17

Integral Reinforcement Learning for Finding Online the Feedback Nash Equilibrium of Nonzero-Sum Differential Games

Draguna Vrabie and Frank L. Lewis
University of Texas at Arlington
United States

1. Introduction

Adaptive/Approximate Dynamic Programming (ADP) is the class of methods that provide 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 is characterizing the classical Dynamic Programming approach (Bellman, 2003). These methods were initially developed for systems with finite state and action spaces and are based on Sutton's temporal difference learning (Sutton, 1988), Werbos' Heuristic Dynamic Programming (HDP) (Werbos, 1992), and Watkins' Q-learning (Watkins, 1989).

The applicability of these online learning methods to real world problems is enabled by approximation tools and theory. The value that is associated with a given admissible control policy will be determined using value function approximation, online learning techniques, and data measured from the system. A control policy is determined based on the information on the control performance encapsulated in the value function approximator. Given the universal approximation property of neural networks (Hornik et al., 1990), they are generally used in the reinforcement learning literature for representation of value functions (Werbos, 1992), (Bertsekas and Tsitsiklis, 1996), (Prokhorov and Wunsch, 1997), (Hanselmann et al., 2007). Another type of approximation structure is a linear combination of a basis set of functions and it has been used in (Beard et al., 1997), (Abu-Khalaf et al., 2006), (Vrabie et al. 2009).

The approximation structure used for performance estimation, endowed with learning capabilities, is often referred to as a critic. Critic structures provide performance information to the control structure that computes the input of the system. The performance information from the critic is used in learning procedures to determine improved action policies. The methods that make use of critic structures to determine online optimal behaviour strategies are also referred to as adaptive critics (Prokhorov and Wunsch, 1997), (Al-Tamimi et al., 2007), (Kulkarni & Venayagamoorthy, 2010).

Most of the previous research on continuous-time reinforcement learning algorithms that provide an online approach to the solution of optimal control problems, assumed that the dynamical system is affected only by a single control strategy. In a game theory setup, the controlled system is affected by a number of control inputs, computed by different controllers

that try to optimize individual performance functions. In these situations the control problem is formulated with the purpose of finding the set of control policies that are admissible, i.e. control policies that guarantee the stability of the controlled dynamical system, and minimize the individual performance functions in a Nash equilibrium sense. This kind of solution is characterized by the fact that any change in the policy of any given player (in the space of admissible policies) will result in a worse performance for that player, relative to the performance that it receives by means of the Nash equilibrium solution policy.

Nash differential games have been originally introduced in (Starr & Ho, 1969). Their study is highly relevant as they have a number of potential applications in control engineering and economics (see e.g. (Abou-Kandil et al., 2003); (Engwerda, 2005)). The underlying game theory formulation appears also in the study of coupled large scale systems (Mukaidani, 2007-a), e.g. networking and wireless communication systems (Shah, 1998).

This chapter is presenting an Adaptive Dynamic Programming (ADP) algorithm, formulated using the continuous-time mathematical framework, that provides, in an online manner, the Nash equilibrium solution of two-player nonzero-sum differential games with linear dynamics and infinite horizon quadratic cost. The main advantage of this ADP approach consists in the fact that neither of the two participants in the game makes use of explicit knowledge on the model of the drift dynamics of the system that they influence through their behavior policy. This means that the two players will learn online the most effective behavior policies that correspond to the Nash equilibrium while using no explicit knowledge on the drift dynamics of the differential game. This results in two clear benefits when compared with model based procedures:

- conducting identification experiments for finding the drift term that describes the system dynamics is not required, while this lack of knowledge does not have any impact on the obtained equilibrium solution,
- the resulting equilibrium behavior policies of the two players will not be affected by any error differences between the dynamics of a model of the system and the dynamics of the real system.

For the case when the system has linear dynamics and the cost indices are quadratic and have infinite horizon, it is known that finding the Nash equilibrium to the game problem is equivalent with calculating the solution of a set of coupled algebraic Riccati equations (ARE) (see e.g. (Starr and Ho, 1969), (Abou-Kandil et al., 2003), (Basar and Olsder, 1999), (Engwerda, 2005)). The solution of the coupled ARE has been approached in (Cherfi et al., 2005-a), (Cherfi et al., 2005-b), (Jungers et al., 2007), (Freiling, 1996), (Li and Gajic, 1995) by means of iterative procedures. These algorithms construct sequences of cost functions, or matrices, which converge to the equilibrium solution of the game. In the case of (Cherfi et al., 2005-a), (Cherfi et al., 2005-b), (Freiling et al., 1996), and (Jungers et al., 2007), convergence results of these procedures are still to be determined. It is important to note that all above mentioned algorithms require exact and complete knowledge of the system dynamics and the solution is obtained by means of offline iterative computation procedures.

An ADP procedure that provides solution to the Hamilton-Jacobi-Isaacs equation, associated with the two-player zero-sum nonlinear differential game, has been introduced in (Wei and Zhang, 2008). The ADP algorithm involves calculation of two sequences of cost functions, the upper and lower performance indices, sequences that converge to the saddle point solution of the game. The adaptive critic structure that is required for learning the saddle point solution is comprised by four action networks and two critic networks. The requirement of full knowledge on the system dynamics is still present in the case of that algorithm.

The result presented in this chapter is the first reinforcement learning approach to the saddle point solution of a two player nonzero-sum differential game. By virtue of the online ADP method, that makes use of the integral reinforcement learning (IRL) approach (Vrabie et al., 2009), exact knowledge of part of the system dynamics is not required. To our knowledge, there exists no ADP algorithm that provides the Nash equilibrium solution of the two-player nonzero-sum differential game in an online fashion and without using complete information on the model of the dynamical system to be controlled.
The main traits of this new online procedure are the following:

- It involves the use of ADP techniques that will determine the Nash equilibrium solution of the game in an online data-based procedure that does not require full knowledge of the system dynamics.
- It is the online version of a mathematical algorithm that solves the underlying set of coupled algebraic Riccati equations of the game problem. The equivalent algorithm makes use of offline procedures and requires full knowledge of the system dynamics to determine the Nash equilibrium of the game.

In this ADP approach both game players are actively learning and improving their policy. The algorithm is built on interplay between

- a learning phase, and
- a policy update step.

During the learning phase each of the players is learning the value function that it associates with the use of a given pair of admissible policies. Both players are learning simultaneously. During the policy update step both players are changing their feedback control policies in the sense of performance improvement. That means that each player will change its policy such that it will minimize his cost in front of the previous policy of their opponent.
For learning the value that each player associates with a given admissible pair of control policies we will use value function approximation. In this chapter we will consider the case in which the critic is represented as a linear combination of a set of basis functions which spans the space of value functions to be approximated, see e.g. (Beard et al., 1997). The learning technique that is here employed for value function approximation uses the concept of minimization of the temporal difference error and has been described in (Vrabie, 2009).
The objective of this chapter is to present an online algorithm that makes use of ADP techniques to provide the solution to the two-player differential nonzero-sum game. It will also show that the foundation of the novel online procedure that will be described here is the mathematical result introduced in (Li and Gajic, 1995). That algorithm involves solving a sequence of Lyapunov equations in order to build a sequence of control policies that converges to the Nash equilibrium solution of the game, and thus requires full knowledge on the system dynamics. Herein we will show how, by means of ADP techniques, the solution of these game optimal control problems can be obtained in an online fashion, using measured data from the system, and reduced information on the system dynamics.
We begin our investigation by providing the formulation of the two player nonzero-sum game problem. We then provide an overview of the online integral reinforcement learning (IRL) method that can be used online to determine the value associated with a given pair of admissible control strategies. In Section 3 we describe the online method that provides the Nash equilibrium solution of the two-player nonzero-sum game. The adaptive critic structure associated with the online solution of the game will also be discussed. It will be important to note that in this case, each of the two players will make use of a critic structure

that will use reinforcement learning ideas to learn online the value that the player associates with a given admissible control strategy. Section 4 will investigate the convergence properties of the online reinforcement learning algorithm. It will be shown that the ADP procedure introduced in this chapter is theoretically equivalent with the iterative procedure introduced in (Li & Gajic, 1995), and thus has the same convergence properties. A formulation of the algorithm in the form of a quasi-Newton method will also be provided. Section 5 will present a simulation result.

2. Preliminaries

2.1 Problem formulation

We consider the system described by the equation:

$$\begin{aligned} \dot{x} &= Ax + B_1 u_1 + B_2 u_2 \\ x(t_0) &= x_0 \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n, u_i \in \mathbb{R}^{m_i}$ for $i = 1,2$, and A, B_1 and B_2 are matrices of appropriate dimensions. Each player i, $i = 1,2$, desires to determine the feedback control strategy $u_i = K_i x$ such that the quadratic performance index, where $Q_i \geq 0, R_{ij} \geq 0 (i \neq j), R_{ii} > 0$,

$$J_i = \frac{1}{2} \int_{t_0}^{\infty} (x^T Q_i x + {u_1}^T R_{i1} u_1 + {u_2}^T R_{i2} u_2) d\tau \tag{2}$$

is minimized.

Definition 1

A feedback control pair (u_1, u_2) is admissible if the dynamics of the closed loop system (1) are stable and the performance indices (2) calculated for the given control pair have finite values.

The two-player game problem is defined as follows:

Given the continuous-time system (1), the cost functions $J_i, i = 1,2$ defined by (2), and the set of admissible control inputs $U \subset \mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$, determine the state-feedback admissible control policies such that the closed loop system is stable and the cost functions attain the minimum possible value.

These control strategies corresponds to the Nash equilibrium of the two-player differential game. Thus, the pair of feedback control policies that is sought, denoted (u_1^*, u_2^*), satisfies the following relations for any admissible control pair $(u_1, u_2) \in U$

$$\begin{aligned} J_1(u_1, u_2^*) &\geq J_1(u_1^*, u_2^*) \\ J_2(u_1^*, u_2) &\geq J_2(u_1^*, u_2^*) \end{aligned}. \tag{3}$$

For $i = 1,2$ and $j = 1,2$ $j \neq i$, let us define the minimum cost function by:

$$V_i(x) = \min_{u_i \in U_i} J_i(u_i, {u_j}^*, x) \quad \forall x \in \mathbb{R}^n . \tag{4}$$

Assuming that the optimal value function is differentialble, we can then write two coupled equations, for $i, j = 1,2, j \neq i$,

$$0 = \min_{(u_i, u_j^*) \in U} \left\{ x^T Q_i x + u_i^T R_{i1} u_i + u_j^{*T} R_{i2} u_j^* + \nabla^T V_i [Ax + B_j u_j^* + B_i u_i] \right\} \quad \forall x \in \mathbb{R}^n , \tag{5}$$

that we shall refer to as the Hamilton-Jacobi-Bellman equations.
After performing the minimization in (5) we obtain that the two elements of the closed loop optimal control pair (u_1^*, u_2^*) will have the state feedback form

$$u_i^* = -R_{ii}^{-1} B_i^T P_i^* x = K_i^* x \quad i = 1,2 \tag{6}$$

where the values of the two matrices P_i^*, $i = 1,2$ satisfy the necessary conditions for finding the Nash equilibrium, i.e. the two matrices $P_i^*, i = 1,2$ must be positive definite solutions to the coupled algebraic Riccati equations (ARE)

$$\begin{aligned} N_1(P_1^*, P_2^*) &\triangleq A^T P_1^* + P_1^* A + Q_1 + P_2^* S_{12} P_2^* - P_1^* S_1 P_1^* - P_2^* S_2 P_1^* - P_1^* S_2 P_2^* = 0 \\ N_2(P_1^*, P_2^*) &\triangleq A^T P_2^* + P_2^* A + Q_2 + P_1^* S_{21} P_1^* - P_2^* S_2 P_2^* - P_1^* S_1 P_2^* - P_2^* S_1 P_1^* = 0 \end{aligned} \tag{7}$$

where $S_i = B_i R_{ii}^{-1} B_i^T$, $i = 1,2$ and $S_{ij} = B_j R_{jj}^{-1} R_{ij} R_{jj}^{-1} B_j^T$, $i, j = 1,2, j \neq i$.
Finding Nash equilibrium solutions of the game (u_1^*, u_2^*), defined through (6) by the pair of matrices (P_1^*, P_2^*), resumes to finding solutions to the coupled AREs (7) such that the closed loop system dynamics will be stable, i.e. $A - S_1 P_1^* - S_2 P_2^*$ is Hurwitz.

2.2 Integral reinforcement learning

The online iterative procedure that will be presented in Section 3 relies heavily on value function estimation. Thus the goal of this section is to briefly present the online procedure, introduced in (Vrabie et al., 2009), that uses reinforcement learning ideas to find the value of the parameters of the infinite horizon cost associated with a quadratic cost function such as $J_i, i = 1,2$. We refer to this online method as integral reinforcement learning (IRL).
As stated above, the procedure presented herein is used to find the value of the parameters of the infinite horizon cost associated with a cost function that has a quadratic nature, such as $J_i, i = 1,2$. To bring the general theoretical concept into specific, let us formulate the following problem: Given the dynamical system (1) and an admissible pair of linear state-feedback control policies $(u_1, u_2) = (K_1 x, K_2 x) \subset U$, determine the parameters of the infinite horizon cost function J_i, that player *i* associates with this admissible control pair.
Before giving an online procedure for solving this problem one needs to choose a parametric representation for the value function to be determined. In this particular case the cost functions are quadratic and the control policies have linear state-feedback structure. Thus a quadratic representation in the initial state can provide an exact representation for each of the two cost functions. One can write:

$$J_i = x_0^T P_i x_0 = \frac{1}{2} \int_{t_0}^{\infty} x^T \overline{Q_i} x d\tau \tag{8}$$

where $\overline{Q_i} = Q_i + K_1^T R_{i1} K_1 + K_2^T R_{i2} K_2$, $i = 1,2$.
After choosing a parametric representation for the value function one has to determine the values of its parameters, namely the matrix P_i. The integral reinforcement learning algorithm that will be used for finding the parameters of the value function, i.e. the value of the matrix P_i, is based on the following equation that is satisfied for every time sample $T_0 > 0$

$$x_t^T P_i x_t = \int_t^{t+T_0} x_\tau^T \overline{Q_i} x_\tau d\tau + x_{t+T_0}^T P_i x_{t+T_0} \tag{9}$$

where x_τ denotes the state of the system described by $\dot{x} = (A + B_1 K_1 + B_2 K_2)x$ with initial condition x_t, and x_{t+T_0} is the value of the state at time $t + T_0$.
The online implementation of the algorithm is given next.
The solution of (9) consists of the value of the matrix P_i that is parameterizing the cost function. The quadratic cost functions will be written as:

$$x_t^T P_i x_t = \overline{p}_i^T \overline{x}_t \tag{10}$$

where $\overline{x_t}$ denotes the Kronecker product quadratic polynomial basis vector with the elements $\{x_k(t)x_l(t)\}_{k=1,n;l=k,n}$ and $\overline{p} = v(P)$ with $v(.)$ a vector valued matrix function that acts on symmetric matrices and returns a column vector by stacking the elements of the diagonal and upper triangular part of the symmetric matrix into a vector, where the off-diagonal elements are taken as $2P_{ij}$, (Brewer, 1978). Denote the integral reinforcement over the time interval $[t, t+T_0]$ by:

$$d(\overline{x}_t, K_1, K_2) \equiv \int_t^{t+T_0} x_\tau^T \overline{Q_i} x_\tau d\tau \ . \tag{11}$$

Based on these notations and structures, (9) is rewritten as:

$$\overline{p}_i^T (\overline{x}_t - \overline{x}_{t+T_0}) = d(\overline{x}_t, K_1, K_2) \ . \tag{12}$$

In (12) the vector of unknown parameters is $\overline{p}_i$ and $\overline{x}_t - \overline{x}_{t+T_0}$ acts as a regression vector. The right hand side target integral reinforcement function is measured based on the state trajectories over the time interval $[t, t+T_0]$.
The parameter vector $\overline{p}_i$ is found by minimizing, in the least-squares sense, the error between the target expected cost over the finite horizon, and the measured cost, $d(\overline{x}_t, K_1, K_2)$. Thus the sought parameters satisfy

$$\overline{p}_i = \arg\min_\eta (d(\overline{x}_t, K_1, K_2) - \eta^T (\overline{x}_t - \overline{x}_{t+T_0}))^2 \ . \tag{13}$$

The solution can be obtained online based on data measured along the trajectories of the system, and using batch least squares or the recursive least squares algorithm.
It is important to note that this online algorithm for value function approximation is a data-based approach that uses reinforcement learning ideas. Also, this value function approximation technique does not require explicit knowledge of the model of the controlled system's drift dynamics, i.e. matrix *A*, or input to state matrices B_1, B_2.

3. Online iterative algorithm that solves the coupled algebraic Riccati equations of the nonzero-sum game

3.1 Initialization of the online algorithm

Before we proceed with the description of the online algorithm, we give a necessary assumption.

Assumption 1 The triples $(A, B_i, \sqrt{Q_i})$, $i = 1,2$ are stabilizable and detectable.

Under this assumption one can reasonably say that initial state feedback control strategies $u_i^{(0)} = K_i^{(0)} x \quad i = 1,2$ exist such that closed loop system matrix $A - B_1 K_1^{(0)} - B_2 K_2^{(0)}$ is Hurwitz.

A procedure for obtaininig the two controllers such that the closed loop system is stable is described next. The procedure has two steps and it can be execute in an online manner without using knowledge on the drift dynamics of the system (1), i.e. without knowing the matrix *A*.

Step 1

Let Player 2 use the "no control" policy corresponding to $u_2(x) = 0$, and determine the optimal control strategy of Player 1 with respect to the cost index J_1.

This is a classical linear quadratic regulation problem and the optimal control strategy will have the form $u_1^{(0)}(x) = K_1^{(0)} x = -R_{11}^{-1} B_1^T P_1^{(0)} x$ where $P_1^{(0)}$ is the solution of the ARE

$$A^T P_1^{(0)} + P_1^{(0)} A + Q_1 - P_1^{(0)} B_1 {R_{11}}^{-1} {B_1}^T P_1^{(0)} = 0 . \tag{14}$$

Note that the solution of this single player optimal control problem can be obtained by solving (14) by means of the online ADP technique introduced in (Vrabie et al., 2009), without using any knowledge on the drift dynamics described by matrix *A* .

For completeness we outline the procedure herein.

a. We start from the assumption that an initial stabilizing state-feedback control policy $u_1^{(00)}(x) = -K_1^{(00)} x$ is available such that the matrix describing the closed loop system $A - B_1 K_1^{(00)}$ is Hurwitz.

b. For $k \geq 0, k \in \mathrm{N}$, determine the value function defined as:

$$x_0^T P_1^{(0,k+1)} x_0 = \frac{1}{2} \int_{t_0}^{\infty} x^T(\tau) \left(Q_1 + K_1^{(0,k)} R_{11} K_1^{(0,k)} \right) x(\tau) d\tau , \tag{15}$$

function that is associated with the use of the stabilizing state-feedback controller $u_1^{(0,k)}(x) = -K_1^{(0,k)} x = -R_{11}^{-1} B_1^T P_1^{(0,k)} x$, where $x_0 = x(0)$ is an initial state.

The sequence of matrices $P_1^{(0,k)}$, $k \geq 0, k \in \mathrm{N}$ can be determined using integral reinforcement learning, as described in Section 2.2, using discrete-time data measured from the system and without using any knowledge on the dynamics of the system (1).

Finding this value via de online model free algorithm is equivalent with solving the Lyapunov equation

$$(A - B_1 K_1^{(0,k)})^T P_1^{(0,k+1)} + P_1^{(0,k+1)} (A - B_1 K_1^{(0,k)}) + Q_1 + K_1^{(0,k)} R_{11} K_1^{(0,k)} = 0 , \tag{16}$$

equation that requires complete knowledge on the model of the system.

c. The iterative procedure described in b) has as result a convergent sequence of positive definite matrices, as shown in (Kleinman, 1968), such that $P_1^{(0,k)} \underset{k \to \infty}{\to} P_1^{(0)}$. A stop criterion can be defined as:

$$\left\| P_1^{(0,k+1)} - P_1^{(0,k)} \right\| \leq \varepsilon \tag{17}$$

or:

$$\left\| A^T P_1^{(0,k)} + P_1^{(0,k)} A + Q_1 - P_1^{(0,k)} B_1 R_{11}^{-1} B_1^T P_1^{(0,k)} \right\| \le \varepsilon \,, \tag{18}$$

for a prespecified value of ε, where $\| \|$ denotes a matrix norm. The latter expression, although it requires knowledge of the system dynamics, can be checked using online measured data and equation (12) such as

$$\left(\bar{p}_1^{(0,k)} \right)^T (\overline{x_t} - \overline{x_{t+T_0}}) - d(\overline{x_t}, K_1^{(0,k)}, 0) \le \varepsilon \tag{19}$$

The result is that the dynamics of the system (1) with the control pair $(u_1^{(0)}(x), 0)$ are stable, i.e. $A - S_1 P_1^{(0)}$ is Hurwitz.

Step 2

Let Player 1 use the stabilizing control policy $u_1^{(0)}(x) = K_1^{(0)} x$, and determine the optimal control strategy of Player 2 with respect to the cost index J_2.

Again, this is a classical linear quadratic regulation problem and the optimal control strategy will have the form $u_2^{(0)}(x) = K_2^{(0)} x = -R_{22}^{-1} B_2^T P_2^{(0)} x$ where $P_2^{(0)}$ is the solution of the ARE

$$(A - S_1 P_1^{(0)})^T P + P(A - S_1 P_1^{(0)}) + Q_2 + P_1^{(0)} S_{21} P_1^{(0)} - P B_2 R_{22}^{-1} B_2^T P = 0 \,. \tag{20}$$

Similarly to Step 1, the solution of this single player optimal control problem can be obtained by means of the online ADP IRL technique, introduced in (Vrabie et al., 2009) and outlined above, without using any knowledge on the drift dynamics of the system described by the matrix A.

The resulting control pair $(u_1^{(0)}(x), u_2^{(0)}(x))$ is admissible, i.e. $A - S_1 P_1^{(0)} - S_2 P_2^{(0)}$ is Hurwitz. At this point we are in the possession of an initial admissible pair of feedback control strategies $(u_1^{(0)}, u_2^{(0)}) = (K_1^{(0)} x, K_2^{(0)} x)$, that we shall also represent by $(P_1^{(0)}, P_2^{(0)})$.

It is worth noting that the Step 1 above can also be executed with respect to Player 2, followed by Step 2 that will now be relative to Player 1. Also in this case, a pair of admissible control policies will be obtained.

In the following we formulate the iterative algorithm that learns online the Nash equilibrium solution of the two-player zero-sum differential game. At every step of the iterative procedure each player uses reinforcement learning to estimate the infinite horizon value function that it associates with the current admissible control pair. Following the value function estimation procedure each of the two players makes a decision to improve its control policy. The end result is an online algorithm which leads to the saddle point solution of the differential game while neither of the two players uses any knowledge on the drift dynamics of the environment.

3.2 Online partially model free algorithm for solving the nonzero-sum differential game

Initialization

Start with initial matrices $(P_1^{(0)}, P_2^{(0)})$ such that $A - S_1 P_1^{(0)} - S_2 P_2^{(0)}$ is Hurwitz (i.e. initial control policies for both players are available such that the closed loop dynamics of the system are stable). Let $k = 0$.

Itterative procedure

For $k \geq 0, k \in \mathrm{N}$, let two critic structures use the integral reinforcement learning procedure described in Section 2.2 to determine the value that each of the two players is associating with the control policies described by the matrix pair $(P_1^{(k)}, P_2^{(k)})$. Namely each of the two critics will determine the matrices $P_i^{(k+1)}, i = 1,2, k \geq 0$ that satisfy

$$x_0^T P_i^{(k+1)} x_0 = \frac{1}{2}\int_{t_0}^{\infty} x^T \overline{Q_i}^{(k)} x d\tau \tag{21}$$

where $\overline{Q_i}^{(k)} = Q_i + P_i^{(k)} S_i P_i^{(k)} + P_j^{(k)} S_{ij} P_j^{(k)} \quad i = 1,2, j \neq i$.

Each of the two players will update their control policies such that the new control policy pair is characterized by $(P_1^{(k+1)}, P_2^{(k+1)})$, i.e.

$$\begin{aligned} u_1^{(k+1)}(x) &= K_2^{(k+1)} x = -R_{11}^{-1} B_1^T P_1^{(k+1)} x \\ u_2^{(k+1)}(x) &= K_2^{(k+1)} x = -R_{22}^{-1} B_2^T P_2^{(k+1)} x \end{aligned} . \tag{22}$$

Stop criterion

Stop the online algorithm when the following criterion is satisfied for a specified value of the number ε

$$\max\left(\left\|N_1(P_1^{(k+1)}, P_2^{(k+1)})\right\|, \left\|N_2(P_1^{(k+1)}, P_2^{(k+1)})\right\|\right) \leq \varepsilon \ , \tag{23}$$

where $\| \ \|$ denotes a matrix norm. The latter expression can be checked using online measured data and the following relation

$$\left(\left(\overline{p_1}^{(k+1)}\right)^T (\overline{x_t} - \overline{x_{t+T_0}}) - d(\overline{x_t}, K_1^{(k+1)}, K_2^{(k+1)}), \left(\overline{p_2}^{(k+1)}\right)^T (\overline{x_t} - \overline{x_{t+T_0}}) - d(\overline{x_t}, K_1^{(k+1)}, K_2^{(k+1)})\right) \leq \varepsilon \ . \tag{24}$$

3.3 Adaptive critic structure for solving the two-player Nash differential game

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

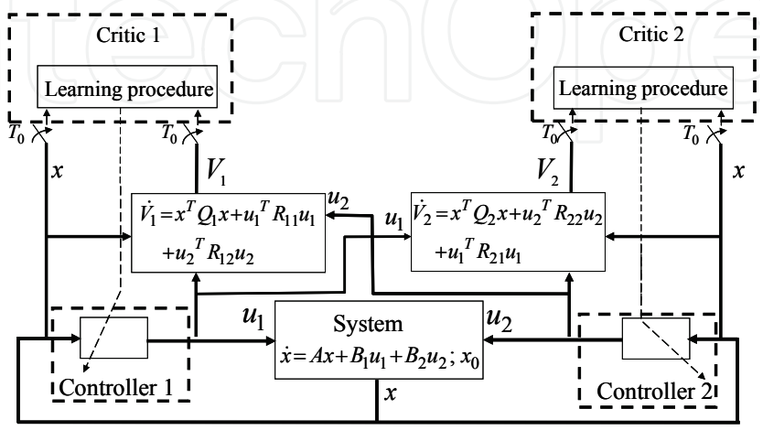


Fig. 1. Adaptive critic structure for the ADP game with IRL.

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

a. the continuous-time scale, represented by the full lines, that is connected with the continuous-time dynamics of the system and the continuous-time computation performed by the two players;
b. a discrete time scale given by T_0. This time scale is connected with the online learning procedure that is based on discrete-time measured data;
c. a slower, discrete-time scale that is a multiple of T_0. This time scale, indicated by the dashed lines, is connected with the update procedure of the control policies of the two players. The update procedure is performed only after the value function learning procedure, that uses integral reinforcement information, has converged.

The values of the time periods T_0 can be variable and are controlled by the two learning critics. Each critic will output a matrix $P_i^{(k)}, i=1,2$, in a synchronous fashion, after both online learning algorithms for the value functions have converged. Each controller will use the information from its corresponding critic to calculate and then implement a new control policy.

From the perspective of two-player games, the proposed online algorithm can be presented as follows:

Initialization

Let the initial policy of Player 2 be zero $u_2^{(00)}=0$.

Let Player 1 determine its optimal control policy $K_1^{(0)}=-R_{11}^{-1}B_1^T P_1^{(0)}$ in an online optimization procedure while Player 2 is not playing the game.

Let Player 2 determine its optimal control policy $K_2^{(0)}=-R_{22}^{-1}B_2^T P_2^{(0)}$ in an online optimization procedure while Player 1 is playing the game using $K_1^{(0)}$.

Iterative procedure

For $k\geq 0$, let both players determine online, using the integral reinforcement learning procedure, the values that they associate with the use of the policy pair $(K_1^{(k)}, K_2^{(k)})$, namely the pair of matrices $(P_1^{(k+1)}, P_2^{(k+1)})$.

Let both players update their control policies using

$$K_i^{(k+1)}=-R_{ii}^{-1}B_i^T P_i^{(k+1)}. \tag{25}$$

Stop criterion

Let both players stop this iterative procedure when there is no change in the control policies is observed at two successive steps (i.e. the Nash equilibrium has been obtained and both players can not further improve their cost function by changing their behavior policy).

4. Analysis of the online learning algorithm

In this section we are providing an analysis for the online algorithm that was introduced in section 3.

4.1 Mathematical formulation of the online algorithm

Using the notation $A^{(k)}=A-S_1P_1^{(k)}-S_2P_2^{(k)}$, it can be shown that equations (21) can be written as:

$$\left(A^{(k)}\right)^T P_i^{(k+1)}+P_i^{(k+1)}A^{(k)}=-\overline{Q}_i^{(k)} \tag{26}$$

where $i = 1,2$.
Thus the online algorithm described in Section 3.2 is equivalent with the following procedure:
Initialization
Start with initial matrices $(P_1^{(0)}, P_2^{(0)})$ such that $A - S_1 P_1^{(0)} - S_2 P_2^{(0)}$ is Hurwitz.
Iterative procedure
For $k \geq 0, k \in \mathrm{N}$, solve the Lyapunov equations (25).
Stop criterion
Stop the online algorithm when the criterion (23) is satisfied for a user specified value of ε.
This offline algorithm that uses iterations on Lyapunov equations has been proposed and analysed in (Li & Gajic, 1995) and its convergence has been further discussed in (Mukaidani, 2006) and (Mukaidani, 2007-b). Considering the mathematical equivalence between the algorithm introduced in (Li & Gajic, 1995) and the online procedure based on reinforcement learning that we proposed in Section 3, we can conclude that the online, partially model free, algorithm that we presented herein has the same convergence properties.

4.2 Analysis of the online algorithm

It is interesting to see that, similarly to the Newton method proposed in (Kleinman, 1968) for solving the classical continuous-time algebraic Riccati equation, the algorithm presented in this chapter relies on iterations on Lyapunov equations. However, the online procedure introduced here, and its underlying algorithm, is not a Newton method for finding the solution of the coupled ARE given in (7). This shall be clarified by means of the next two propositions.
First let us look at the formulation of the Newton method that determines the unique positive definite solution of the classical continuous-time algebraic Riccati equation

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 . \tag{27}$$

Denote with $Ric(P_k)$ the matrix valued function defined as

$$Ric(P_k) = A^T P_k + P_k A + Q - P_k BR^{-1}B^T P_k \tag{28}$$

and let Ric'_{P_k} denote the Frechet derivative of $Ric(P_k)$ taken with respect to P_k. The matrix function Ric'_{P_k}, evaluated at a given matrix M, will thus be

$$Ric'_{P_k}(M) = (A - BR^{-1}B^T P_k)^T M + M(A - BR^{-1}B^T P_k) . \tag{29}$$

Proposition 1 The unique positive solution of (27) can be determined by Newton's method given by:

$$P_k = P_{k-1} - (Ric'_{P_{k-1}})^{-1} Ric(P_{k-1}) , \tag{30}$$

provided that the initial matrix P_0 is such that $A - BR^{-1}B^T P_0$ is Hurwitz; and considering that the regular conditions for existence and uniqueness of positive definite solution are satisfied. For a proof see (Vrabie et al., 2009).
Next we will use the same mathematical tools to provide formulation to the algorithm used herein.

Consider the notations introduced in (7) for the two coupled algebraic Riccati equations, and let $N_{1_{P_1^{(k)}}}^{'}$ and $N_{2_{P_2^{(k)}}}^{'}$ denote the Frechet derivatives of $N_1(P_1^{(k)}, P_2^{(k)})$ and $N_2(P_1^{(k)}, P_2^{(k)})$, taken with respect to $P_1^{(k)}$ and respectively $P_2^{(k)}$, such that

$$\begin{aligned} N_{1_{P_1^{(k)}}}^{'}(M) &= (A^{(k)})^T M + MA^{(k)} \\ N_{2_{P_2^{(k)}}}^{'}(M) &= (A^{(k)})^T M + MA^{(k)} \end{aligned}, \tag{31}$$

where $A^{(k)} = A - S_2 P_2^{(k)} - S_1 P_1^{(k)}$.

Proposition 2 Consider that the regular conditions for existence and uniqueness of solution of the infinite horizon nonzero-sum differential game with quadratic performance are satisfied. Then, provided that an initial pair $(P_1^{(0)}, P_2^{(0)})$ is such that $A^{(0)} = A - S_2 P_2^{(0)} - S_1 P_1^{(0)}$ is Hurwitz, the online algorithm described in Section 3.2, that provides the Nash equilibrium solution of (7), can be formulated as the following quasi-Newton method

$$\begin{aligned} P_1^{(k+1)} &= P_1^{(k)} - (N_{1_{P_1^{(k)}}}^{'})^{-1} N_1(P_1^{(k)}, P_2^{(k)}) \\ P_2^{(k+1)} &= P_2^{(k)} - (N_{2_{P_2^{(k)}}}^{'})^{-1} N_2(P_1^{(k)}, P_2^{(k)}) \end{aligned}. \tag{32}$$

Proof We first show that the two equations (26)

$$\left(A^{(k)}\right)^T P_i^{(k+1)} + P_i^{(k+1)} A^{(k)} = -\overline{Q}_i^{(k)} \tag{33}$$

can be written in the form:

$$(P_1^{(k+1)} - P_1^{(k)})A^{(k)} + (A^{(k)})^T (P_1^{(k+1)} - P_1^{(k)}) + N_1(P_1^{(k)}, P_2^{(k)}) = 0 \tag{34}$$

and respectively:

$$(P_2^{(k+1)} - P_2^{(k)})A^{(k)} + (A^{(k)})^T (P_2^{(k+1)} - P_2^{(k)}) + N_2(P_1^{(k)}, P_2^{(k)}) = 0\,. \tag{35}$$

For i=1, we write (33) as:

$$\left(A^{(k)}\right)^T P_1^{(k+1)} + P_1^{(k+1)} A^{(k)} = -(Q_1 + P_1^{(k)} S_1 P_1^{(k)} + P_2^{(k)} S_{12} P_2^{(k)})\,. \tag{36}$$

Using the definition of $N_1(P_1^{(k)}, P_2^{(k)})$ we can write:

$$\begin{aligned} N_1(P_1^{(k)}, P_2^{(k)}) = (A - S_1 P_1^{(k)} - S_2 P_2^{(k)})^T P_1^{(k)} + P_1^{(k)}(A - S_1 P_1^{(k)} - S_2 P_2^{(k)}) + \\ + Q_1 + P_2^{(k)} S_{12} P_2^{(k)} + P_1^{(k)} S_1 P_1^{(k)} \end{aligned} \tag{37}$$

and thus we have

$$N_1(P_1^{(k)}, P_2^{(k)}) - \left(A^{(k)}\right)^T P_1^{(k)} - P_1^{(k)} A^{(k)} = Q_1 + P_2^{(k)} S_{12} P_2^{(k)} + P_1^{(k)} S_1 P_1^{(k)}\,. \tag{38}$$

Adding equations (36) and (38) we obtain

$$\left(P_1^{(k+1)} - P_1^{(k)}\right)A^{(k)} + \left(A^{(k)}\right)^T\left(P_1^{(k+1)} - P_1^{(k)}\right) + N_1(P_1^{(k)}, P_2^{(k)}) = 0\,. \tag{39}$$

Similarly, for *i*=2, one can obtain (35) using (33) and the definition of $N_2(P_1^{(k)}, P_2^{(k)})$.
Using (31) we write

$$N_{1_{P_1^{(k)}}}^{'}(P_1^{(k+1)} - P_1^{(k)}) = (A^{(k)})^T\left(P_1^{(k+1)} - P_1^{(k)}\right) + \left(P_1^{(k+1)} - P_1^{(k)}\right)A^{(k)} \tag{40}$$

and thus (39) becomes

$$N_{1_{P_1^{(k)}}}^{'}(P_1^{(k+1)} - P_1^{(k)}) = -N_1(P_1^{(k)}, P_2^{(k)})\,, \tag{41}$$

and the sequence of matrices $\left\{P_1^{(k)}\right\}$ will be determined using the iterative relation

$$P_1^{(k+1)} = P_1^{(k)} - (N_{1_{P_1^{(k)}}}^{'})^{-1}N_1(P_1^{(k)}, P_2^{(k)})\,. \tag{42}$$

In a similar fashion we can show that the sequence of matrices $\left\{P_2^{(k)}\right\}$ is the result of the iterative procedure $P_2^{(k+1)} = P_2^{(k)} - (N_{2_{P_2^{(k)}}}^{'})^{-1}N_2(P_1^{(k)}, P_2^{(k)})$.

5. Simulation result for the online algorithm

This section presents the results that were obtained in simulation while finding the state-feedback controllers that correspond to the Nash equilibrium solution of the differential game.
Here we considered the system used in Example 1 in (Jungers et al., 2007). The purpose of the design method is to allow the two players to determine by means of online measurements and reinforcement learning techniques the control strategies that satisfy the equilibrium characterized by (3). It is important to emphasize that the equilibrium result will be obtained without making use of any knowledge on the drift dynamics of the system, matrix *A*.
The matrices of the model of the plant, that are used in this simulation are:

$$A_{nom} = \begin{bmatrix} -0.0366 & 0.0271 & 0.0188 & -0.4555 \\ 0.0482 & -1.0100 & 0.0024 & -4.0208 \\ 0.1002 & 0.2855 & -0.7070 & 1.3229 \\ 0 & 0 & 1.0000 & 0 \end{bmatrix} \tag{43}$$

$$\begin{aligned} B_1 &= \begin{bmatrix} 0.4422 & 3.0447 & -5.52 & 0 \end{bmatrix}^T \\ B_2 &= \begin{bmatrix} 0.1761 & -7.5922 & 4.99 & 0 \end{bmatrix}^T \end{aligned} \tag{44}$$

The following cost function parameters were chosen $Q_1 = diag(3.5;2;4;5)$, $Q_2 = diag(1.5;6;3;1)$, $R_{11} = 1$, $R_{22} = 2$, $R_{12} = 0.25$, $R_{21} = 0.6$.
For the purpose of demonstrating the online learning algorithm the closed loop system was excited with an initial condition, the initial state of the system being $x_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$. The

simulation was conducted using data obtained from the system at every 0.2s. The value of the stop criterion ε was 10^{-6}.

The algorithm was initialized using the matrices $P_i^{(0)}, i=1,2$ that were calculated using the initialization procedure that was outlined above. It is important to mention here that the two admissible control policies $K_i^{(0)}, i=1,2$ corresponding to the solutions $P_i^{(0)}, i=1,2$ can also be determined online by means of the online policy iteration algorithm introduced in (Vrabie et al., 2009), a procedure that does not require knowledge on the drift dynamics of the system, namely matrix A.

In order to solve online for the values of the $P_i^{(k)}, i=1,2$, a least-squares problem of the sort described in Section 2.2 was set up before each iteration step in the online algorithm. Since there are 10 independent elements in the symmetric matrices $P_i^{(k)}, i=1,2$ 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. Here we chose to solve a least squares problem after a set of 15 data samples was acquired and thus the policy of the controller was updated every 3 sec.

Figure 2 and Figure 3 present the evolution of the parameters of the value of the game seen by Player 1 and Player 2 respectively.

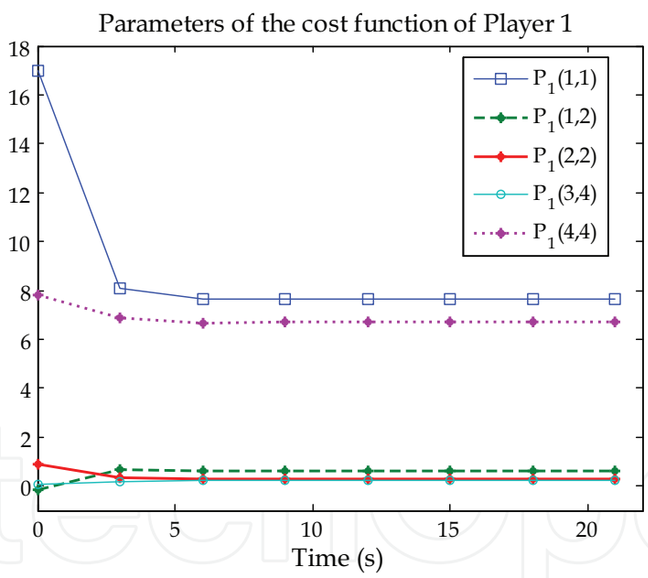


Fig. 2. Convergence of the cost function of Player 1 using the ADP method with integral reinforcement learning technique.

The matrices that are characterizing the equilibrium solution were obtained in simulation after 7 iteration steps. These are:

$$P_1^{(7)} = \begin{bmatrix} 7.6586 & 0.6438 & 0.6398 & -3.0831 \\ 0.6438 & 0.2878 & 0.2855 & -0.0945 \\ 0.6398 & 0.2855 & 0.5620 & 0.2270 \\ -3.0831 & -0.0945 & 0.2270 & 6.6987 \end{bmatrix} \tag{45}$$

and

$$P_2^{(7)} = \begin{bmatrix} 3.4579 & 0.1568 & 0.2047 & -1.8480 \\ 0.1568 & 0.6235 & 0.2889 & -0.0711 \\ 0.2047 & 0.2889 & 0.4014 & 0.0729 \\ -1.8480 & -0.0711 & 0.0729 & 3.7850 \end{bmatrix}. \quad (46)$$

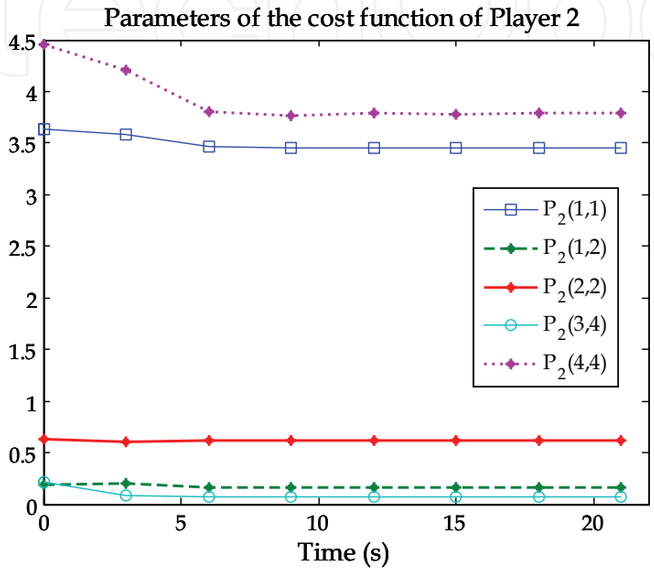


Fig. 3. Convergence of the cost function of Player 2 using the ADP method with integral reinforcement learning technique

The same results have been obtained in (Freiling et al., 1996) by means of a different iterative method.

The two saddle point control policies are:

$$K_1^{(7)} = [-1.8151 \quad 0.4150 \quad 1.9501 \quad 2.9041] \quad (47)$$

and

$$K_2^{(7)} = [-0.22 \quad 1.6323 \quad 0.0772 \quad -0.2891]. \quad (48)$$

It is important to note that the ADP online gaming method described in Section 3, uses measurements from the system and does not require any knowledge of the matrix A. Nonetheless, the resulting solution is close to the exact solution of the game problem that can be obtained via numerical methods that require an exact model of the system.

6. Conclusion

This chapter introduced an online data-based approach that makes use of reinforcement learning techniques to determine in an online fashion the solution of the two-player

nonzero-sum differential game with linear dynamics. The algorithm is suitable for online implementation and furthermore does not require exact knowledge of the system drift dynamics given by matrix A.

The two participants in the continuous-time differential game are competing in real-time and the feedback Nash control strategies will be determined based on online measured data from the system. The algorithm is built on interplay between a learning phase, where each of the players is learning online the value that they associate with a given set of play policies, and a policy update step, performed by each of the payers towards decreasing the value of their cost. The players are learning concurrently.

It was shown that the online procedure is based on a mathematical algorithm that solves offline the coupled ARE associated with the differential game problem and involves iterations on Lyapunov equations to build a sequence of controllers. The Lyapunov equations that appear at each step of the iteration are solved online using measured data by means of an integral reinforcement learning procedure.

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 detail in a future research. Also, herein we restricted the discussion to the case of two-player games. However it is straightforward to formulate the ADP algorithm for the general case with N players.

7. References

Abou-Kandil, H.; Freiling, G. & Jank, G. (2003). Necessary and sufficient conditions for constant solutions of coupled Riccati equations in Nash games, *Systems and Control Letters*, 21, pp. 295-306.

Abu-Khalaf, M., Lewis, F. L. & Huang, J. (2006). Policy Iterations and the Hamilton-Jacobi-Isaacs Equation for H-infinity State-Feedback Control with Input Saturation, *IEEE Transactions on Automatic Control,* pp. 1989- 1995.

Al-Tamimi, A., Abu-Khalaf, M. & Lewis F. L. (2007). Adaptive Critic Designs for Discrete-Time Zero-Sum Games with Application to H-infinity Control, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37, 1, pp. 240-247.

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

Beard, R., Saridis, G., & Wen, J., (1997). Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation, *Automatica*, 33, 11, pp. 2159-2177, ISSN:0005-1098.

Bellman, R. (2003). *Dynamic Programming*, Dover Publications, ISBN 0-486-42809-5, Mineola, New York.

Bertsekas D. P. & Tsitsiklis J. N. (1996). *Neuro-Dynamic Programming*, Athena Scientific, ISBN: 1-886529-10-8, Massachusetts.

Brewer J. W. (1978). Kronecker Products and Matrix Calculus in System Theory, *IEEE Trans. on Circuit and System*, 25, 9, pp. 772–781, ISSN: 0098-4094.

Cherfi L., Abou-Kandil H. & Bourles H. (2005-a). Iterative method for general algebraic Riccati equation, *Proceedings ACSE'05*, pp. 1-6, December 2005.

Cherfi L., Chitour Y. & Abou-Kandil H. (2005-b). A new algorithm for solving coupled algebraic Riccati equations, *Proceedings of CIMCA'05*, pp. 83 – 88.

Engwerda J.C. (2005). *LQ dynamic optimization and differential games*, Chichester: Wiley.

Freiling G., Jank G. & Abou-Kandil H. (1996). On Global Existence of Solutions to Coupled Matrix Riccati Equations in Closed-Loop Nash Games, *IEEE Transaction on Automatic Control*, 41, 2, pp. 264-269.

Hanselmann T., Noakes L. & Zaknich A. (2007). Continuous-time adaptive critics, *IEEE Transactions on Neural Networks*, 18, 3, pp. 631-647, ISSN: 1045-9227.

Hornik, K., Stinchcombe M. & White H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, *Neural Networks*, 3, 5, pp. 551-560, ISSN:0893-6080.

Jungers M., De Pieri E. & Abou-Kandil H. (2007). Solving Coupled Riccati Equations for Closed-Loop Nash Strategy, by Lack of Trust Approach, *International Journal of Tomography and Statistics*, 7, F07, pp. 49-54.

Kleinman, D. (1968). On an Iterative Technique for Riccati Equation Computations, *IEEE Trans. on Automatic Control*, 13, 1, pp. 114 – 115, ISSN: 0018-9286.

Kulkarni, R. V. & Venayagamoorthy, G. K. (2010), Adaptive critics for dynamic optimization, *Neural Networks*, 23, 5, pp. 587-591.

Li, T. & Gajic, Z. (1995). Lyapunov iterations for solving coupled algebraic Riccati equations of Nash differential games and algebraic Riccati equations of zero-sum games, In: *New Trends in Dynamic Games*, G. Olsder (Ed.), Birkhauser, pp. 333-351.

Mukaidani, H. (2006). Optimal numerical strategy for Nash games of weakly coupled large scale systems, *Dynamics of Continuous, Discrete and Impulsive Systems, Series B: Applications and Algorithms*, 13, pp. 249-268.

Mukaidani, H. (2007-a). Newton's method for solving cross-coupled sign-indefinite algebraic Riccati equations for weakly coupled large-scale systems, *Applied Mathematics and Computation*, 188, pp. 103-115.

Mukaidani, H. (2007-b). Numerical computation of sign-indefinite linear quadratic differential games for weakly coupled large scale systems, *International Journal of Control*, 80, 1, pp. 75-86.

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

Shah, V. (1998). *Power control for wireless data services based on utility and pricing*, M.S. thesis, Rutgers University.

Starr, A. & Ho Y. (1969). Nonzero-sum differential games, *Journal of Optimization Theory and Applications*, 3, 3, pp. 184-206.

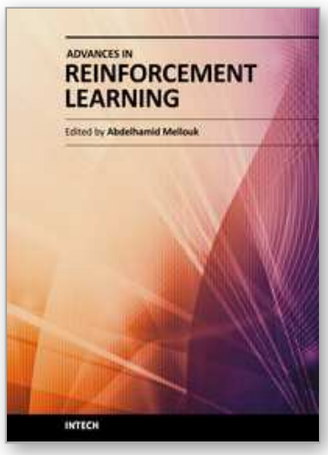
Sutton, R. (1988). Learning to predict by the method of temporal differences, *Machine Learning*, 3:9-44, ISSN: 0885-6125.

Vrabie D., Pastravanu O., Lewis F., Abu-Khalaf, M. (2009). Adaptive Optimal Control for Continuous-Time Linear Systems Based on Policy Iteration, *Automatica*, 45, 2, pp. 477-484, February 2009, ISSN:0005-1098.

Watkins, C. (1989). *Learning from Delayed Rewards*, Ph.D. Thesis, Cambridge University, Cambridge, England.

Werbos, P. J. (1992). Approximate dynamic programming for real-time control and neural modelling, In: *Handbook of Intelligent Control, Neural, Fuzzy, and, Adaptive Approaches*, White D. & Sofge D. (Eds.), New York: Van Nostrand.

Wei, Q. & Zhang, H. (2008). 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 (ICNSC'08)*, 6, 8, pp. 507 – 512, ISBN: 978-1-4244-1685-1



Advances in Reinforcement Learning

Edited by Prof. Abdelhamid Mellouk

ISBN 978-953-307-369-9
Hard cover, 470 pages
Publisher InTech
Published online 14, January, 2011
Published in print edition January, 2011

Reinforcement Learning (RL) is a very dynamic area in terms of theory and application. This book brings together many different aspects of the current research on several fields associated to RL which has been growing rapidly, producing a wide variety of learning algorithms for different applications. Based on 24 Chapters, it covers a very broad variety of topics in RL and their application in autonomous systems. A set of chapters in this book provide a general overview of RL while other chapters focus mostly on the applications of RL paradigms: Game Theory, Multi-Agent Theory, Robotic, Networking Technologies, Vehicular Navigation, Medicine and Industrial Logistic.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Draguna Vrabie and Frank L. Lewis (2011). Integral Reinforcement Learning for Finding Online the Feedback Nash Equilibrium of Nonzero-Sum Differential Games, Advances in Reinforcement Learning, Prof. Abdelhamid Mellouk (Ed.), ISBN: 978-953-307-369-9, InTech, Available from: http://www.intechopen.com/books/advances-in-reinforcement-learning/integral-reinforcement-learning-for-finding-online-the-feedback-nash-equilibrium-of-nonzero-sum-diff

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen