Preprints of the 10th IFAC International Symposium on Dynamics and Control of Process Systems
The International Federation of Automatic Control
December 18-20, 2013. Mumbai, India

Online Partially Model-Free Solution of Two-Player Zero Sum Differential Games

Praveen P * **Shubhendu Bhasin** **

* *Department of Electrical Engineering, Indian Institute of Technology Delhi, India (e-mail: eez138263@ee.iitd.ac.in)*
** *Department of Electrical Engineering, Indian Institute of Technology Delhi, India (e-mail: sbhasin@ee.iitd.ac.in)*

Abstract: An online adaptive dynamic programming based iterative algorithm is proposed for a two-player zero sum linear differential game problem arising in the control of process systems affected by disturbances. The objective in such a scenario is to obtain an optimal control policy that minimizes the specified performance index or cost function in presence of worst case disturbance. Conventional algorithms for the solution of such problems require full knowledge of system dynamics. The algorithm proposed in this paper is partially model-free and solves the two-player zero sum linear differential game problem without knowledge of state and control input matrices.

Keywords: Two-player zero sum differential game, Adaptive dynamic programming, Approximate Dynamic Programming

1. INTRODUCTION

In a two-player zero sum game, each player's gain is exactly balanced by the loss of his competitor and the net gain of the players at any point in time is zero [Isaacs (1965)]. In scenarios of perfect competition such as that exist in a zero sum game, each player tries to make the best possible decision taking into account the fact that his opponent also tries to do the same. The theory of zero sum differential game finds applications in a wide variety of disciplines including that of control theory [Tomlin et al. (2000); Wei and Liu (2012)].

The problem of designing optimal controller for any process system subject to worst possible disturbance is a minimax optimization problem where the controller tries to minimize and the disturbance tries to maximize the infinite horizon quadratic cost. Using game theoretic framework, the above minimax optimization problem can be viewed as a zero sum differential game where the controller acts as the 'minimizer' or minimizing player, while the disturbance acts as the 'maximizer' or maximizing player [Basar and Bernhard (1995); Basar and Olsder (1995)]. The optimal control in such a scenario is equivalent to finding the Nash equilibrium or saddle point equilibrium of the corresponding two-player zero sum differential game [Basar and Bernhard (1995)].

To obtain the saddle point equilibrium strategy of each player in the two-player zero sum linear differential game, one needs to solve an Algebraic Riccati Equation (ARE) with a sign indefinite quadratic term known as the Game Algebraic Riccati Equation (GARE). [Kleinman (1968)] proposed an iterative method for solving the ARE with a sign definite quadratic term. A series of Lyapunov equations are constructed at each iteration and the positive semi-definite solutions of the Lyapunov equations are shown to converge to the solution of ARE. However the algorithm proposed in [Kleinman (1968)] for solving the ARE could not be extended to the GARE due to the presence of a sign indefinite quadratic term.

Subsequently, several Newton-type algorithms were proposed for the solution the GARE [Arnold and Laub (1984); Damm and Hinrichsen (2001); Mehrmann and Tan (1988)]. [Mehrmann (1991)] and [Sima (1996)] proposed a matrix sign function method for solving GARE. A more robust iterative algorithm for solving GARE was proposed by [Lanzon et al. (2008)], where the GARE with a sign indefinite quadratic term is replaced by a sequence of AREs with sign definite quadratic terms. Each of these AREs can then be sequentially solved using Kleinman's algorithm or any other existing algorithm and the recursive solution of these AREs converge to the solution of GARE. However, all the above results require knowledge of full system dynamics, which is a severe restriction owing to the uncertainties in system modelling.

The concept of Adaptive Dynamic Programming (ADP) was proposed by [Werbos (1992)] for solving the dynamic programming problems related to classical optimal control in a forward in time fashion. ADP is based on the concepts of Dynamic Programming [Bellman (2003)] and Reinforcement Learning [Sutton and Barto (1998)] and has been widely used to reach approximate solutions of optimal control problems [Abu-Khalaf and Lewis (2005); Lewis and Liu (2013)]. The classical optimal control problem is a single player linear differential game problem [Isaacs (1965)] and a detailed analysis of the application of ADP for the solution of single player linear differential game problems (optimal control problems) is provided in [Wang et al. (2009)] and [Lewis et al. (2012)].

Recently, [Vrabie et al. (2009)] proposed a partially model-free, online algorithm for finding the solution of single player zero sum differential game in continuous time. However the proposed algorithm required partial knowledge of system dynamics, namely the control input matrix. Subsequently, a completely model-free algorithm was proposed by [Jiang and Jiang (2012)] in order to arrive at the optimal control of linear system without requiring knowledge of the control input and the system drift matrices.

Extending the principle of ADP to differential games, a model-free algorithm was proposed by [Al-Tamimi et al. (2007)] for solving the two-player linear differential zero sum game appearing in discrete time H_∞ control. [Abu-Khalaf et al. (2008)] proposed an algorithm to solve continuous time two-player nonlinear differential game problem with full knowledge of system dynamics. In [Vrabie and Lewis (2011)], the saddle point solution of a two player zero sum differential game was obtained using a partially model-free algorithm. The algorithm required no knowledge of system drift dynamics, while the knowledge of control input and disturbance matrices was still needed.

In this paper, an online partially model free iterative algorithm is proposed to solve the two-player zero sum differential game problem. The contribution of this result over existing results in the literature is that the saddle point solution is arrived at without the knowledge of system drift and control input matrices. Only knowledge of the disturbance input matrix is required and hence, the proposed algorithm is comparatively more robust than the one proposed in [Vrabie and Lewis (2011)]. The proposed algorithm thereby reduces the need of rigorous system identification requirements, which otherwise is necessary for obtaining the system drift and control input matrices. The algorithm is data-driven and the saddle point solution of the game problem can be achieved online.

The algorithm makes use of two iterative steps namely, the outer level iteration and the inner level iteration. In the outer level iteration, the two-player zero sum differential game problem is converted to a sequence of optimal control problems. The inner level iteration motivated by the work of [Jiang and Jiang (2012)], is used to find the model-free solution of each of these optimal control problems. It is shown that the solution of optimal control problems converge to the saddle point equilibrium solution of the underlying game problem. The algorithm is implemented on a linearized power system model [Wang et al. (1963)] to arrive at the optimal controller in presence of worst case disturbance. The algorithm can be extended to any process system with partially unknown dynamics.

2. PRELIMINARIES

2.1 Problem Formulation

The H_∞ controller design was formulated by [Basar and Bernhard (1995)] as a two-player zero sum differential game. Consider the differential game formulation given below,

$$\dot{x} = Ax + B_1 w + B_2 u \tag{1}$$

$$y = Cx, \tag{2}$$

where $x \in R^n$ is the state vector assumed to be measurable, $u \in R^m$ is the control input and $w \in R^d$ is the disturbance input.

The controller player's objective is to minimize the infinite horizon cost function given by,

$$J(x(0), u(t), w(t)) = \frac{1}{2}\int_0^\infty (x^T C^T C x + u^T u - w^T w)dt. \tag{3}$$

At the same time, the disturbance player who is in perfect competition with the controller player tries to maximize the cost function (3). The optimal strategies to be adopted by the players in the zero sum game is termed as the *saddle-point equilibrium.*

The saddle point equilibrium strategy for each player is defined as follows:-

Definition 1. A pair of strategies $\{u^*, w^*\}$ is in *saddle point equilibrium*, if the following set of inequalities are satisfied for all permissible u and w [Basar and Olsder (1995)]:

$$J(x, u^*, w) \leq J(x, u^*, w^*) \leq J(x, u, w^*) \tag{4}$$

The quantity $J(x, u^*, w^*)$ is called the *saddle point value* of the zero sum game. The two-player zero sum differential game described above is a two-player optimization problem and the *saddle point strategies* can be found by using the Pontryagin's Maximum Principle.

Hamiltonian for the cost function in (1) and the system in (3) can be defined as,

$$H(x, u, v, \lambda) = \frac{1}{2}(x^T C^T C x + u^T u - w^T w) + \lambda^T (Ax + B_1 w + B_2 u). \tag{5}$$

Applying the necessary condition for optimality of $\frac{\partial H}{\partial u} = 0$ and $\frac{\partial H}{\partial w} = 0$,

$$u^* = -B_2^T \Pi^* x(t) \tag{6}$$

$$w^* = B_1^T \Pi^* x(t), \tag{7}$$

where Π^* is the positive definite symmetric solution to the ARE given by,

$$0 = A^T \Pi^* + \Pi^* A + C^T C - \Pi^* (B_2 B_2^T - B_1 B_1^T) \Pi^*. \tag{8}$$

The saddle point equilibrium value is given by ,

$$J(x(t), u^*, w^*) = x(t)^T \Pi^* x(t). \tag{9}$$

Hence, to obtain the saddle point equilibrium strategy of each player in the two-player zero sum differential game, one needs to solve the ARE (8) with the sign indefinite quadratic term, $\Pi(B_2 B_2^T - B_1 B_1^T)\Pi$.

2.2 Iterative Solution Using Knowledge of Full System Dynamics

The iterative solution of the GARE (8) with full knowledge on system dynamics was given by [Lanzon et al. (2008)]. Consider real matrices *A,* B_1, B_2 and *C*, where *(C,A)* is observable and *(A,* B_2*)* is stabilizable. Then, the iterative method for solving the ARE (8) is given as [Lanzon et al. (2008)]:

1. Start with

$$\Pi^0 = 0. \tag{10}$$

2. Solve, for the unique positive definite Z^k where $k \geq 0$,

$$\begin{aligned} 0 = (A + B_1 B_1^T \Pi^{k-1} - B_2 B_2^T \Pi^{k-1})^T Z^k \\ + Z^k (A + B_1 B_1^T \Pi^{k-1} - B_2 B_2^T \Pi^{k-1}) \\ - Z^k B_2 B_2^T Z^k + F(\Pi^{k-1}), \end{aligned} \tag{11}$$

where,

$$\begin{aligned} F(\Pi^{k-1}) = A^T \Pi^{k-1} + \Pi^{k-1} A + C^T C \\ - \Pi^{k-1} (B_2 B_2^T - B_1 B_1^T) \Pi^{k-1}. \end{aligned} \tag{12}$$

3. Update

$$\Pi^k = \Pi^{k-1} + Z^k. \tag{13}$$

where the Π^k and Z^k possess the following properties:

(1) $(A + B_1 B_1^T \Pi^k, B_2)$ is stabilizable $\forall k$
(2) $F(\Pi^{k+1}) = Z_k B_1 B_1^T Z_k \ \forall k$
(3) $A + B_1 B_1^T \Pi^k - B_2 B_2^T \Pi^{k+1}$ is Hurwitz $\forall k$
(4) $\Pi^* \geq \Pi^{k+1} \geq \Pi^k \geq 0 \ \forall k$

Then,

$$\lim_{k \to \infty} \Pi^k = \Pi^*. \tag{14}$$

The convergence proof of the algorithm to the unique positive definite solution, Π^* of the GARE is provided by [Lanzon et al. (2008)].

3. MAIN RESULTS

3.1 Solution Approach

In the two-player zero sum differential game defined through (1) and (3), the controller and the disturbance are two competing players. If any of the players adopts a constant strategy, it becomes a single player optimization problem or a single player differential game problem (optimal control problem). In the proposed game strategy, the controller acts as the active, optimising player and disturbance acts a passive player, whose policy remains constant during a time frame. With disturbance policy remaining constant, the two-player zero sum game takes the shape of a classical optimal control problem.

The solution approach proposed is to break down the original two-player zero sum differential game into a sequence of optimal control problems. The active controller player seeks the optimal policy which minimizes the cost function of the associated optimal control problem. After the controller player converges to his optimal policy, the disturbance player updates his policy accordingly. With the changed disturbance policy, a new optimal control problem is constructed and the controller player then seeks the new optimal policy.

The aim of this paper is to arrive at the saddle point solution of the game with less dependency on the knowledge of system dynamics. This objective translates to solving each of the optimal control problem without knowledge of A and B_2 matrices. The proposed algorithm only requires knowledge of B_1 for implementation.

The game solution is achieved using two iterative steps namely outer level iteration and the inner level iteration. In each outer level iteration, an optimal control problem is constructed from the underlying game problem. The inner level iteration process is embedded within each outer level iteration and finds the online solution of each of those optimal control problems. It is shown that the inner level iterations converge to the solution of the corresponding optimal control problem and the outer level iterations converge to the saddle point solution of the game problem.

Outer Level Iteration

The outer level iteration is motivated by [Lanzon et al. (2008)], where an optimal control problem is constructed considering that the disturbance player acts as a passive player whose policy remains constant during each iteration. The controller player, after considering the disturbance player's policy, then minimizes the cost function by solving the corresponding optimal control problem.

Let at the k^{th} outer level iteration, the disturbance player chooses the strategy, $w = B_1^T \Pi^{k-1} x$. Considering that the strategy of the passive disturbance player remains constant, the game cost function (3) can be transformed into the following optimal control form,

$$\begin{aligned} J(x(0), u(t)) = \\ \frac{1}{2} \int_0^\infty (x^T (C^T C - \Pi^{k-1} B_1 B_1^T \Pi^{k-1}) x + u^T u) dt. \end{aligned} \tag{15}$$

subject to system dynamics given by,

$$\dot{x} = (A + B_1 B_1^T \Pi^{k-1}) x + B_2 u. \tag{16}$$

Lemma 2. The solution of optimal control problem defined by (15) and (16) is equivalent to the iteration steps defined by (11) and (13).

Proof. The solution of the optimal control problem defined in (15) and (16) can be found by solving the ARE,

$$\begin{aligned} 0 = (A + B_1 B_1^T \Pi^{k-1})^T \Pi^k + \Pi^k (A + B_1 B_1^T \Pi^{k-1}) \\ - \Pi^k B_2 B_2^T \Pi^k + (C^T C - \Pi^{k-1} B_1 B_1^T \Pi^{k-1}). \end{aligned} \tag{17}$$

In the equation (11), substituting for $F(\Pi^{k-1})$ from (12) and Z_k from (13) result in the ARE (17), which proves the lemma.

At the k^{th} outer level iteration, the optimal control problem to minimize the cost function (15) will be found subject to the modified system dynamical equation (16) resulting in the optimal values, Π^k and K^k. The controller strategy then becomes, $u_k = -K^k x$. The disturbance being a passive player, adopts the policy $w_k = B_1^T \Pi^k x$. With the new disturbance policy, the outer level iteration is carried out where the optimal control problem defined by (15) and (16) are modified and the optimal control solution is sought again to obtain $K^{k+1}$and Π^{k+1}.

The process is continued till convergence where $\| K^{k+1} - K^k \| \ll \epsilon_1$, where ϵ_1 is a specified threshold. At convergence, the policies $u_k = -K^{k+1} x$ and $w_k = B_1^T \Pi^{k+1} x$ converge to the saddle point equilibrium values $u^* = -K^* x$ and $w^* = B_1^T \Pi^* x$, for controller and disturbance respectively.

Inner Level Iteration

The optimal control problem constructed in each outer level iteration is solved using the inner level iteration process. The inner level iteration is an ADP algorithm based on the work by [Jiang and Jiang (2012)]. It comprises of a policy evaluation stage and a policy improvement stage. At the policy evaluation stage, the current control policy is evaluated and the policy is updated in the policy improvement stage.

Lemma 3. The optimal control problem defined by (15) and (16) can be solved by sequentially using the following Lyapunov equations [Kleinman (1968)],

$$(A_i^k)^T\Pi_i + \Pi_i A_i^k + C^TC - \Pi^{k-1}B_1B_1^T\Pi^{k-1} + (K_i)^TK_i = 0. \quad (18)$$

and performing policy update as,

$$K_{i+1} = B_2^T\Pi_i, \quad (19)$$

where $A_i^k = A + B_1B_1^T\Pi^{k-1} - B_2K_i$ and i is the iteration. Then,

$$\lim_{i\to\infty} K_i = K^k.$$

Proof. The proof is provided by [Kleinman (1968)].

The Kleinman's algorithm guarantees the convergence of K_i to the optimal value of k^{th} outer level iteration, K_k. With an intention to arrive at the solution of optimal control problem without knowledge on system matrices A and B_2, the equation (16) is reformulated as [Jiang and Jiang (2012)],

$$\dot{x} = A_i^k x + B_2(u + K_i x), \quad (20)$$

where $A_i^k = A + B_1B_1^T\Pi^{k-1} - B_2K_i$, k is the outer level iteration stage and i is the current inner level iteration stage. With quadratic parametrization, the infinite horizon cost (15) at i^{th} inner level iteration of k^{th} outer level iteration can be written as,

$$J_i(x(t)) = x(t)^T\Pi_i x(t)$$

where Π_i is a positive definite symmetric matrix.

Then using (18), (19) and (20),

$$\begin{aligned}\frac{dJ_i}{dt} &= \dot{x}(t)^T\Pi_i x(t) + x(t)^T\Pi_i\dot{x}(t)\\ &= (A_i^k x + B_2(u+K_ix))^T\Pi_i x(t)\\ &\quad + x(t)^T\Pi_i(A_i^k x + B_2(u+K_ix))\\ &= x(t)^T((A_i^k)^T\Pi_i + \Pi_i A_i^k)x(t) + 2(u+K_ix)^TB_2^T\Pi_i x(t)\\ &= -x(t)^T(C^TC - \Pi^{k-1}B_1B_1^T\Pi^{k-1} + {K_i}^TK_i)x(t)\\ &\quad + 2(u+K_ix)^TB_2^T\Pi_i x(t) \quad (21)\end{aligned}$$

Then using $K_{i+1} = B_2^T\Pi_i$ from (19), it follows from (21) that,

$$\begin{aligned}&x(t)^T\Pi_i x(t) - x(t+T)^T\Pi_i x(t+T)\\ &= \int_t^{t+T} x^T(C^TC - \Pi^{k-1}B_1B_1^T\Pi^{k-1} + {K_i}^TK_i)d\tau\\ &\quad - 2\int_t^{t+T}(u+K_ix)^TK_{i+1}x d\tau, \quad (22)\end{aligned}$$

where i is the inner iteration stage.

The LHS of the equation (22) can be reformulated as

$$x(t)^T\Pi_i x(t) - x(t+T)^T\Pi_i x(t+T) = (\bar{x}(t) - \bar{x}(t+T))^T\hat{\Pi}_i. \quad (23)$$

where $\bar{x}(t)$ is a modified Kronecker product vector with elements $\{x_p(t)x_q(t)\}_{p=1:n;q=1:n}$ and $\hat{\Pi}_i$ is a vector formed by stacking the diagonal and upper triangular part of a symmetric matrix into a vector whose off diagonal terms are multiplied by two.

The integrand in the second part of RHS can then be modified as

$$\begin{aligned}&(u+K_ix)^TK_{i+1}x =\\ &\quad [(x^T\otimes u^T) + (x^T\otimes x^T)(I_n\otimes(K_i)^T)]\hat{K}_{i+1}. \quad (24)\end{aligned}$$

where $\otimes$ denotes Kronecker product and $\hat{K}_{i+1}$ is the vector form of matrix K_{i+1} stacking columns one over another.

Now using (23) and (24), the entire equation (22) can be rewritten as,

$$\begin{aligned}&(\bar{x}(t) - \bar{x}(t+T))^T\hat{\Pi}_i\\ &\quad + 2[(x^T\otimes u^T) + (x^T\otimes x^T)(I_n\otimes K_i^T)]\hat{K}_{i+1}\\ &= \int_t^{t+T} x^T(C^TC - \Pi^{k-1}B_1B_1^T\Pi^{k-1} + (K_i)^TK_i)d\tau. \quad (25)\end{aligned}$$

The equation (25) can be transformed into linear regression form given by

$$X\theta = Y \quad (26)$$

where

$X = [(\bar{x}(t) - \bar{x}(t+T))^T \quad 2((x^T\otimes u^T) + (x^T\otimes x^T)(I_n\otimes K_i^T))]$,

the regression vector

$$\theta = [(\hat{\Pi}_i)^T(\hat{K}_{i+1})^T]^T,$$

and

$$Y = \int_t^{t+T} x^T(C^TC - \Pi^{k-1}B_1B_1^T\Pi^{k-1} + (K_i)^TK_i)d\tau.$$

Assuming C and B_1 are known, the cost X can be measured along the state trajectory. After getting enough measurements in Y and X, θ can be obtained as a least square estimate without knowledge of A and B_2.

$$\hat{\theta} = (X^TX)^{-1}X^TY. \quad (27)$$

The above estimation stage is the policy evaluation stage of ADP, where Π_i and K_{i+1} are estimated from measurements of the system states. These measurements can be considered as reinforcements from the system, analogous to the one described in the theoretical framework of reinforcement learning. The minimum number of measurements required for the solution of the least square problem (27) equals the number of unknowns in the regression vector, θ. After the policy evaluation stage, policy updation stage is performed where the current policy is improved as $u_{i+1} = -K_{i+1}x + e$, where e is the exploration signal.

Remark 4. The algorithm which is based on the ADP concepts of policy evaluation and policy updation requires persistence of excitation (PE) so that (27) is a well posed least square problem. In cases wherein the system states reach a stationary phase, persistence of excitation condition fails and it may affect the numerical stability [Lee et al. (2012)] of the algorithm. Hence, exploration signals

are added to the control input as it ensures that system states never become stationary and hence satisfy the PE condition. Any sufficiently exciting signal like random noise or sinusoidal signal can be used as exploration signal.

The inner level iteration defined by (26) is done again with measurements obtained with new controller policy, u_{i+1} and the process is continued till convergence. A threshold ϵ_2 is specified and in each outer level iteration, the inner level iterations are carried out till $\| K_{i+1} - K_i \| \ll \epsilon_2$. The optimal control problem specified by (15) and (16) is solved once inner level iteration is completed . The optimal values K_i^* and Π_i^* are then be passed on for updation in next outer level iteration as K^k and Π^k.

Remark 5. As K^k is directly estimated through the inner level iteration process, the optimal control can be arrived at with out the knowledge of A and B_2 matrices.

3.2 Algorithm

The proposed algorithm consists of outer level and inner level iterative stages detailed as follows:-

(1) Start the algorithm with an initial stabilizing policy K^0 and $\Pi^0 = 0$.
(2) Let k be the current outer iteration stage with known values for $K^{k-1}$and Π^{k-1}. The disturbance policy is defined as $w_k = B_1 \Pi^{k-1} x$.
(3) *Outer level iteration* :- With the disturbance policy remaining constant, solve the corresponding optimal control problem defined by (15) and (16) for Π^k and K^k using the following inner level iteration.
(4) *Inner level iteration* :-
   (a) Solve online for Π_i and K_{i+1} from state measurements where i denotes the inner level iteration stage,

$$\begin{aligned}&(\bar{x}(t) - \bar{x}(t+T))^T \Pi_i \\ &+ 2[(x^T \otimes u^T) + (x^T \otimes x^T)(I_n \otimes K_i^T)] K_{i+1} \\ &= \int_t^{t+T} x^T (C^T C - \Pi^{k-1} B_1 B_1^T \Pi^{k-1} + K_i^T K_i) d\tau.\end{aligned} \tag{28}$$

   (b) Update $u = -K_{i+1} x + e$.
   (c) Continue inner level iteration till $\| K_{i+1} - K_i \| \ll \epsilon_2$.
(5) Set $K^k = K_i^*$ and $\Pi^k = \Pi_i^*$.
(6) Update disturbance policy as $w_{k+1} = B_1^T \Pi^k x$. Modify the optimal control problem described by (15) and (16) using updated disturbance policy.
(7) Continue outer level iteration(Steps (3) to (6)) till $\| K^k - K^{k-1} \| \ll \epsilon_1$.
(8) Set saddle point policy, $K^* = K^k$.

Remark 6. In the Step (1) of the algorithm, the initial stabilizing gain K^0 can be obtained by using some nominal knowledge about the system, e.g for stable systems K^0 can be chosen.

3.3 Convergence Analysis.

Theorem 7. The proposed algorithm defined by the outer level and inner level iterative stages (Steps (1) to (6) in Section 3.2) converge to the saddle point solution of the two-player zero sum game. The proposed algorithm creates a sequence of control policies, $\{\mathrm{K}^k,\ k= 1,2,3...\}$ which converges to the saddle point control policy, K^* so that

$$\lim_{k \to \infty} ||K^k - K^*|| = 0.$$

Proof.

The proposed algorithm is based on outer and inner level iterative stages. Convergence of proposed algorithm is proved by proving convergence of inner level and outer level iterations. Since the inner level iteration is embedded in the outer level iteration, the convergence of inner level iteration is analyzed first.

The inner level iteration based on (28) is solved using least square estimate of K_{i+1} and Π_i using (27). Since (28) is derived from (18) and (19) using (20), (21), (22) and (23), the inner level iteration stage is equivalent to Kleinman's iteration (18) and (19). Hence convergence of inner level iteration is proved using *Lemma 3.*

The outer level iterations are based on the solution of (15) and (16), which is equivalent to the solution of (11) and (13) using *Lemma 2.* Then using [Lanzon et al. (2008)], convergence of outer level iteration is proved. The convergence results of inner level and the outer level iterations guarantee convergence of the proposed algorithm to the saddle point solution of the two-player zero sum differential game.

Remark 8. The proposed algorithm assumes knowledge of the disturbance input matrix while not requiring the knowledge of drift dynamics and control input matrix.

4. SIMULATION RESULTS

The algorithm is implemented on a linearized power system model proposed in [Wang et al. (1963)]. The system matrices of the nominal model are

$$A = \begin{bmatrix} -0.0665 & 8 & 0 & 0 \\ 0 & -3.663 & 3.663 & 0 \\ -6.86 & 0 & -13.736 & -13.736 \\ 0.6 & 0 & 0 & 0 \end{bmatrix},$$

$$B_2 = [\,0\ 0\ 13.736\ 0\,]^T, [B_1 = [\,1\ 0\ 5\ 0\,]^T,$$

$$C = I_4. \tag{29}$$

With complete knowledge of system dynamics, the feedback gain of the controller player corresponding to the saddle point equilibrium can be calculated using the algorithm put forward by [Lanzon et al. (2008)] as,

$$K_{saddle} = [\,1.2244\ 2.3407\ 0.8455\ 0.6052\,] \tag{30}$$

The adaptive algorithm proposed in this paper does not require the knowledge of A and B_2 and only the knowledge of B_1 is required. Using the proposed algorithm with random noise as exploration signal, the following saddle point policy is obtained after four outer level iterations and eleven inner level iterations.

$$K_{alg} = [\,1.2244\ 2.3404\ 0.8455\ 0.6051\,] \tag{31}$$

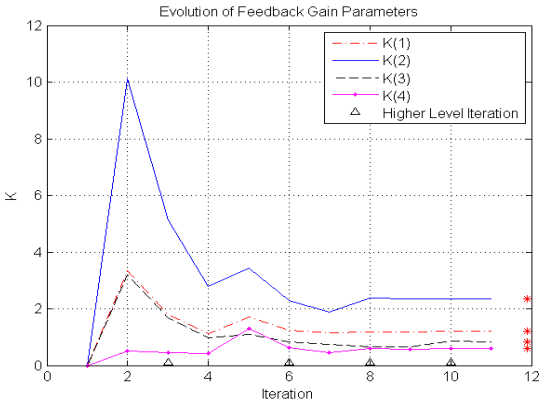

Fig. 1. Evolution of Feedback Gain

5. CONCLUSION

An adaptive dynamic programming based algorithm for finding an online saddle point solution of two-player zero sum linear differential games is presented in this paper. The proposed algorithm is capable of arriving at the saddle point solution of the game problem with partial knowledge of system dynamics, i.e only the knowledge of disturbance input matrix is required. The algorithm is used to find the optimal control policy in a power system under worst case disturbance. Future efforts will be focussed on making the algorithm completely model-free so that saddle point solution can be achieved without any knowledge system dynamics.

REFERENCES

Abu-Khalaf, M., Lewis, F., and Huang, J. (2008). Neurodynamic programming and zero-sum games for constrained control systems. *IEEE Transactions on Neural Networks*, 19(7), 1243–1252.

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

Al-Tamimi, A., Lewis, F.L., and Abu-Khalaf, M. (2007). Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control. *Automatica*, 43(3), 473 – 481.

Arnold, W.F., I. and Laub, A. (1984). Generalized eigenproblem algorithms and software for Algebraic Riccati Equations. *Proceedings of the IEEE*, 72(12), 1746–1754.

Basar, T. and Bernhard, P. (1995). *H-infinity Optimal Control and Related Minimax Design Problems*. Birkhauser.

Basar, T. and Olsder, G.J. (1995). *Dynamic Non-Cooperative Game Theory*. SIAM.

Bellman, R.E. (2003). *Dynamic Programming*. Dover Publications.

Damm, T. and Hinrichsen, D. (2001). Newton's method for a rational matrix equation occurring in stochastic control. *Linear Algebra and its Applications*, 332(0), 81 – 109.

Isaacs, R. (1965). *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*. Dover Publications.

Jiang, Y. and Jiang, Z. (2012). Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica*, 48(10), 2699–2704.

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

Lanzon, A., Feng, Y., Anderson, B., and Rotkowitz, M. (2008). Computing the positive stabilizing solution to Algebraic Riccati Equations with an indefinite quadratic term via a recursive method. *IEEE Transactions on Automatic Control*, 53(10), 2280 –2291.

Lee, J.Y., Park, J.B., and Choi, Y.H. (2012). Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems. *Automatica*, 48(11), 2850 – 2859.

Lewis, F. and Liu, D. (2013). *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. Wiley-IEEE Press.

Lewis, F., Vrabie, D., and Vamvoudakis, K. (2012). Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine*, 32(6), 76–105.

Mehrmann, V. and Tan, E. (1988). Defect correction method for the solution of Algebraicl Riccati Equations. *IEEE Transactions on Automatic Control*, 33(7), 695–698.

Mehrmann, V. (1991). *The Autonomous Linear Quadratic Control Problem: Theory and Numerical Solution*. Lecture Notes in Control and Information Sciences. Springer.

Sima, V. (1996). *Algorithms for Linear Quadratic Optimization*. Marcel Dekker Incorporated.

Sutton, R.S. and Barto, A.G. (1998). *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. A Bradford Book.

Tomlin, C., Lygeros, J., and Sastry, S. (2000). A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE*, 88(7), 949–970.

Vrabie, D., Pastravanu, O., Abu-Khalaf, M., and Lewis, F. (2009). Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica*, 45(2), 477–484.

Vrabie, D. and Lewis, F. (2011). Adaptive dynamic programming for online solution of a zero-sum differential game. *Journal of Control Theory and Applications*, 9(3), 353–360.

Wang, F.Y., Zhang, H., and Liu, D. (2009). Adaptive dynamic programming: An introduction. *IEEE Computational Intelligence Magazine*, 4(2), 39–47.

Wang, Y., Zhou, R., and Wen, C. (1963). Robust load-frequency controller design for power systems. *IEE Proceedings C Generation, Transmission and Distribution*, 140(1), 11–16.

Wei, Q. and Liu, D. (2012). Self-learning control schemes for two-person zero-sum differential games of continuous-time nonlinear systems with saturating controllers. In *International conference on Advances in Neural Networks*, 534–543.

Werbos, P. (1992). Approximate dynamic programming for real-time control and neural modeling. *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*.