0$ implies that $P_0=0$, one concludes that $Q_i\to Q^*$. □

We have just proved convergence of the Q-learning algorithm assuming the least-squares problem (34) is solved completely; i.e. the excitation condition is satisfied. Note that this implies that Q-learning, can be interpreted as solving the GARE of the zero-sum game without requiring the plant model.

4. Online adp H_∞ autopilot controller design for an F-16 aircraft

H_∞ controllers have been proven to be highly effective in the design of feedback control systems with robustness and disturbance rejection capabilities for F-16 aircraft autopilot design. The presented H_∞ controller design is a model-free online tuning design that is based on the Q-learning method presented in this paper.

The F-16 short period dynamics has three states given as $x=[\alpha\ \ q\ \ \delta_e]^{\mathrm{T}}$, where α is the angle of attack, q is the pitch rate and δ_e is the elevator deflection angle. The discrete-time plant model of this aircraft dynamics is a discretized version of the continuous-time one given in Stevens and Lewis (2003). We used standard zero-order-hold discretization techniques

$$A=\begin{bmatrix} 0.906488 & 0.0816012 & -0.0005 \\ 0.0741349 & 0.90121 & -0.000708383 \\ 0 & 0 & 0.132655 \end{bmatrix},$$
$$B=\begin{bmatrix} -0.00150808 \\ -0.0096 \\ 0.867345 \end{bmatrix},\quad E=\begin{bmatrix} 0.00951892 \\ 0.00038373 \\ 0 \end{bmatrix} \tag{42}$$

with sampling time $T=0.1$. The disturbance attenuation is selected to be $\gamma=1$.

4.1. H_∞ solution based on the GARE

The solution of the GARE (5) given (42) is

$$P=\begin{bmatrix} 15.5109 & 12.4074 & -0.0089 \\ 12.4074 & 15.5994 & -0.0078 \\ -0.0089 & -0.0078 & 1.0101 \end{bmatrix}. \tag{43}$$

The corresponding policies have the following gains $L=[0.0733\ \ 0.0872\ \ -0.0661]$ and $K=[0.1476\ \ 0.1244\ \ 0]$. Note that $P\geqslant 0$ and hence from Başar and Bernhard (1995) this implies that for all finite energy disturbances, $u^*(x_k)$ has the well-known robustness and disturbance rejection capabilities of H_∞ control.

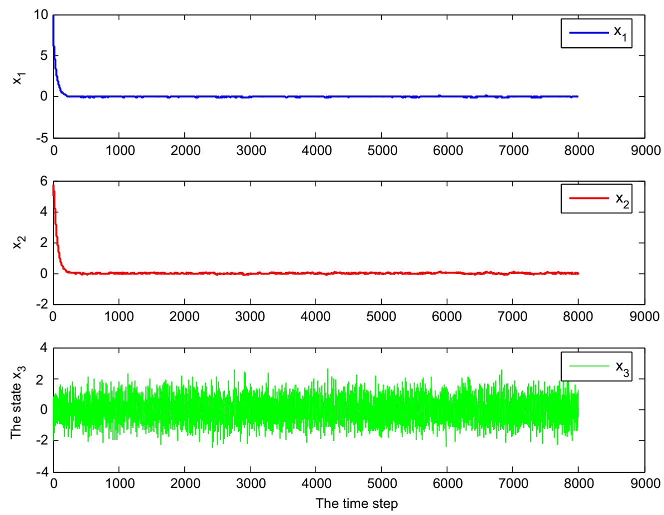


Fig. 2. States trajectories.

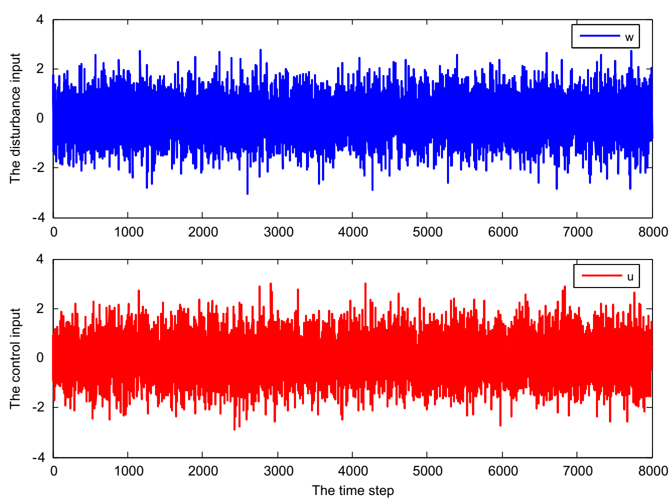


Fig. 3. The control and disturbance .

4.2. Q-learning based H_∞ autopilot controller design

In this part, the Q-learning algorithm developed in Section 3 of this paper is applied to solve for the H_∞ autopilot controller forward in time. The recursive least-squares algorithm is used to tune the parameters of the critic network on-line. The parameters of the actions networks are updated according to (20).

The states of the aircraft are initialized to be $x_0 = [10 \;\; 5 \;\; -2]$. Any values could be selected. The parameters of the critic network and the actions networks are initialized to zero. Following this initialization step, the aircraft dynamics are run forward in time and tuning of the parameter structures is performed by observing the states on-line.

In Figs. 2 and 3, the states and the inputs to the aircraft are shown with respect to time. In this example, we inject probing noise to the control and disturbance inputs. Hence, the persistency of excitation condition required for the convergence of the recursive least-squares tuning, i.e. avoiding the parameter drift problem, will hold. In Figs. 4–6, the convergence of the

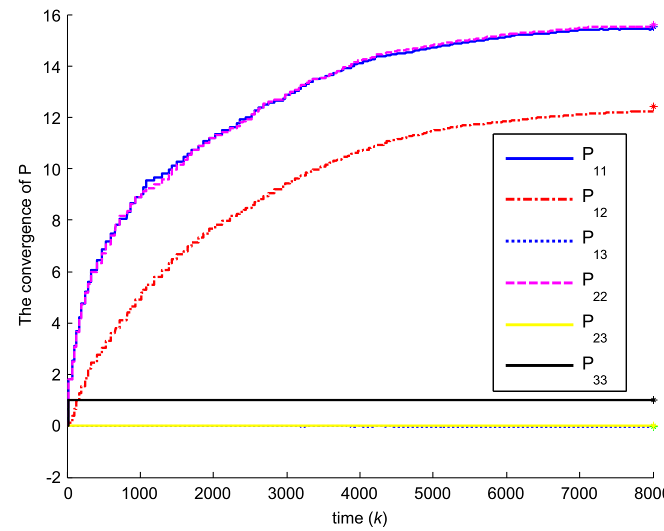


Fig. 4. Online model-free convergence of P_i to P.

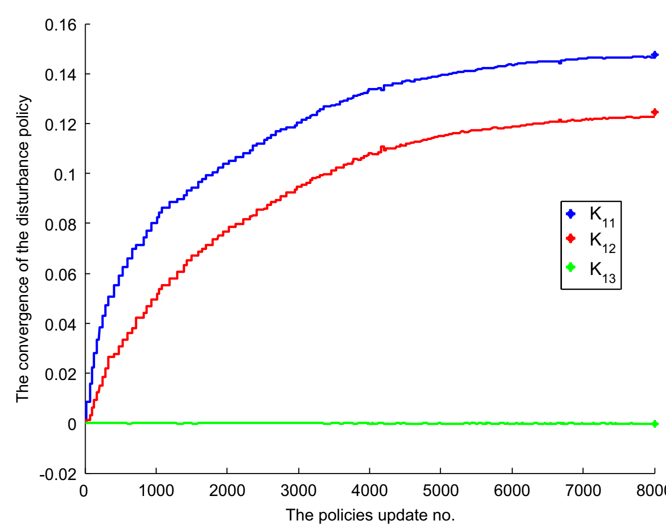


Fig. 5. Convergence of the disturbance action network parameters.

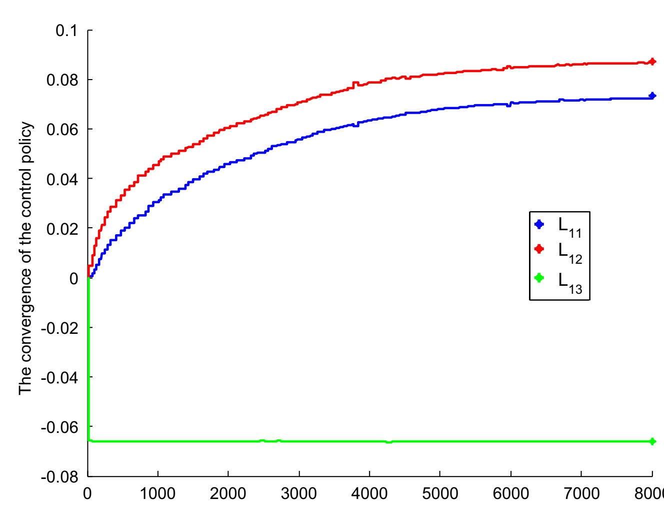


Fig. 6. Convergence of the control action network parameters.

critic and action networks is shown. Using (39), it is shown that the critic network parameters H_i converge to the corresponding game value P that solves (5).

5. Conclusion

In this paper we introduced an on-line ADP technique based on Q-learning to solve the discrete-time zero-sum game problem with continuous state and action spaces. The derivation of the policies and the convergence of the Q-learning are provided. In the Q-learning algorithm the system model is not needed to tune the action networks nor the critic network. The results in this paper can be summarized as a model-free approach to solve the linear quadratic discrete-time zero-sum game forward in time.

It is interesting to see that when designing the H_∞ controller in forward time, one needs to provide an input signal that acts as a disturbance that is tuned to be the worst case disturbance in forward time. Once the H_∞ controller is found, one can use the parameters of the control action network as the final parameters of the controller, without having to deliberately inserting any disturbance signal to the system.

Note that if $\gamma \to \infty$ or the disturbance gain matrix $E = 0$, a special case of this approach can be the solution of the discrete-time linear quadratic regulator (LQR) in optimal control

References

Abu-Khalaf, M., Lewis, F.L., & Huang, J. (2004). Hamilton-Jacobi-Isaacs formulation for constrained input nonlinear systems. *43rd IEEE conference on decision and control*. pp. 5034–5040.

Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike elements that can solve difficult learning control problems. *IEEE Transactions on Systems Man and Cybernetics*, *SMC-13*, 835–846.

Başar, T., & Bernhard, P. (1995). *H_∞ Optimal control and related minimax design problems*. Basel: Birkhäuser.

Başar, T., & Olsder, G.J. (1999). *Dynamic noncooperative game theory*. SIAN.

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

Bradtke, S.J., Ydestie, B.E., & Barto, A.G. (1994). Adaptive linear quadratic control using policy iteration. *Proceedings of the American control conference* (pp. 3475–3476).

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

Cao, Xi.-R. (2002). Learning and optimization—from a systems theoretic perspective. *Proceedings of IEEE conference on decision and control* (pp. 3367–3371).

Hagen, S., & Krose, B. (1998). Linear quadratic regulation using reinforcement learning. *Belgian_Dutch conference on mechanical learning* (pp. 39–46).

He, P., & Jagannathan, S. (2005). Reinforcement learning-based output feedback control of nonlinear systems with input constraints. *IEEE Transactions on Systems Man and Cybernetics—Part B*, *35*(1), 150–154.

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

Jacobson, D. H. (1977). On values and strategies for infinite-time linear quadratic games. *IEEE TAC*, *22*(3), 490–491.

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

Lewis, F. L. (1995). *V.L. Syrmos optimal control*. New York: Wiley.

Lin, W., & Byrnes, C. I. (1996). H_∞ control of discrete-time nonlinear system. *IEEE Transactions on Automatic Control*, *41*(4), 494–510.

Mageirou, E. F. (1976). Value and strategies for infinite time linear quadratic games. *IEEE TAC*, *21*(4), 547–550.

Prokhorov, D., & Wunsch, D. (1997). Adaptive critic designs. *IEEE Transactions on Neural Networks*, *8*(5), 997–1007.

Si, J., & Wang, (2001). On-Line learning by association and reinforcement. *IEEE Transactions on Neural Networks*, *12*, 264–276.

Si, J., Barto, A., Powell, W., & Wunsch, D. (2004). *Handbook of learning and approximate dynamic programming*. New Jersey: Wiley.

Stevens, B., & Lewis, F. L. (2003). *Aircraft control and simulation*. 2nd ed., New Jersey: Wiley.

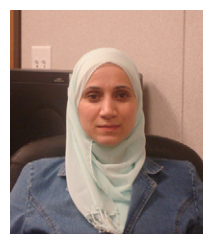
Stoorvogle, A. A., & Weeren, A. J. T. M. (1994). The discrete-time Riccati equation related to the H_∞ control problem. *IEEE Transactions on Automatic Control*, *39*(3), 686–691.

Watkins, C. (1989). *Learning from delayed rewards*, Ph.D. Thesis, Cambridge, England: Cambridge University.

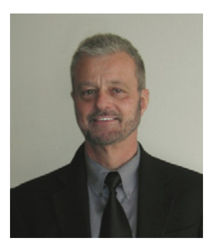
Werbos, P. J. (1990). Neural networks for control and system identification. *Heuristics*, *3*(1), 18–27.

Werbos, P.J. (1991). A menu of designs for reinforcement learning over time. *In Neural networks for control* (pp. 67–95). MA: MIT Press.

Werbos, P. J. (1992). Approximate dynamic programming for real-time control and neural modeling. In: D. A. White, & D. A. Sofge (Eds.), *Handbook of intelligent control* New York: Van Nostrand Reinhold.



Asma Al-Tamimi was born in Amman, Jordan in 1976. She did her high school studies at Ajnadeen high school in Zarqa. She received her Bachelor's Degree in Electromechanical Engineering from Al-Balqa University in Amman, Jordan in 1999. She then joined The University of Texas at Arlington from which she received the Master's of Science in Electrical Engineering in 2003. Currently she is working on her PhD degree at The University of Texas at Arlington and working as a research assistant at the Automation and Robotics Research Institute.



Frank L. Lewis was born in Würzburg, Germany, subsequently studying in Chile and Gordonstoun School in Scotland. He obtained the Bachelor's Degree in Physics/Electrical Engineering and the 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 the Master's of Science in Aeronautical Engineering from the University of West Florida. In 1981 he obtained the Ph.D. degree at The Georgia Institute of Technology in Atlanta, where he was employed as a professor from 1981 to 1990 and is currently an Adjunct Professor. 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 and Robotics Research Institute. He is a Fellow of the IEEE, a member of the New York Academy of Sciences, and a registered Professional Engineer in the State of Texas. He is a Charter Member (2004) of the UTA Academy of Distinguished Scholars. He has served as Visiting Professor at Democritus University in Greece, Hong Kong University of Science and Technology, Chinese University of Hong Kong, National University of Singapore. He is an elected Guest Consulting Professor at both Shanghai Jiao Tong University and South China University of Technology. Dr. Lewis' current interests include intelligent control, neural and fuzzy systems, microelectromechanical systems (MEMS), wireless sensor networks, nonlinear systems, robotics, condition-based maintenance, and manufacturing process control. He is the author/co-author of 3 U.S. patents, 157 journal papers, 23 chapters and encyclopedia articles, 239 refereed conference papers, nine 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*. He was selected to the Editorial Boards of *International Journal of Control*, *Neural Computing and*

Applications, and *Int. J. Intelligent Control Systems*. He served as an Editor for the flagship journal *Automatica*. He is the recipient of an NSF Research Initiation Grant and has been continuously funded by NSF since 1982. Since 1991 he has received $4.8 million in funding from NSF and other government agencies, including significant DoD SBIR and industry funding. His SBIR program was instrumental in ARRI's receipt of the SBA Tibbets Award in 1996. He has received a Fulbright Research Award, the American Society of Engineering Education *F.E. Terman* Award, three Sigma Xi Research Awards, the UTA Halliburton Engineering Research Award, the UTA University-Wide Distinguished Research Award, the ARRI Patent Award, various Best Paper Awards, the IEEE Control Systems Society Best Chapter Award (as Founding Chairman), and the National Sigma Xi Award for Outstanding Chapter (as President). He was selected as Engineer of the year in 1994 by the Ft. Worth IEEE Section. He was appointed to the NAE Committee on Space Station in 1995 and to the IEEE Control Systems Society Board of Governors in 1996. In 1998 he was selected as an IEEE Control Systems Society *Distinguished Lecturer*. He is a Founding Member of the Board of Governors of the Mediterranean Control Association.

Murad Abu-Khalaf was born in Jerusalem, Palestine in 1977. He obtained his B.S. in Electronics and Electrical Engineering from Boğaziçi University in Istanbul, Turkey in 1998, and the M.S. and Ph.D. in Electrical Engineering from The University of Texas at Arlington in 2000 and 2005, respectively. His research interest is in the areas of nonlinear control, optimal control, neural network control, and adaptive intelligent systems. He is the author/co-author of one book, two book chapters, 8 journals papers and 15 refereed conference proceedings. He is a member of IEEE member, and a member of Etta Kappa Nu honor society, and is listed in Who's Who in America. His interest is in the areas of nonlinear control, optimal control, neural network control, adaptive intelligent systems.