

Center



Discussion Paper

No. 2006–109

**ALGORITHMS FOR COMPUTING NASH EQUILIBRIA IN
DETERMINISTIC LQ GAMES**

By Jacob Engwerda

October 2006

ISSN 0924-7815

Algorithms for computing Nash equilibria in deterministic LQ games.

Jacob Engwerda
Tilburg University
Dept. of Econometrics and O.R.
P.O. Box: 90153, 5000 LE Tilburg, The Netherlands
e-mail: engwerda@uvt.nl

Abstract In this paper we review a number of algorithms to compute Nash equilibria in deterministic linear quadratic differential games. We will review the open-loop and feedback information case. In both cases we address both the finite and the infinite-planning horizon.

Keywords: Algebraic Riccati equations; linear quadratic differential games; Nash equilibria.

Jel-codes: C61, C72, C73.

1 Introduction

Many problems in economics and management are dynamic and involve strategic considerations. That is, actions taken by a decision maker not only have an effect on the current period but also on the future and they usually have a direct impact on variables that are important for other decision makers as well. This naturally leads to reactions of these other decision makers. In case they like the impact a consequence may be that they support the decision maker in his actions, which may even result in a cooperation between the decision makers. On the other hand, if the impact is disliked by some other decision maker, that agent may try to counterbalance the impact on the variable by taking actions himself. In most cases such a noncooperative behaviour will lead to a situation which can be improved upon by both decision makers if they would cooperate. It may even be the case that the decision maker takes the action with the intention to eliminate the other decision maker from the market so that, after some reduction in his short term profits, in the long run he will gain. Topical examples may be found in the liberalization of markets which previously were controlled by the government. More or less by definition (since entering costs are usually prohibitive), these liberalized markets will be controlled by only a few players. Once a market has been liberalized (which is of common interest for all initial players) a scenario as sketched above may then take place. This may yield ultimately a worse outcome for a large majority of all those which are represented by the government (citizens). In such a case it is in the interest of the government to come up either with rules which prevent such an outcome or not to liberalize these markets.

Another example can be found in the global warming of the climate. There are a few players who contribute a lot to the global warming. However they also profit at this moment a lot from it. Since

the consequences of global warming will be visible only after the current players have died, there is almost no incentive for them to invest in counteracting measures (from which they would lose most). So, though there is a common (vague) objective in the future all players have an incentive to contribute as less as possible at this moment to realize this goal.

These are just two examples demonstrating the need to model situations in economics and management that are characterized by multiple decision makers/players and enduring consequences of decisions. Moreover, the examples demonstrate that a careful modeling may be a tough job (and maybe sometimes even impossible). This, since problems not only invoke the modeling of markets but may be also affected by, e.g., the institutional organization of a country, property rights, division of power etc. On the other hand, sometimes, a trade-off is possible between model complexity and inclusion of model uncertainty within a relatively simple model.

The aim of this paper is to present for a class of simple deterministic models numerical tools for solving them. For simplicity results will be presented only for the two-player case. In particular this class of games includes the so-called zero-sum games. However, results will not be elaborated here for that special case. Uncertainty can be incorporated in various ways within the considered class of models. Results on a stochastic and a worst-case scenario approach, respectively, as well as elaborated results for the zero-sum game can be found e.g. in [9].

The theory which conceptualizes problems which involve more than one player is dynamic games. Dynamic game theory tries to arrive at appropriate models describing a process that evolves over time. Depending on the specific problem this model sometimes can be used by an individual decision maker to optimize his performance. In other cases it may serve as a starting point to introduce new communication lines which may help to improve upon the outcome of the current process. Furthermore it is possible by the introduction of "nature" as an additional player in games, which is trying to work against the other decision makers in a process, to analyze the robustness of strategies of players w.r.t. worst-case scenarios.

Further examples of dynamic games in economics and management science can be found e.g. in [7], [14] and [29].

As already mentioned, in this paper we consider a special class of dynamic games. We study games where the process can be modeled by a set of linear differential equations and the preferences are formalized by quadratic utility functions. The so-called linear quadratic differential games. These games are very popular in literature and a recent exposition (and additional references) of this theory can be found in [9]. The popularity of these games is caused on the one hand by practical considerations. To some extent these kinds of differential games are analytically and numerically solvable. On the other hand this linear quadratic setting naturally appears, e.g., if the decision makers' objective is to minimize the effect of a small perturbation of their optimally controlled nonlinear process. By solving a linear quadratic control problem, and using the optimal actions implied by this problem, players can avoid most of the additional cost incurred by this perturbation.

In a dynamic game, information available to the players at the time of their decisions plays an important role and, therefore, has to be specified before one can analyze these kind of games appropriately. We will distinguish two cases: the open-loop and the feedback information case, respectively. In the open-loop information case it is assumed that all players know just the initial state of the process and the model structure. More specifically, it is assumed that players simultaneously determine their actions for the whole planning horizon of the process before it starts. Next they submit their actions to some authority who then enforces these plans as binding commitments. So players can not react on any deviations occurring during the evolution of the process. In the feedback

information case it is assumed that all players can observe at every point in time the current state of the process and determine their actions based on this observation.

An advantage of feedback over open-loop information is, e.g., that it requires no precommitment of the players. The proposed feedback actions remain optimal from the point of view of an individual player even if the state of the system differs at some point in time from the ex ante expected realization of the state. A property which is in literature known as subgame perfectness.

In the rest of this paper we will concentrate on the case that players do not cooperate with each other in order to arrive at their decisions. This may be caused by individual motivations or for physical reasons. Under these conditions it seems reasonable that all players individually will try to play actions which are optimal for them. That is, for actions they can not improve upon themselves. If there exists a set of actions such that none of the players has an incentive to deviate from his action (or stated otherwise, given the actions of the other players his choice of action is optimal), we call such a set of actions a Nash¹ equilibrium of the game. In general, a game may either have none, one or more than one Nash equilibrium. This leads on the one hand to the question under which conditions these different situations will occur and on the other hand, in case there is more than one equilibrium solution, whether there are additional motivations to prefer one equilibrium outcome to another.

As already indicated above linear quadratic differential games have been studied a lot in the past. In this paper we will review some basic results and present algorithms to compute equilibria for the open-loop and feedback information case. A more detailed exposition, additional results and references can be found in [9]. In particular this reference contains proofs of results that will be quoted in this paper. One can find there, e.g., also the relationship between the Bellman equations/Hamiltonians and the various Riccati equations that arise further on in this review.

The outline of the rest of the paper is as follows. Section 2 considers the "one-player" case. That is, the solution of the ordinary linear quadratic optimal control problem. The solution of this problem plays an important role in solving the various games studied in the subsequent sections. Section 3 considers the open-loop information case and section 4 the feedback information case. Finally section 5 reviews some extensions that can be found elsewhere in literature.

2 The one-player case

In this section we consider the minimization w.r.t. $u(\cdot)$ of

$$J = \int_0^{t_f} \{x^T(t)Qx(t) + u^T(t)Ru(t)\}dt + x^T(t_f)Q_{t_f}x(t_f), \quad (1)$$

subject to

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0, \quad (2)$$

where R is a positive definite matrix and Q, R and Q_{t_f} are symmetric matrices of appropriate dimensions, $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$.

This problem is in literature known as the linear quadratic control problem. Here, $x(t)$ is usually called the state of the system and $u(t)$ the (vector of) control(s).

¹This after J.F. Nash who proved in a number of papers from 1950-1953 the existence of such equilibria.

In economic literature one often considers instead of (1) a cost function where future cost are discounted. That is, the minimization of

$$\tilde{J} = \int_0^{t_f} e^{-rt} \{ \tilde{x}^T(t) Q \tilde{x}(t) + \tilde{u}^T(t) R \tilde{u}(t) \} dt + e^{-rt_f} \tilde{x}^T(t_f) Q_{t_f} \tilde{x}(t_f), \quad (3)$$

subject to

$$\dot{\tilde{x}}(t) = A \tilde{x}(t) + B \tilde{u}(t), \quad \tilde{x}(0) = x_0. \quad (4)$$

Introducing $x(t) := e^{-\frac{1}{2}rt} \tilde{x}(t)$ and $u(t) := e^{-\frac{1}{2}rt} \tilde{u}(t)$, it is easily verified that this optimization problem is equivalent with the minimization of (1), w.r.t. $u(\cdot)$, subject to

$$\dot{x}(t) = (A - \frac{1}{2}rI)x(t) + Bu(t), \quad x(0) = x_0.$$

So, to solve this discounted problem, all one has to do is to replace in the formulae below matrix A everywhere by matrix $A - \frac{1}{2}rI$. For that reason, we will just present the results for the undiscounted problem. The same remark applies for the games considered in the next sections.

The solution of the linear quadratic control problem is tightly connected to the existence of a symmetric solution of the following matrix Riccati differential equation

$$\dot{K}(t) = -A^T K(t) - K(t)A + K(t)SK(t) - Q, \quad K(t_f) = Q_{t_f}, \quad (RDE) \quad (5)$$

where $S := BR^{-1}B^T$.

Theorem 2.1 *The linear quadratic control problem (1,2) has for every initial state x_0 a solution if and only if the Riccati differential equation (5) has a symmetric solution $K(\cdot)$ on $[0, t_f]$.*

If the linear quadratic control problem has a solution, then it is unique and the optimal control in feedback form is

$$u^*(t) = -R^{-1}B^T K(t)x(t)$$

whereas in open-loop form it is

$$u^*(t) = -R^{-1}B^T K(t)\Phi(t, 0)x_0$$

with Φ being the transition matrix of

$$\dot{x}^*(t) = (A - SK(t))x^*(t).$$

Moreover, $J(u^*) = x_0^T K(0)x_0$. □

An important property of this Riccati differential equation (RDE) is that its solution can be found by solving a set of linear differential equations (see [30, Chapter 2.2]). This is in particular important from a computational point of view. For there are many efficient numerical algorithms that can calculate solutions of linear differential equations accurately. These algorithms have been implemented, e.g., in the computer software MATLAB to calculate solutions of Riccati differential equations.

To show this equivalence, consider the following, more general, non-symmetric matrix Riccati differential equation

$$\dot{K}(t) = -DK(t) - K(t)A + K(t)\tilde{S}K(t) - \tilde{Q}, \quad (6)$$

where $K, \tilde{Q} \in \mathbb{R}^{m \times n}$, $D \in \mathbb{R}^{m \times m}$, $A \in \mathbb{R}^{n \times n}$ and $\tilde{S} \in \mathbb{R}^{n \times m}$.

The solution of this Riccati differential equation (6) is intimately connected with the next set of linear differential equations

$$\begin{bmatrix} \dot{U}(t) \\ \dot{V}(t) \end{bmatrix} = \begin{bmatrix} A + \mu I & -\tilde{S} \\ -\tilde{Q} & -D + \mu I \end{bmatrix} \begin{bmatrix} U(t) \\ V(t) \end{bmatrix}, \quad (7)$$

where μ is a scalar parameter which may be chosen arbitrarily (and in particular equal to zero). This parameter may be used in computations to increase the numerical stability of calculating the solution of (7).

The relationship between (6) and (7) is summarized in the following theorem. A proof can be given along the lines of e.g. [9, Theorem 5.12].

Theorem 2.2 *If U, V is a solution pair of (7) with U nonsingular on the interval $[0, t_f]$, then $K(t) = VU^{-1}$ is a solution of the Riccati differential equation (6) on $[0, t_f]$. Conversely, if $K(t)$ is a solution of (6) on $[0, t_f]$ and $U(\cdot)$ is a fundamental solution of*

$$\dot{U}(t) = (A - \tilde{S}K(t) + \mu I)U(t)$$

then the pair $U(t), V(t) := K(t)U(t)$ is a solution of (7) on $[0, t_f]$. □

By considering in the above theorem the special case $D = A^T$ we get then the next result.

Corollary 2.3 *The Riccati differential equation (5) has a solution on $[0, t_f]$ if and only if the set of linear differential equations*

$$\begin{bmatrix} \dot{U}(t) \\ \dot{V}(t) \end{bmatrix} = \begin{bmatrix} A + \mu I & -S \\ -Q & -A^T + \mu I \end{bmatrix} \begin{bmatrix} U(t) \\ V(t) \end{bmatrix}; \quad \begin{bmatrix} U(t_f) \\ V(t_f) \end{bmatrix} = \begin{bmatrix} I \\ Q_{t_f} \end{bmatrix} \quad (8)$$

has a solution on $[0, t_f]$, with $U(\cdot)$ nonsingular.

Moreover, if (8) has an appropriate solution $(U(\cdot), V(\cdot))$, the solution of (5) is

$$K(t) = V(t)U^{-1}(t). \quad \square$$

Notice that with $H := \begin{bmatrix} A + \mu I & -S \\ -Q & -A^T + \mu I \end{bmatrix}$, the solution of the above differential equation (8) is

$$\begin{bmatrix} U(t) \\ V(t) \end{bmatrix} = e^{H(t_f-t)} \begin{bmatrix} I \\ Q_{t_f} \end{bmatrix}.$$

Consequently, if one can find an analytic expression for the Jordan canonical form of matrix H , it is possible to determine an analytic expression for U and V . From this one gets then an analytic solution for the Riccati differential equation (5) by calculating $K(t) := V(t)U^{-1}(t)$. Moreover, by choosing μ as a large negative number matrix H becomes stable. This property can be used to increase the numerical stability of the calculation of e^{Ht} .

Next, we lift the restriction that the final time t_f in the planning horizon must be finite. We consider the problem of finding a control function $u(\cdot) = Fx(\cdot)$ (where F is a time-invariant matrix) for each $x_0 \in \mathbb{R}^n$ that minimizes the cost functional

$$J(x_0, u) := \int_0^{\infty} \{x^T(t)Qx(t) + u^T(t)Ru(t)\}dt. \quad (9)$$

Notice that, since we do not make any definiteness assumptions w.r.t. matrix Q , the optimal state trajectory $x(t)$ of (2) (if it exists) may have the property that it will not converge. Since in economic applications this is usually a rare situation, we will consider the above minimization problem under the additional constraint that $\lim_{t \rightarrow \infty} x(t) = 0$. The imposed stabilization constraint is equivalent with the requirement that the system is stabilizable. Therefore, throughout this section, the assumption is made that the pair (A, B) is stabilizable (i.e. $\text{rank} [A - sI \ B] = n, \forall s \in \mathbb{C}^{+2}$). Furthermore, we introduce the set of linear, stabilizing, time-invariant feedback matrices, i.e.

$$\mathcal{F} := \{F \mid A + BF \text{ is stable}\}.$$

The next algebraic Riccati equation (ARE)

$$A^T X + XA - XSX + Q = 0 \quad (10)$$

plays an important role in the problem under consideration. A solution K of this equation will be called stabilizing if the matrix $A - SK$ is stable. It can be shown that (ARE) has at most one stabilizing solution. We have

Theorem 2.4 *Assume that (A, B) is stabilizable and $u = Fx$, with $F \in \mathcal{F}$. The linear quadratic control problem (2,9) has a minimum $\hat{F} \in \mathcal{F}$ for $J(x_0, u)$ for every $x_0 \in \mathbb{R}^n$ if and only if the algebraic Riccati equation (10) has a symmetric stabilizing solution K .*

If the linear quadratic control problem has a solution, then the solution is uniquely given by $\hat{F} = -R^{-1}B^TK$ and the optimal control in feedback form is

$$u^*(t) = -R^{-1}B^TKx(t).$$

In open-loop form it is

$$u^*(t) = -R^{-1}B^TK\Phi(t, 0)x_0$$

where Φ is the transition matrix of

$$\dot{x}^*(t) = (A - SK)x^*(t).$$

Moreover, $J(u^) = x_0^TKx_0$.* □

To calculate the stabilizing solution of (ARE) one can consult e.g. MATLAB again. There have been suggested in literature many ways to find in a numerical reliable and efficient way this stabilizing

² C^+ is the set of all complex numbers which have a nonnegative real part; C^- is the set of all complex numbers with a negative real part.

solution (see e.g. the review paper by Laub [20] and Abou-Kandil et al. [2]). Here we will consider an approach which will also be used in the next section to compute equilibria of the open-loop game.

For that purpose we consider the more general (nonsymmetric) algebraic Riccati equation

$$DK + KA - K\tilde{S}K + \tilde{Q} = 0, \quad (11)$$

where the dimensions correspond with those of the corresponding matrices in (6). Similarly as in (10) we will call a solution X of the nonsymmetric algebraic Riccati equation (11) stabilizing if all eigenvalues of matrix $A - \tilde{S}X$ have a negative real part.

Equation (11) can be rewritten as

$$[I_{2n} \ K] \begin{bmatrix} \tilde{Q} & D \\ A & -\tilde{S} \end{bmatrix} \begin{bmatrix} I_n \\ K \end{bmatrix} = 0.$$

From this we infer that the image of matrix $[I_{2n} \ K]$ is orthogonal to the image of $\begin{bmatrix} \tilde{Q} & D \\ A & -\tilde{S} \end{bmatrix} \begin{bmatrix} I_n \\ K \end{bmatrix}$.

Or, stated differently, the image of $\begin{bmatrix} \tilde{Q} & D \\ A & -\tilde{S} \end{bmatrix} \begin{bmatrix} I_n \\ K \end{bmatrix}$ belongs to the orthogonal complement of the image of matrix $[I_{2n} \ K]$. It is easily verified that the orthogonal complement of the image of matrix $[I_{2n} \ K]$ is given by the image of $\begin{bmatrix} -K \\ I_n \end{bmatrix}$. Therefore, ARE has a solution if and only if there exists a matrix $\Lambda \in \mathbb{R}^{n \times n}$ such that

$$\begin{bmatrix} \tilde{Q} & D \\ A & -\tilde{S} \end{bmatrix} \begin{bmatrix} I_n \\ K \end{bmatrix} = \begin{bmatrix} -K \\ I_n \end{bmatrix} \Lambda.$$

Premultiplication of both sides from the above equality with the matrix $\begin{bmatrix} 0 & I_n \\ -I_{2n} & 0 \end{bmatrix}$ yields then that (11) has a solution X if and only if there exists a matrix $\Lambda \in \mathbb{R}^{n \times n}$ such that

$$\begin{bmatrix} A & -\tilde{S} \\ -\tilde{Q} & -D \end{bmatrix} \begin{bmatrix} I_n \\ X \end{bmatrix} = \begin{bmatrix} I_n \\ X \end{bmatrix} \Lambda.$$

So, all solutions of (11) can be obtained by considering all n -dimensional invariant subspaces $V = \text{Im} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ of

$$M := \begin{bmatrix} A & -\tilde{S} \\ -\tilde{Q} & -D \end{bmatrix}, \quad (12)$$

with $X_1 \in \mathbb{R}^{n \times n}$ invertible. A subspace V that satisfies this property is called a graph subspace (since it can be "visualized" as the graph of the map: $x \rightarrow X_2 X_1^{-1} x$). The corresponding solution of (11) is then $X := X_2 X_1^{-1}$. To calculate the stabilizing solution of (10) one might pursue then as follows.

Algorithm 2.5

Step 1: Calculate matrix $H := \begin{bmatrix} A & -S \\ -Q & -A^T \end{bmatrix}$. Next calculate the spectrum of H . If the number of eigenvalues with a negative real part (counted with algebraic multiplicities) is less than n , goto Step 4 (there is no stabilizing solution in this case).

Step 2: Calculate the n -dimensional H -invariant subspace \mathcal{P} corresponding with the set of eigenvalues of H that have a negative real part. If \mathcal{P} is not a graph subspace, goto Step 4. There is no solution again.

Step 3: Calculate $n \times n$ matrices X and Y such that $\text{Im} \begin{bmatrix} X \\ Y \end{bmatrix} = \mathcal{P}$.

Denote $K := YX^{-1}$. Then

$$u^*(t) := -R^{-1}B^TKe^{A_{cl}t}x_0$$

solves the infinite planning horizon problem. Here $A_{cl} := A - SK$. The spectrum of the corresponding closed-loop matrix A_{cl} equals $\sigma(H|_{\mathcal{P}})$ ³. The involved cost for player i is $x_0^TKx_0$.

Step 4: End of algorithm. □

3 The open-loop game

In this section we consider two players who try to minimize their individual quadratic performance criterion. Each player controls a different set of inputs u_i to the single system described by the differential equation

$$\dot{x}(t) = Ax(t) + B_1u_1(t) + B_2u_2(t), \quad x(0) = x_0. \quad (13)$$

Here x is the n -dimensional state of the system, u_i is an m_i -dimensional (control) vector player i , $i = 1, 2$, can manipulate and x_0 is the initial state of the system.

The players have an open-loop information structure. The set of allowed actions is

$$\Gamma_i = \{u_i(\cdot) \mid u_i(\cdot) = f_i(t, x_0) \text{ and } u_i \text{ is piecewise continuous}\}, \quad i = 1, 2.$$

We are looking for Nash equilibria of this game. That is, for combinations of actions of all players which are secure against any attempt by one player to unilaterally alter his strategy. Or, stated differently, for those set of actions which are such that if one player deviates from his action he will only lose. In the literature on dynamic games this problem is known as the open-loop Nash non-zero-sum linear quadratic differential game and has been analyzed by several authors (see e.g. Starr et al. [33], Simaan et al. [32], Başar et al [5], Abou-Kandil et al. [1], Feucht [12], Kremer [15] and Engwerda [9]). To avoid cumbersome notation, we will restrict the analyses to the two-player case.

We will state first some basic results in case the performance criterion player i , $i = 1, 2$, aims to minimize w.r.t. u_i is:

$$J_1(u_1, u_2) := \int_0^{t_f} \{x^T(t)Q_1x(t) + u_1^T(t)R_{11}u_1(t) + u_2^T(t)R_{12}u_2(t)\}dt + x^{t_f}(t_f)Q_{1t_f}x(t_f), \quad (14)$$

³ $\sigma(H)$ denotes the spectrum (set of eigenvalues) of matrix H .

and

$$J_2(u_1, u_2) := \int_0^{t_f} \{x^T(t)Q_2x(t) + u_1^T(t)R_{21}u_1(t) + u_2^T(t)R_{22}u_2(t)\}dt + x^T(t_f)Q_{2t_f}x(t_f). \quad (15)$$

Again, here all matrices are symmetric and, moreover, R_{ii} are positive definite. So, we are looking for actions that satisfy simultaneously

$$J_1(u_1^*, u_2^*) \leq J_1(u_1, u_2^*) \text{ and } J_2(u_1^*, u_2^*) \leq J_2(u_1^*, u_2),$$

for all admissible (u_1, u_2) .

Using the shorthand notation $S_i := B_i R_{ii}^{-1} B_i^T$ we have the following.

Theorem 3.1 *Assume that*

1. *the set of coupled Riccati differential equations*

$$\dot{P}_1 = -A^T P_1 - P_1 A - Q_1 + P_1 S_1 P_1 + P_1 S_2 P_2; \quad P_1(t_f) = Q_{1t_f}, \quad (16)$$

$$\dot{P}_2 = -A^T P_2 - P_2 A - Q_2 + P_2 S_2 P_2 + P_2 S_1 P_1; \quad P_2(t_f) = Q_{2t_f} \quad (17)$$

has a solution P_i , $i = 1, 2$, on $[0, t_f]$, and

2. *the two Riccati differential equations,*

$$\dot{K}_i(t) = -A^T K_i(t) - K_i(t)A + K_i(t)S_i K_i(t) - Q_i(t), \quad K_i(t_f) = Q_{it_f}, \quad i = 1, 2, \quad (18)$$

have a symmetric solution $K_i(\cdot)$ on $[0, t_f]$.

Then the linear quadratic differential game (13-15) has a unique open-loop Nash equilibrium for every initial state.

Moreover, the set of equilibrium actions is given by:

$$u_i^*(t) = -R_{ii}^{-1} B_i^T P_i(t) \Phi(t, 0) x_0, \quad i = 1, 2.$$

Here $\Phi(t, 0)$ satisfies the transition equation

$$\dot{\Phi}(t, 0) = (A - S_1 P_1 - S_2 P_2) \Phi(t, 0); \quad \Phi(t, t) = I_n. \quad \square$$

Notice that the above equilibrium actions can also be obtained as a state feedback control. That is, $u_i^*(t)$ can be rewritten as:

$$u_i^*(t) = -R_{ii}^{-1} B_i^T P_i(t) x^*(t), \quad i = 1, 2,$$

where $x^*(t)$ satisfies:

$$\dot{x}^*(t) = (A - S_1 P_1 - S_2 P_2) x^*(t); \quad x^*(0) = x_0.$$

Assumption 2 in Theorem 3.1 states that for both players a linear quadratic control problem associated with this game problem should be solvable on $[0, t_f]$. That is, the optimal control problem that arises if the action of his opponent is known must be solvable for each player.

Theorem 3.1 presents a local result. That is, it just states a sufficient condition for existence of an equilibrium strategy at some fixed endpoint t_f in time. However, it can be shown that this condition is almost necessary too. For completeness we state this result here too.

Theorem 3.2 *Assume that the two Riccati differential equations (18) have a solution on $[0, t_1]$. Then, for all $t_f \in [0, t_1]$ the game defined on the interval $[0, t_f]$ has an open-loop Nash equilibrium for all x_0 if and only if the set of Riccati differential equations (16,17) has a solution on the interval $[0, t_1]$.*

Moreover, in case the above condition is satisfied the equilibrium is unique. □

With

$$P := \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}; \quad D := \begin{bmatrix} A^T & 0 \\ 0 & A^T \end{bmatrix}; \quad S := [S_1 \ S_2]; \quad \text{and } Q := \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix},$$

the set of coupled Riccati equations (16,17) can be rewritten as the non-symmetric matrix Riccati differential equation

$$\dot{P} = -DP - PA + PSP - Q; \quad P^T(t_f) = [Q_{1t_f}, \ Q_{2t_f}].$$

From section 2 we know that numerically the solution of such a Riccati differential equation can be obtained by solving a set of linear differential equations. From Theorem 2.2 we get the next Corollary.

Corollary 3.3 *The set of coupled Riccati differential equations (16,17) has a solution on $[0, t_f]$ if and only if the set of linear differential equations*

$$\begin{bmatrix} \dot{U}(t) \\ \dot{V}_1(t) \\ \dot{V}_2(t) \end{bmatrix} = M \begin{bmatrix} U(t) \\ V_1(t) \\ V_2(t) \end{bmatrix}; \quad \begin{bmatrix} U(t_f) \\ V_1(t_f) \\ V_2(t_f) \end{bmatrix} = \begin{bmatrix} I \\ Q_{1t_f} \\ Q_{2t_f} \end{bmatrix} \quad (19)$$

has a solution on $[0, t_f]$, with $U(\cdot)$ nonsingular.

Moreover, if (19) has an appropriate solution $(U(\cdot), V_1(\cdot), V_2(\cdot))$, the solution of (16,17) is obtained as $P_i(t) := V_i(t)U^{-1}(t)$, $i = 1, 2$. □

Next we consider the infinite planning horizon case. We assume now that the performance criterion player $i = 1, 2$, likes to minimize is:

$$\lim_{t_f \rightarrow \infty} J_i(x_0, u_1, u_2, t_f) \quad (20)$$

where

$$J_1(u_1, u_2) := \int_0^{t_f} \{x^T(t)Q_1x(t) + u_1^T(t)R_{11}u_1(t) + u_2^T(t)R_{12}u_2(t)\}dt, \quad (21)$$

and

$$J_2(u_1, u_2) := \int_0^{t_f} \{x^T(t)Q_2x(t) + u_1^T(t)R_{21}u_1(t) + u_2^T(t)R_{22}u_2(t)\}dt. \quad (22)$$

We make the same assumptions as before w.r.t. the above involved matrices.

The matrix pairs (A, B_i) , $i = 1, 2$, are assumed to be stabilizable. So, in principle, each player is capable to stabilize the system on his own.

The information both players have at the beginning of the game is similar to the finite-planning horizon case. Each player only knows the initial state of the system. The admissible control actions

are now functions of time, where time runs from zero to infinity. Since we only like to consider those outcomes of the game that yield a finite cost to both players and the players are assumed to have a common interest in stabilizing the system, we restrict ourselves to functions belonging to the set

$$\Gamma(x_0) = \left\{ u \in L_2 \mid J_i(x_0, u) \text{ exists in } \mathbb{R} \cup \{-\infty, \infty\}, \lim_{t \rightarrow \infty} x(t) = 0 \right\}.$$

Notice that $\Gamma(x_0)$ depends on the initial state of the system. For simplicity of notation we omit, however, this dependency. Moreover, the restriction to this set of control functions requires some form of communication between the players. That is, it is assumed that both players have the meta-objective to stabilize the system.

Before we present the basic theorem about existence of a unique equilibrium for this game, we reconsider the nonsymmetric algebraic Riccati equation (11). A solution X of (11) will be called strongly stabilizing if both $\sigma(A - \tilde{S}X) \subset \mathbb{C}^-$ and $\sigma(D^T - \tilde{S}^T X^T) \subset \mathbb{C}^-$. The next lemma shows how one can verify whether (11) has a strongly stabilizing solution.

Lemma 3.4 *(11) has a strongly stabilizing solution (P_1, P_2) if and only if matrix M in (12) has an n -dimensional stable graph subspace and M has $2n$ eigenvalues (counting algebraic multiplicities) in \mathbb{C}^+ . \square*

Theorem 3.5 *The linear quadratic differential game (13,20-22) has a unique open-loop Nash equilibrium for every initial state if and only if*

1. *The (set of coupled) algebraic Riccati equations*

$$\begin{bmatrix} A^T & 0 \\ 0 & A^T \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} + \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} A - \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \begin{bmatrix} S_1 & S_2 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} + \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = 0 \quad (23)$$

have a strongly stabilizing solution, and

2. *the two algebraic Riccati equations*

$$A^T K_i(t) + K_i(t)A - K_i(t)S_i K_i(t) + Q_i(t), \quad i = 1, 2, \quad (24)$$

have a symmetric stabilizing solution.

Moreover, the unique equilibrium actions are given by

$$u_i^*(t) = -R_i^{-1} B_i^T P_i \Phi(t, 0) x_0, \quad i = 1, 2.$$

Here $\Phi(t, 0)$ satisfies the transition equation

$$\dot{\Phi}(t, 0) = (A - S_1 P_1 - S_2 P_2) \Phi(t, 0) =: A_{cl} \Phi(t, 0); \quad \Phi(t, t) = I_n.$$

The costs, by using these actions, for the players are

$$x_0^T M_i x_0, \quad i = 1, 2,$$

where M_i is the unique solution of the Lyapunov equation

$$A_{cl}^T M_i + M_i A_{cl} + Q_i + P_i^T S_i P_i = 0. \quad \square$$

Note that in case the game has a unique open-loop Nash equilibrium, the corresponding Nash actions can also be implemented as a feedback control. That is,

$$u_i^*(t) = -R_i^{-1} B_i^T P_i x^*(t), \quad i = 1, 2, \quad \text{where } x^*(t) \text{ solves } \dot{x}(t) = (A - S_1 P_1 - S_2 P_2)x(t), \quad x(0) = x_0.$$

To calculate the equilibrium one can either use an algorithm based on the calculation of the eigenstructure of a certain matrix or try to solve the algebraic Riccati equations iteratively. We will start with the former approach.

Algorithm 3.6

Step 1: Calculate the eigenstructure of $H_i := \begin{bmatrix} A & -S_i \\ -Q_i & -A^T \end{bmatrix}$.

If H_i , $i = 1, 2$, has an n -dimensional stable graph subspace, then proceed. Otherwise there is no unique equilibrium and goto Step 5.

Step 2: Calculate matrix $M := \begin{bmatrix} A & -S_1 & -S_2 \\ -Q_1 & -A^T & 0 \\ -Q_2 & 0 & -A^T \end{bmatrix}$.

If M has n stable eigenvalues and $2n$ unstable eigenvalues (counting algebraic multiplicities) then proceed. Else there is not a unique equilibrium and goto Step 5.

Step 3: Calculate the M -invariant subspaces \mathcal{P} corresponding with the eigenvalues for which $\text{Re } \lambda < 0$. If this is a graph subspace then proceed. Else there exists no unique equilibrium and goto Step 5.

Step 4: Calculate three $n \times n$ matrices X , Y and Z such that $\text{Im} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathcal{P}$.

Denote $P_1 := YX^{-1}$ and $P_2 := ZX^{-1}$. Then

$$u_i^*(t) := -R_i^{-1} B_i^T P_i e^{A_{cl}t} x_0$$

is the open-loop Nash equilibrium strategy. Here $A_{cl} := A - S_1 P_1 - S_2 P_2$. The spectrum of the corresponding closed-loop matrix A_{cl} equals $\sigma(M|_{\mathcal{P}})$.

Step 5: End of algorithm. □

Remark 3.7 Step 1 in the above algorithm verifies whether the algebraic Riccati equations (24) have a stabilizing solution. Of course one can use here MATLAB to verify this. □

Concerning the numerical stability of Algorithm 3.6 we notice that various suggestions have been made in literature to calculate solutions of Riccati equations in a numerical reliable way (see e.g. Laub [19] and [20], Paige et al. [26], Van Dooren [8], Mehrmann [22] and Abou-Kandil et al. [2] for a more general survey on various types of Riccati equations). These methods also can be used to improve the numerical stability of Algorithm 3.6. Particularly if one considers the implementation of large scale models one should consult this literature.

As already indicated one can also try to solve the (set of coupled) algebraic Riccati equations (23) iteratively. Particularly for large scale systems one might hope that such algorithms will be

more efficient. For that reason various iteration schemes have been suggested in literature. However, since in general (23) admits several solutions, convergence of any algorithm is quite difficult to obtain under general conditions (see e.g. [3]). An important problem with these algorithms is, in case one does not want to verify the strong stabilizability of the system beforehand, how one should proceed in case the algorithm terminates at a non-stabilizing solution.

We will present here an iterative scheme based on the Newton-Raphson method (see e.g. [17]).

To motivate this scheme reconsider (11) and let

$$F(K) := DK + KA - K\tilde{S}K + \tilde{Q}.$$

Suppose that at the k^{th} iteration a solution P_k has been obtained, which is close to the solution X satisfying $F(X) = 0$. Now, write

$$X = P_k + \Delta_k.$$

If Δ_k is small we can approximate $F(X)$ by omitting quadratic terms in Δ_k and we obtain

$$F(X) \approx D(P_k + \Delta_k) + (P_k + \Delta_k)A - P_k\tilde{S}P_k - \Delta_k\tilde{S}P_k - P_k\tilde{S}\Delta_k + \tilde{Q}. \quad (25)$$

The basic idea of the Newton-Raphson method is to estimate Δ_k by setting the right-hand side of (25) to zero. Denoting $P_k + \Delta_k =: P_{k+1}$ and substitution of Δ_k by $P_{k+1} - P_k$ yields then from (25) the next iterative scheme for P_{k+1} :

$$(D - P_k\tilde{S})P_{k+1} + P_{k+1}(A - \tilde{S}P_k) + P_k\tilde{S}P_k + \tilde{Q} = 0. \quad (26)$$

Notice that (26) is a linear matrix (Sylvester) equation in P_{k+1} . This equation can be efficiently solved by using e.g. MATLAB (using the function `lyap`). Starting the initialization at a strongly stabilizing solution P_0 one may hope then that all P_k will be strongly stabilizing and that the P_k converge. This leads to the following algorithm.

Algorithm 3.8

- Step 1:* Verify whether the algebraic Riccati equations (24) have a stabilizing solution (see e.g. Step 1 of Algorithm 3.6).
- Step 2:* Calculate a P_0 such that both $D - P_0\tilde{S}$ and $A - \tilde{S}P_0$ are stable. Choose the desired approximation error ϵ .
- Step 3:* Solve P_{k+1} from (26).
- Step 4:* Calculate $eps := \|P_{k+1} - P_k\|$. If $eps > \epsilon$, increase k by one and return to Step 3.
- Step 5:* Verify whether both $D - P_k\tilde{S}$ and $A - \tilde{S}P_k$ are stable. If P_k is a strongly stabilizing solution, goto Step 8. The problem has a solution and P_k can be used in Theorem 3.5 to determine the equilibrium actions.
- Step 6:* If just $A - \tilde{S}P_k$ is stable, goto Step 8. The problem has not a unique equilibrium.
- Step 7:* If both $D - P_k\tilde{S}$ and $A - \tilde{S}P_k$ are unstable, goto Step 2 (after maybe now first a verification of whether the problem has a strongly stabilizing solution (see Lemma 1)).
- Step 8:* End of algorithm. □

4 The feedback game

As argued in the introduction we now look for actions which are functions of the current state of the system and time or, so-called, Markov functions.

Since the system we consider is linear, it is often argued that the equilibrium actions should be also a linear function of the state. We will adopt this point of view here. We restrict the set of actions the players may choose their actions from, in case the planning horizon t_f is finite, to

$$\Gamma_i^{lfb} := \{u_i(\cdot) \mid u_i(t) = F_i(t)x(t) \text{ where } F_i(\cdot) \text{ is a piecewise continuous function on } [0, t_f], i = 1, 2\}.$$

So, in this section we say that a set of actions $u_i^*(t) = F_i^*(t)x(t)$ (or with some abuse of notation (F_1, F_2)) constitute a feedback Nash equilibrium if

$$J_1(u_1^*, u_2^*) \leq J_1(u_1, u_2^*) \text{ and } J_2(u_1^*, u_2^*) \leq J_2(u_1^*, u_2), \text{ for all } u_i \in \Gamma_i^{lfb}.$$

Since by definition the equilibrium actions are a function of the current state of the system, they can be interpreted as policy rules (see Reinganum et al. [31]). They require no precommitment of the players, and hence are also applicable if players are not "credible".

Similar as for open-loop Nash equilibria, it turns out that linear feedback Nash equilibria can be explicitly determined by solving a set of coupled Riccati equations.

Theorem 4.1 *The linear quadratic differential game (13-15) has for every initial state a linear feedback Nash equilibrium if and only if the next set of coupled Riccati differential equations has a set of symmetric solutions K_1, K_2 on $[0, t_f]$*

$$\begin{aligned} \dot{K}_1(t) &= -(A - S_2 K_2(t))^T K_1(t) - K_1(t)(A - S_2 K_2(t)) + K_1(t) S_1 K_1(t) - Q_1 - K_2(t) S_{21} K_2(t), \\ K_1(t_f) &= Q_{1t_f}; \end{aligned} \tag{27}$$

$$\begin{aligned} \dot{K}_2(t) &= -(A - S_1 K_1(t))^T K_2(t) - K_2(t)(A - S_1 K_1(t)) + K_2(t) S_2 K_2(t) - Q_2 - K_1(t) S_{12} K_1(t), \\ K_2(t_f) &= Q_{2t_f}. \end{aligned} \tag{28}$$

Moreover, in that case there is a unique equilibrium. The equilibrium actions are

$$u_i^*(t) = -R_i^{-1} B_i^T K_i(t)x(t), \quad i = 1, 2.$$

The cost incurred by player i is $x_0^T K_i(0)x_0$, $i = 1, 2$. □

To calculate the solutions of the endpoint boundary-value differential equations (27,28), one usually first rewrites this set of equations as an initial boundary-value problem by reversing the time axis. That is introduce $\tilde{K}_i(t) := K_i(t_f - t)$ and $s := t_f - t$. Then (27,28) have a solution on $[0, t_f]$ if and only if the next set of nonlinear differential equations has a solution:

$$\begin{aligned} \dot{\tilde{K}}_1(s) &= (A - S_2 \tilde{K}_2(s))^T \tilde{K}_1(s) + \tilde{K}_1(s)(A - S_2 \tilde{K}_2(s)) - \tilde{K}_1(s) S_1 \tilde{K}_1(s) + Q_1 + \tilde{K}_2(s) S_{21} \tilde{K}_2(s), \\ \tilde{K}_1(0) &= Q_{1t_f}; \end{aligned} \tag{29}$$

$$\begin{aligned} \dot{\tilde{K}}_2(s) &= (A - S_1 \tilde{K}_1(s))^T \tilde{K}_2(s) + \tilde{K}_2(s)(A - S_1 \tilde{K}_1(s)) - \tilde{K}_2(s) S_2 \tilde{K}_2(s) + Q_2 + \tilde{K}_1(s) S_{12} \tilde{K}_1(s), \\ \tilde{K}_2(0) &= Q_{2t_f}. \end{aligned} \tag{30}$$

(29,30) can be rewritten now as a set of nonlinear differential equations. There are many numerical tools to solve a set of nonlinear differential equations. A number of them have been implemented in MATLAB too. Using e.g. the function `ode23` within MATLAB the above differential equations can be solved in a numerical reliable way.

Next we proceed with the infinite planning horizon case. That is, the minimization of (20-22) subject to (13). We consider here Nash equilibria within the class of linear time-invariant state feedback policy rules. That is, we shall restrain our set of permitted controls to the constant linear feedback strategies. That is, to $u_i = F_i x$, with $F_i \in \mathbb{R}^{m_i \times n}$, $i = 1, 2$, and where (F_1, F_2) belongs to the set

$$\mathcal{F} := \{F = (F_1, F_2) \mid A + B_1 F_1 + B_2 F_2 \text{ is stable}\}.$$

The stabilization constraint is imposed to ensure the finiteness of the infinite-horizon cost integrals that we will consider.

To make sure that our problem setting makes sense, we assume throughout this chapter that the set \mathcal{F} is non-empty. A necessary and sufficient condition for this to hold is that the matrix pair $(A, [B_1, B_2])$ is stabilizable.

Summarizing, $(F_1^*, F_2^*) \in \mathcal{F}$ is called a feedback Nash equilibrium if the following inequalities hold:

$$J_1(x_0, F_1^*, F_2^*) \leq J_1(x_0, F_1, F_2^*) \text{ and } J_2(x_0, F_1^*, F_2^*) \leq J_2(x_0, F_1^*, F_2)$$

for each x_0 and for each state feedback matrix F_i , $i = 1, 2$ such that (F_1^*, F_2) and $(F_1, F_2^*) \in \mathcal{F}$.

Next, consider the set of coupled algebraic Riccati equations

$$0 = -(A - S_2 K_2)^T K_1 - K_1 (A - S_2 K_2) + K_1 S_1 K_1 - Q_1 - K_2 S_{21} K_2, \quad (31)$$

$$0 = -(A - S_1 K_1)^T K_2 - K_2 (A - S_1 K_1) + K_2 S_2 K_2 - Q_2 - K_1 S_{12} K_1. \quad (32)$$

Theorem 4.2 below states that feedback Nash equilibria are completely characterized by stabilizing solutions of (31,32). That is, by solutions (K_1, K_2) for which the closed-loop system matrix $A - S_1 K_1 - S_2 K_2$ is stable.

Theorem 4.2 *Let (K_1, K_2) be a stabilizing solution of (31,32) and define $F_i^* := -R_{ii}^{-1} B_i^T K_i$ for $i = 1, 2$. Then (F_1^*, F_2^*) is a feedback Nash equilibrium. Moreover, the cost incurred by player i by playing this equilibrium action is $x_0^T K_i x_0$, $i = 1, 2$.*

Conversely, if (F_1^, F_2^*) is a feedback Nash equilibrium, there exists a stabilizing solution (K_1, K_2) of (31,32) such that $F_i^* = -R_{ii}^{-1} B_i^T K_i$. \square*

The above theorem shows that all infinite-planning horizon feedback Nash equilibria can be found by solving a set of coupled algebraic Riccati equations. To find all stabilizing solutions of the system (31,32) is in general a difficult problem.

To get some intuition for the solution set we next consider the scalar two-player game, where players are not interested in the control actions pursued by the other player. This case was extensively studied in [9, Chapter 8.4]. In particular it was shown that in this game never more than three equilibria occur. Furthermore a complete characterization of parameters which give rise to either 0, 1, 2, or 3 equilibria was provided. Also a numerical algorithm, based on the calculation of invariant

subspaces for a certain matrix, was given to calculate all equilibria. Below we will summarize this algorithm. Notice that this algorithm can be extended (in a non-obvious way, see [9, Chapter 8.5.3]) to the N -player scalar case.

So, for the moment we consider the next game.

$$J_i(x_0, u_1, u_2) = \int_0^\infty \{q_i x^2(t) + r_i u_i^2\} dt, \quad i = 1, 2, \quad (33)$$

subject to the dynamical system

$$\dot{x}(t) = ax(t) + b_1 u_1(t) + b_2 u_2(t), \quad x(0) = x_0. \quad (34)$$

The algebraic Riccati equations which provide the key to finding the feedback Nash equilibria for this game (see Theorem 4.2) are obtained from (31,32) by substitution of $R_{21} = R_{12} = 0$, $A = a$, $B_i = b_i$, $Q_i = q_i$, $R_{ii} = r_i$ and $s_i := b_i^2/r_i$, $i = 1, 2$, into these equations. By Theorem 4.2 then a pair of control actions $f_i^* := -\frac{b_i}{r_i} k_i$, $i = 1, 2$, constitute a feedback Nash equilibrium if and only if the next equations have a solution $x_i = k_i$, $i = 1, 2$:

$$s_1 x_1^2 + 2s_2 x_1 x_2 - 2ax_1 - q_1 = 0 \quad (35)$$

$$s_2 x_2^2 + 2s_1 x_1 x_2 - 2ax_2 - q_2 = 0 \quad (36)$$

$$a - s_1 x_1 - s_2 x_2 < 0. \quad (37)$$

Geometrically, the equations (35) and (36) represent two hyperbolas in the (x_1, x_2) plane, whereas the inequality (37) divides this plane into a “stable” and an “anti-stable” region. So, all feedback Nash equilibria are obtained as the intersection points of both hyperbolas in the “stable” region.

To get a better intuition for the numerical algorithm that looks for all solutions satisfying (35-37) we recall the next theorem from [9].

Theorem 4.3

1. Assume that (k_1, k_2) is a feedback Nash equilibrium strategy. Then the negative of the corresponding closed-loop system parameter $\lambda := -a + \sum_{i=1}^2 s_i k_i > 0$ is an eigenvalue of the matrix

$$M := \begin{bmatrix} -a & s_1 & s_2 & 0 \\ q_1 & a & 0 & -s_2 \\ q_2 & 0 & a & -s_1 \\ 0 & \frac{1}{3}q_2 & \frac{1}{3}q_1 & \frac{1}{3}a \end{bmatrix}. \quad (38)$$

Furthermore, $[1, k_1, k_2, k_1 k_2]^T$ is a corresponding eigenvector and $\lambda^2 \geq \sigma_{max}$.

2. Assume that $[1, k_1, k_2, k_3]^T$ is an eigenvector corresponding to a positive eigenvalue λ of M , satisfying $\lambda^2 \geq \sigma_{max}$, and that the eigenspace corresponding with λ has dimension one. Then, (k_1, k_2) is a feedback Nash equilibrium. \square

From theorem 4.3 then the next numerical algorithm can be derived to calculate all feedback Nash equilibria.

Algorithm 4.4 The following algorithm calculates all feedback Nash equilibria of the linear quadratic differential game (33,34).

Step 1: Calculate matrix M in (38) and $\sigma := \max_i \frac{b_i^2 q_i}{r_i}$.

Step 2: Calculate the eigenstructure (λ_i, m_i) , $i = 1, \dots, k$, of M , where λ_i are the eigenvalues and m_i the corresponding algebraic multiplicities.

Step 3: For $i = 1, \dots, k$ repeat the following steps:

3.1) If i) $\lambda_i \in \mathbb{R}$; ii) $\lambda_i > 0$ and iii) $\lambda_i^2 \geq \sigma$ then proceed with Step 3.2 of the algorithm. Otherwise, return to Step 3.

3.2) If $m_i = 1$ then

3.2.1) calculate an eigenvector v corresponding with λ_i of M . Denote the entries of v by $[v_0, v_1, v_2, \dots]^T$. Calculate $k_j := \frac{v_j}{v_0}$ and $f_j := -\frac{b_j k_j}{r_j}$. Then, (f_1, \dots, f_N) is a feedback Nash equilibrium and $J_j = k_j x_0^2$, $j = 1, \dots, N$. Return to Step 3.

If $m_i > 1$ then

3.2.2) Calculate $\sigma_i := \frac{b_i^2 q_i}{r_i}$.

3.2.3) For all 2^N sequences (t_1, \dots, t_N) , $t_k \in \{-1, 1\}$,

i) calculate

$$y_j := \lambda_i + t_j \sqrt{\lambda_i^2 - \sigma_j}, \quad j = 1, \dots, N$$

ii) If $\lambda_i = -a + \sum_{j=1, \dots, N} y_j$ then calculate $k_j := \frac{y_j r_j}{b_j^2}$ and $f_j := -\frac{b_j k_j}{r_j}$. Then, (f_1, \dots, f_N) is a feedback Nash equilibrium and $J_j = k_j x_0^2$, $j = 1, \dots, N$.

Step 4: End of the algorithm. □

Generically, one may expect that the eigenvalues of matrix M in the above algorithm will have an algebraic multiplicity of one. In that case it is quite clear how to proceed in the above algorithm. In case an eigenvalue has a multiplicity larger than one, however, one may be forced to proceed with Step 3.2.2 and 3.2.3 in the algorithm. Since this part of the algorithm is somewhat technical we illustrate this part of the algorithm with an example.

Example 4.5 Consider $A = 3$; $B_i = Q_i = 2$ and $R_i = 1$, $i = 1, 2$. To calculate the feedback Nash equilibria for this game, according Algorithm 4.4, we first calculate the eigenstructure of matrix

$$M := \begin{bmatrix} -3 & 4 & 4 & 0 \\ 2 & 3 & 0 & -4 \\ 2 & 0 & 3 & -4 \\ 0 & \frac{2}{3} & \frac{2}{3} & 1 \end{bmatrix}.$$

Using Matlab, we find the eigenvalues $\{-4.8297, 2.8297, 3, 3\}$. Since both the square of 2.8297 and 3 are larger than $\sigma := 8$, we have to process Step 3 of the algorithm for both these eigenvalues.

First, consider the eigenvalue 2.8297. A corresponding eigenvector is

$$[v_0, v_1, v_2, v_3] := [-0.6532, -0.4760, -0.4760, -0.3468].$$

So,

$$k_1 := \frac{v_1}{v_0} = 0.7287 \text{ and } k_2 := \frac{v_2}{v_0} = k_1.$$

This yields the symmetric feedback Nash equilibrium actions

$$u_i = -\frac{b_i k_i}{r_i} x(t) = -1.4574x(t).$$

The corresponding closed-loop system and cost are

$$\dot{x}(t) = -2.8297x(t), \quad x(0) = x_0; \text{ and } J_i = 0.7287x_0^2, \quad i = 1, 2,$$

respectively. Next consider the eigenvalue 3. This eigenvalue has an algebraic multiplicity 2. So we have to proceed with Step 3.2.3 of the algorithm to calculate the equilibria associated with this eigenvalue.

For the sequence $(t_1, t_2) := (1, -1)$ we obtain in Step 3.2.3.i) $y_1 = 4$ and $y_2 = 2$, respectively. These numbers satisfy the equality under Step 3.2.3.ii). Therefore, with

$$k_1 := \frac{y_1 r_1}{b_1^2} = 1 \text{ and } k_2 := \frac{y_2 r_2}{b_2^2} = \frac{1}{2},$$

the corresponding equilibrium actions are

$$u_1 = -\frac{b_1 k_1}{r_1} x(t) = -2x(t) \text{ and } u_2 = -\frac{b_2 k_2}{r_2} x(t) = -x(t).$$

The resulting closed-loop system and cost are in this case

$$\dot{x}(t) = -3x(t), \quad x(0) = x_0; \quad J_1 = x_0^2 \text{ and } J_2 = \frac{1}{2}x_0^2.$$

In a similar way we obtain for the sequence $(-1, 1)$ the "reversed" solution

$$k_1 = \frac{1}{2}; \quad k_2 = 1; \quad f_1 = -1; \text{ and } f_2 = -2,$$

respectively. Which gives rise to the closed-loop and cost

$$\dot{x}(t) = -3x(t), \quad x(0) = x_0; \quad J_1 = \frac{1}{2}x_0^2 \text{ and } J_2 = x_0^2.$$

Finally, it is easily verified that the numbers y_j implied by both the sequences $(1, 1)$ and $(-1, -1)$ do not satisfy the equality mentioned under Step 3.2.3.ii). So, the game has three feedback Nash equilibria.

Using Algorithm 3.6 one can verify easily that this game has neither a unique open-loop Nash equilibrium too. The with this game corresponding matrix M has two stable eigenvalues. \square

As already mentioned, one can proceed for the scalar N -player case in a similar way to find all equilibria. For the multivariable case there are unfortunately no computational algorithms available which provide all equilibrium points. Like for the open-loop information case some iterative schemes

have been proposed in literature to find an equilibrium (see e.g. Krikelis et al. [16], Tabak [35], Papavassilopoulos et al. [27] and Li et al. [21]). However, all of them just provide one equilibrium (if convergence occurs). Since the number of equilibria can vary between zero and infinity it is clear that, particularly when there is no additional information that a certain type of equilibrium point is preferred or the number of equilibria is unknown, one would like to have an overview of all possible equilibria.

Papavassilopoulos et al. considered in [28] a geometric approach for calculating the stabilizing solutions of the feedback Nash algebraic Riccati equations similar to the open-loop approach of section 3. In that approach subspaces have to be calculated which satisfy simultaneously some invariance properties. However, up to now, it is unknown how to find these subspaces.

Two other different methods that have been proposed in the past for finding all isolated solutions to a system of polynomials are interval methods (see e.g. Van Hentenryck et al. [13] for references) and continuation methods (see e.g. Morgan [23] and Verschelde et al. [36]). Continuation methods have been shown to be effective for problems for which the total degree of the constraints is not too high, since the number of paths explored depends on the estimation of the number of solutions. Interval methods are generally robust but used to be slow. The approach taken in Van Hentenryck et al. [13], however, seems to overcome this bottleneck and be rather efficient. One important point that has to be managed in using the interval method is, however, the choice of the initial interval that contains all equilibria. Moreover, it is unclear how this method will perform in case there are an infinite number of feedback Nash solutions. Numerical experience using the above methods to find feedback Nash equilibria is, unfortunately, still lacking at this moment.

We conclude this section with three iterative schemes that can be used to calculate equilibria.

Algorithm 4.6 The following algorithm calculates a feedback Nash equilibrium of the linear quadratic differential game (13,20-22) (see [21]).

Step 1: Determine the stabilizing solution K_1^0 of

$$A^T K_1^0 + K_1^0 A + Q_1 - K_1^0 S_1 K_1^0 = 0.$$

Next determine the stabilizing solution K_2^0 of the Riccati equation

$$(A - S_1 K_1^0)^T K_2^0 + K_2^0 (A - S_1 K_1^0) + Q_2 + K_1^0 S_{12} K_1^0 - K_2^0 S_2 K_2^0 = 0.$$

Step 2: Let $i := 0$. Repeat the next iterations until the matrices K_j^i , $j = 1, 2$, below have converged.

Here $A_{cl}^i := A - S_1 K_1^i - S_2 K_2^i$, and K_j^{i+1} , $j = 1, 2$, are the solutions of the Lyapunov equations

$$\begin{aligned} A_{cl}^{iT} K_1^{i+1} + K_1^{i+1} A_{cl}^i &= -(Q_1 + K_1^i S_1 K_1^i + K_2^i S_{21} K_2^i) \\ A_{cl}^{iT} K_2^{i+1} + K_2^{i+1} A_{cl}^i &= -(Q_2 + K_2^i S_2 K_2^i + K_1^i S_{12} K_1^i), \end{aligned}$$

respectively, for $i = 0, 1, \dots$

Step 3: End of the algorithm. □

In the above algorithm (4.6) the hope is that the matrices A_{cl}^i will be stable for all i . In case the stability condition is violated at some iteration, this might indicate that the iteration at hand will have no appropriate solution. The next iteration scheme we present here is motivated by the fact that its convergence speed is often faster than that of the previous algorithm.

Algorithm 4.7 The following algorithm calculates a feedback Nash equilibrium of the linear quadratic differential game (13,20-22).

Step 1: Determine (K_1^0, K_2^0) such that $A - S_1 K_1^0 - S_2 K_2^0$ is stable.

Step 2: Let $i := 0$. Repeat the next iterations until the matrices K_j^i , $j = 1, 2$, below have converged. Here K_j^{i+1} , $j = 1, 2$, are the stabilizing solutions of the algebraic Riccati equations

$$\begin{aligned} 0 &= (A - S_2 K_2^i)^T K_1^{i+1} + K_1^{i+1} (A - S_2 K_2^i) - K_1^{i+1} S_1 K_1^{i+1} + Q_1 + K_2^i S_{21} K_2^i, \\ 0 &= (A - S_1 K_1^{i+1})^T K_2^{i+1} + K_2^{i+1} (A - S_1 K_1^{i+1}) - K_2^{i+1} S_2 K_2^{i+1} + Q_2 + K_1^{i+1} S_{12} K_1^{i+1}. \end{aligned}$$

Step 3: End of the algorithm. □

The third iteration we present is based on the Newton-Raphson method. Following the lines of the derivation of Algorithm 3.8 one obtains the iteration scheme outlined in Algorithm 4.8. Notice that in Step 2 below one has to solve a set of linear matrix equations in K_1^{i+1} and K_2^{i+1} . To solve this set of equations one can, e.g., first rewrite them as one big set of linear equations $Gx = d$ (using Kronecker notation) and then use a numerical reliable computerpackage to solve this set of equations. Notice that the K_j^{i+1} are symmetric, which reduces the number of linear equations that has to be solved.

Algorithm 4.8 The following algorithm calculates a feedback Nash equilibrium of the linear quadratic differential game (13,20-22).

Step 1: Determine (K_1^0, K_2^0) such that $A - S_1 K_1^0 - S_2 K_2^0$ is stable.

Step 2: Let $i := 0$. Repeat the next iterations until the matrices K_j^i , $j = 1, 2$, below have converged. Here K_j^{i+1} , $j = 1, 2$, are the solutions of the linear matrix equations

$$\begin{aligned} 0 &= (A - S_1 K_1^i - S_2 K_2^i)^T K_1^{i+1} + K_1^{i+1} (A - S_1 K_1^i - S_2 K_2^i) + K_2^{i+1} (-S_2 K_1^i + S_{21} K_2^i) + \\ &\quad (-K_1^i S_2 + K_2^i S_{21}) K_2^{i+1} + K_2^i S_2 K_1^i + K_1^i S_2 K_2^i + K_1^i S_1 K_1^i + Q_1 - K_2^i S_{21} K_2^i, \\ 0 &= (A - S_1 K_1^i - S_2 K_2^i)^T K_2^{i+1} + K_2^{i+1} (A - S_1 K_1^i - S_2 K_2^i) + K_1^{i+1} (-S_1 K_2^i + S_{12} K_1^i) + \\ &\quad (-K_2^i S_1 + K_1^i S_{12}) K_1^{i+1} + K_1^i S_1 K_2^i + K_2^i S_1 K_1^i + K_2^i S_2 K_2^i + Q_2 - K_1^i S_{12} K_1^i. \end{aligned}$$

Step 3: End of the algorithm. □

5 An Example

Consider the interaction of fiscal stabilization policies of two countries. Assume that the competitiveness between both countries is described by the differential equation

$$\dot{s}(t) = -as(t) + f_1(t) - f_2(t), \quad s(0) = s_0,$$

where $a > 0$, the variable $s(t)$ denotes the difference in prices between both countries at time t and $f_i(t)$ is the fiscal deficit set by the fiscal authority in country i , $i = 1, 2$. Each fiscal authority seeks to minimize the following intertemporal loss function that is assumed to be quadratic in the price differential and fiscal deficits,

$$J_i = \int_0^\infty e^{-rt} \{s^2(t) + r_i f_i^2(t)\} dt, \quad i = 1, 2.$$

Here r is a discount factor. Assume that both countries do not cooperate their policies aimed at reducing the initial price differential between both countries.

Introducing $x(t) := e^{-\frac{1}{2}rt}s(t)$ and $u_i(t) := e^{-\frac{1}{2}rt}f_i(t)$ the problem is equivalent with the minimization of

$$J_i = \int_0^\infty \{x^2(t) + r_i u_i^2(t)\} dt, \quad i = 1, 2,$$

subject to

$$\dot{x}(t) = \left(-a - \frac{1}{2}r\right)x(t) + u_1(t) - u_2(t), \quad x(0) = s_0.$$

Assume $a = 0.975$, $r = 0.05$, $r_1 = 1$ and $r_2 = 2$. Then $s_1 = 1$ and $s_2 = 0.5$.

To calculate the open-loop Nash equilibrium actions and involved cost we follow the lines of Algorithm 3.6.

Step 1: $H_1 = \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix}$ and $H_2 = \begin{bmatrix} -1 & -0.5 \\ -1 & 1 \end{bmatrix}$. The eigenvalues of H_1 are $\{\sqrt{2}, -\sqrt{2}\}$. An eigenvector corresponding with the eigenvalue $-\sqrt{2}$ is $[1 \ 1 - \sqrt{2}]^T$. So, H_1 has a one-dimensional stable graph subspace. In the same way one can show that H_2 has a stable graph subspace too. Therefore we can proceed with the next step in the algorithm.

Step 2: $M = \begin{bmatrix} -1 & -1 & -0.5 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$. M has the eigenvalues $\{-1.5811, 1, 1.5811\}$. Since M has one stable eigenvalue, we can proceed with step 3.

Step 3: An eigenvector of M corresponding with the eigenvalue -1.5811 is $[0.8770 \ 0.3398 \ 0.3398]^T$. From this it is clear that M has a one-dimensional stable graph subspace.

Step 4: With $p_1 = p_2 := 0.3398/0.8770 = 0.3874$ we obtain the equilibrium actions:

$$u_1^*(t) = -0.3874x^*(t) \text{ and } u_2^*(t) = -0.1937x^*(t),$$

where $x^*(t) = e^{-1.5811t}s_0$.

The with these equilibrium actions involved cost for player i are $m_i s_0^2$, where m_1 is the solution of $2 * -1.5811m_1 + 1 + 0.3874^2 = 0$ and m_2 solves $2 * -1.5811m_2 + 1 + 0.5 * 0.3874^2 = 0$ (see Theorem 3.5). That is, $J_1 = 0.3637s_0^2$ and $J_2 = 0.34s_0^2$.

Next we calculate for this example the corresponding feedback equilibrium actions and involved cost following the lines of Algorithm 4.4.

Step 1: $M = \begin{bmatrix} 1 & 1 & 0.5 & 0 \\ 1 & -1 & 0 & -0.5 \\ 1 & 0 & -1 & -1 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{-1}{3} \end{bmatrix}$ and $\sigma = 1$.

Step 2: The eigenvalues of M are $\{-1.2715, -0.8012 \pm 0.2852i, 1.5405\}$. An eigenvector of M corresponding with the eigenvalue 1.5405 is $[0.8873 \ 0.3271 \ 0.3050 \ 0.1124]$.

Step 3: From Step 2 we conclude that there is just one feedback Nash equilibrium, which results by considering the graph subspace associated with the eigenvalue 1.5405 . It follows from this graph subspace that $k_1 = 0.3271/0.8873 = 0.3687$ and $k_2 = 0.3050/0.8873 = 0.3437$. The resulting equilibrium actions are:

$$u_1^*(t) = -0.3687x^*(t) \text{ and } u_2^*(t) = -0.1719x^*(t),$$

where $x^*(t) = e^{-1.5405t} s_0$.

The with these equilibrium actions involved cost for the players are $J_1 = 0.3687s_0^2$ and $J_2 = 0.3437s_0^2$.

In Table 1 we tabulated for some different choices of r_i parameters equilibrium actions and involved cost. For simplicity we choose $s_0 = 1$. Furthermore we used the shorthand notation f_i^j and a_{cl}^j which are related to the equilibrium strategies as follows: $u_i^*(t) = -F_i^j e^{a_{cl}^j t} s_0$, $j = OL, FB$. A general observation is that the involved cost for the players are lowest for the open-loop case. Furthermore, the actions in the feedback case (FB) are initially more cautious than in the open-loop case (OL) whereas, when time evolves, from some point in time on this picture reverses. We can also see that if player one likes to follow a more stringent fiscal policy, this is only partially realized. Comparing the benchmark case, $r_1 = 1$, with the case $r_1 = 4$ we see that initially indeed its fiscal policy is less pronounced, but in the end its policy is larger than in the benchmark case. For player 2 this more stringent fiscal policy goal by player 1 implies a more active fiscal policy at every point in time.

r_1, r_2	J_1^{OL}, J_2^{OL}	J_1^{FB}, J_2^{FB}	$F_1^{OL}, F_2^{OL}, a_{cl}^{OL}$	$F_1^{FB}, F_2^{FB}, a_{cl}^{FB}$
1,2	0.3637, 0.34	0.3687, 0.3437	0.3874, 0.1937, -1.5811	0.3687, 0.1718, -1.5405
4,2	0.3955, 0.4130	0.3976, 0.4155	0.1076, 0.2153, -1.3229	0.0994, 0.2078, -1.3072
0.25,2	0.2894, 0.2227	0.2941, 0.2240	1.1957, 0.1495, -2.3452	1.1764, 0.1120, -2.2883

Table 1: Cost and Equilibrium strategies for different r_i parameters

6 Concluding Remarks

In this paper we reviewed some computational schemes to calculate the open-loop and feedback Nash equilibria in linear quadratic differential games.

For the finite planning horizon problem there exist computationally efficient tools to implement algorithms to test (and calculate) the (in general) unique equilibrium. Also for the infinite planning horizon open-loop and scalar feedback case we presented algorithms which make it (in principle) possible to calculate the unique, respectively, all equilibria. Here the addition "in principle" is added, because for large scale systems the presented invariant subspace algorithms should be implemented in a numerically reliable way. For the multivariable infinite planning horizon feedback case there exist unfortunately no general conditions under which one can conclude that there exists a unique equilibrium. Therefore, though we presented some iterative schemes which usually perform quite well, the use of these iterative schemes remains a little bit unsatisfactory since they do not provide conclusive answers about (in case convergence occurs) whether still more equilibrium points exist.

For ease of presentation we considered the two-player case. The presented algorithms can however be generalized for N -players where the cost functions have a linear quadratic structure and the system is linear affine. For the open-loop infinite planning horizon, e.g., a numerical toolbox in the line of Algorithm 3.6 is under construction in case dimensions are not too large (see [10]). Furthermore, in case the planning horizon is finite, all results can be generalized for nonstationary systems. In literature also linear uncertain systems have been considered. In, e.g., [9] one can find both a stochastic and worst-case approach of such systems together with some related computational schemes (see also [11]).

Several special cases of LQ differential games have been considered in literature. We like to mention here two cases where also recently new numerical results were reported.

First, the set of weakly coupled large-scale systems has been studied extensively by e.g. Mukaidani

in a number of papers (see e.g. [24]). This are systems where each player controls a set of states which are only marginally affected by other players. So, the corresponding LQ game almost equals an ordinary optimal LQ control problem. It can be shown that under the assumption that the coupling between the various "subsystems" is marginal the LQ game will have a unique equilibrium. In [24] a numerical algorithm which achieves quadratic convergence is proposed, based on Newton's method and fixed point iterations, to calculate the feedback Nash equilibrium.

In [4] the set of positive systems has recently been considered. That is, the case that both the state and used controls should be positive at any point in time. In this paper conditions are stated under which such a system has an equilibrium and some algorithms are devised to calculate an equilibrium.

Finally we like to mention that for discrete time systems much work has been done by Neck and coauthors in the development of the numerical software OPTGAME for the calculation of Nash equilibria in (non-)linear systems in case the performances of players are quadratic (see e.g. [25]).

References

- [1] Abou-Kandil H. and Bertrand P., 1986, Analytic solution for a class of linear quadratic open-loop Nash games, *International Journal of Control*, Vol.43, pp.997-1002.
- [2] Abou-Kandil H., Freiling G., Ionescu V. and Jank G., 2003, *Matrix Riccati Equations in Control and Systems Theory*, Birkhäuser, Berlin.
- [3] Azevedo-Perdicoúlis T.P. and Jank G., 2005, Iterative solution of algebraic matrix Riccati equations in open loop Nash games, *International Journal of Robust and Nonlinear Control*, Vol.15, pp.55-62.
- [4] Azevedo-Perdicoúlis T.P. and Jank G., 2005, Linear quadratic Nash games on positive linear systems, *European Journal of Control*, Vol.11, no.6, pp.525-535.
- [5] Başar T. and Olsder G.J., 1999, *Dynamic Noncooperative Game Theory*, SIAM, Philadelphia.
- [6] Broek W.A. van den, Engwerda J.C. and Schumacher J.M., 2003, An equivalence result in linear-quadratic theory, *Automatica*, vol.39, pp.355-359.
- [7] Dockner E., Jørgensen S., Long N. van and Sorger G., 2000, *Differential Games in Economics and Management Science*, Cambridge University Press, Cambridge.
- [8] Dooren P. van, 1981, A generalized eigenvalue approach for solving Riccati equations, *Siam Journal of Scientific Statistical Computation*, Vol.2, pp.121-135.
- [9] Engwerda J.C., 2005, *LQ Dynamic Optimization and Differential Games*, Chichester, Wiley.
- [10] Engwerda J.C., Michalak T. and Plasmans J.E.J., 2006, A numerical toolbox to solve the N -player affine LQ open-loop differential game, under construction.
- [11] Engwerda J.C., 2006, A numerical algorithm to find soft-constrained Nash equilibria in scalar LQ-games, *Int. J. of Control*, Vol.79, no.6, pp.592-603.
- [12] Feucht M., 1994, *Linear-quadratische Differentialspiele und gekoppelte Riccatische Matrixdifferentialgleichungen*, Ph.D. Thesis, Universität Ulm, Germany.

- [13] Van Hentenryck P., McAllester D. and Kapur D., 1997, Solving polynomial systems using a branch and prune approach, *SIAM J. Num. Anal.*, Vol. 34, no.2, pp.797-827.
- [14] Jørgensen S. and Zaccour G., 2004, *Differential Games in Marketing*, Kluwer, Deventer.
- [15] Kremer D., 2002, *Non-Symmetric Riccati Theory and Noncooperative Games*, Ph.D. Thesis, RWTH-Aachen, Germany.
- [16] Krikelis N. and Rekasius Z., 1971, On the solution of the optimal linear control problems under conflict of interest, *IEEE Trans. Automat. Contr.*, Vol.16, pp.140-147.
- [17] Kwakernaak H. and Sivan R., 1972, *Linear Optimal Control Systems*, Wiley, London.
- [18] Lancaster, P. and Rodman, L., 1995, *Algebraic Riccati Equations*, Oxford, Oxford University Press.
- [19] Laub A.J., 1979, A Schur method for solving algebraic Riccati equations, *IEEE Trans. Automat. Contr.*, Vol.24, pp.913-921.
- [20] Laub A.J., 1991, Invariant subspace methods for the numerical solution of Riccati equations, In: Bittanti, Laub and Willems (Eds.), *The Riccati equation*, Springer-Verlag, Berlin, pp.163-199.
- [21] Li T-Y and Gajic Z., 1994, Lyapunov iterations for solving couple algebraic Riccati equations of Nash differential games and algebraic Riccati equations of zero-sum games, *Annals of Dynamic Games*, Vol.3, pp.333-351.
- [22] Mehrmann V.L., 1991, *The autonomous Linear Quadratic Control Problem: Theory and Numerical Solution*, In: Thoma and Wyner (Eds.), *Lecture Notes in Control and Information Sciences 163*, Springer-Verlag, Berlin.
- [23] Morgan A.P., 1987, *Solving Polynomial Systems Using Continuation for Scientific and Engineering Problems*, Prentice-Hall, Englewood Cliffs, N.J..
- [24] Mukaidani H., 2006, A numerical analysis of the Nash strategy for weakly coupled large-scale systems, *IEEE Trans. Automat. Contr.*, Vol.51, pp.1371-1377.
- [25] Neck R., Hager M. and Behrens D., 2001, Solving dynamic macroeconomic policy games using the algorithm OPTGAME 1.0, *Optimal Control Applications and Methods*, 22, 301-332.
- [26] Paige C. and Van Loan C., 1981, A Schur decomposition for Hamiltonian matrices, *Linear Algebra and its Applications*, Vol.41, pp.11-32.
- [27] Papavassilopoulos G.P., Medanic J. and Cruz J., 1979, On the existence of Nash strategies and solutions to coupled Riccati equations in linear-quadratic games, *Journal of Optimization Theory and Applications*, Vol. 28, pp.49-75.
- [28] Papavassilopoulos G.P. and Olsder G.J., 1984, On the linear-quadratic, closed-loop, no-memory Nash game, *Journal of Optimization Theory and Applications*, Vol. 42, no.4, pp.551-560.
- [29] Plasmans J., Engwerda J., Aarle B. van, Bartolomeo G. di, Michalak T., 2006, *Dynamic Modeling of Monetary and Fiscal Cooperation Among Nations*, Springer-Verlag, Berlin.

- [30] Reid W.T., 1972, *Riccati Differential Equations*, Academic Press, London.
- [31] Reinganum J.F. and Stokey N.L., 1985, Oligopoly extraction of a common property natural resource: the importance of the period of commitment in dynamic games, *International Economic Review*, Vol.26, pp.161-173.
- [32] Simaan M. and Cruz J.B., Jr., 1973, On the solution of the open-loop Nash Riccati equations in linear quadratic differential games, *International Journal of Control*, Vol.18, pp.57-63.
- [33] Starr A.W. and Ho Y.C., 1969, Nonzero-sum differential games, *Journal of Optimization Theory and Applications*, Vol.3, pp.184-206.
- [34] Starr A.W. and Ho Y.C., 1969, Further properties of nonzero-sum differential games, *Journal of Optimization Theory and Applications*, Vol.3, pp.207-219.
- [35] Tabak D., 1975, Numerical solution of differential game problems, *Int. J. Syst. Science*, Vol. 6, pp.591-599, 1975.
- [36] Verschelde J., Verlinden P. and Cools R., 1994, Homotopies exploiting Newton polytopes for solving sparse polynomial systems, *SIAM J. Num. Anal.*, Vol. 31., pp.915-930.