CentER

Discussion Paper

No. 2005–33

A NUMERICAL ALGORITHM TO FIND SOFT-CONSTRAINED NASH EQUILIBRIA IN SCALAR LQ-GAMES

By Jacob Engwerda

December 2004

ISSN 0924-7815

Tilburg University

A Numerical Algorithm to find Soft-Constrained Nash Equilibria in Scalar 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

December, 2004

*This paper was presented at the MTNS 2004 Conference in Leuven, Belgium.

Abstract: In this paper we provide a numerical algorithm to calculate all soft-constrained Nash equilibria in a regular scalar indefinite linear-quadratic game. The algorithm is based on the calculation of the eigenstructure of a certain matrix. The analysis follows the lines of the approach taken by Engwerda in [7] to calculate the solutions of a set of scalar coupled feedback Nash algebraic Riccati equations.

Keywords: Linear quadratic differential games; Feedback Nash equilibria; Deterministic Uncertainty; Algebraic Riccati equations.
Jel-codes: C63, C72, C73.

1 Introduction

Dynamic game theory brings together three features that are key to many situations in economy, ecology, and elsewhere: optimizing behavior, presence of multiple agents, and enduring consequences of decisions. Usually, the dynamic model is supposed to be an exact representation of the environment in which the players act; optimization takes place with no regard of possible deviations. It can safely be assumed, however, that agents in reality follow a different strategy. If an accurate model can be formed at all, it would in general be complicated and difficult to handle. Moreover it may be unwise to optimize on the basis of a too detailed model, in view of possible changes in dynamics that may take place in the course of time and that may be hard to predict. It makes more sense for agents to work on the basis of a relatively simple model and to look for strategies that are robust with respect to deviations between the model and reality. In an economic context, the importance of incorporating aversion to specification uncertainty has been stressed for instance by Hansen *et al.* [9].

In control theory, an extensive theory of robust design is already in place; see Başar [2] for a recent survey. In van den Broek et al. [5] this background is used to arrive at suitable ways of describing aversion to model risk in a dynamic game context in a linear quadratic control setting. Such a setting is reasonable for situations of dynamic quasi-equilibrium, where no large excursions of the state vector are to be expected. One of the approaches taken in [5] is to introduce a malevolent disturbance input to model aversion to specification uncertainty. That is, it is assumed that the dynamics of the system are corrupted by a deterministic noise component, and that each player has his own expectation about this noise. This is modeled by adapting for each player his cost function accordingly. The players cope with this uncertainty by considering a worst-case scenario. As a consequence the equilibria of the game, in general, depend on the worst-case scenario expectations about the noise of the players.

More specifically, our dynamic model reads as follows:

$$\dot{x}(t) = Ax(t) + \sum_{i=1}^{N} B_i u_i(t) + Ew(t),\ x(0) = x_0. \tag{1}$$

Here $w \in L_2^q(0,\infty)$ is a q-dimensional disturbance vector affecting the system and E is a constant matrix.

We have to specify the strategy space and the information structure available to players in this setting. We assume in this deterministic approach that all players observe the state of the system, and that they use stabilizing constant linear feedback strategies. That is, we shall only consider controls u_i of the type $u_i = F_i x$, with $F_i \in I\!R^{m_i \times n}$, and where $(F_1, \cdots, F_N)$ belongs to the set

$$\mathcal{F} := \{F = (F_1, \cdots, F_N) \mid A + \textstyle\sum_{i=1}^N B_i F_i \text{ is stable}\}.$$

The stabilization constraint is imposed to ensure the finiteness of the infinite-horizon cost integrals that we will consider; also, the assumption helps to justify our basic supposition that the state vector remains close to the origin. The constraint is a bit unwieldy since it introduces dependence between the strategy spaces of the players. However, we will focus below on equilibria in which the inequalities that ensure the stability property are inactive constraints. It will be a standing assumption that the set $\mathcal{F}$ is non-empty; a necessary and sufficient condition for this to hold is that the matrix pair $(A, [B_1 \cdots B_N])$ is stabilizable. Given that we work below with an infinite horizon, restraining the players to constant feedback strategies seems reasonable; to prescribe linearity may also seem natural in the linear-quadratic context that we assume, although there is no way to exclude *a priori* equilibria in nonlinear feedback strategies. Questions regarding the existence of such equilibria are outside the scope of this paper.
For some results dealing with an open-loop information structure see e.g. [3] and [11].

We now come to the formulation of the objective functions of the players. Our starting point is the usual quadratic criterion which assigns to player i the cost function

$$J_i := \int_0^\infty \{x(t)^T Q_i x(t) + \sum_{j=1}^N u_j(t)^T R_{ij} u_j(t)\} dt. \tag{2}$$

Here, Q_i is symmetric and R_{ii} is positive definite for all $i = 1, \ldots, N$. Under our assumption that the players use constant linear feedbacks, the criterion in (2) may be rewritten as

$$J_i := \int_0^\infty \{x^T (Q_i + \sum_{j=1}^N F_j^T R_{ij} F_j) x\} dt \tag{3}$$

where F_i is the feedback chosen by player i. Written in the above form, the criterion may be looked at as a function of the initial condition x_0 and the state feedbacks F_i.

The description of the players' objectives given above needs to be modified in order to express a desire for robustness. To that end, we modify the criterion (3) to

$$\bar{J}_i^{SC}(F_1, \cdots, F_N, x_0) := \sup_{w \in L_2^q(0,\infty)} J_i(F_1, \cdots, F_N, w, x_0) \tag{4}$$

where

$$J_i(F_1, \cdots, F_N, w, x_0) := \int_0^\infty \{x^T (Q_i + \sum_{j=1}^N F_j^T R_{ij} F_j) x - w^T V_i w\} dt. \tag{5}$$

The weighting matrix V_i is symmetric and positive definite for all $i = 1, \ldots, N$. Because it occurs with a minus sign in (5), this matrix constrains the disturbance vector w in an indirect way so that it can be used to describe the aversion to model risk of player i. Specifically, if the quantity $w^T V_i w$ is large for a vector $w \in \mathbb{R}^q$, this means that player i does not expect large deviations of the nominal dynamics in the direction of Ew. In line with the nomenclature used in control theory literature we will call this the "soft-constrained" formulation. Note that since we do not assume positive definiteness of the state weighting matrix, this development extends even in the one-player case the standard results that may be found for instance in [8], [3], [12, Section 20.2], [13], [4, Section 6.6] and [2]).

The equilibrium concept that will be used throughout this chapter is based on the adjusted cost functions (4). A formal definition is given below.

Definition 1.1 An N-tuple $\overline{F} = (\overline{F}_1, \ldots, \overline{F}_N) \in \mathcal{F}$ is called a *soft-constrained Nash equilibrium* if for each $i = 1, \ldots, N$ the following inequality holds:

$$\overline{J}_i^{SC}(\overline{F}, x_0) \leq \overline{J}_i^{SC}(\overline{F}_{-i}(F), x_0) \tag{6}$$

for all $x_0 \in I\!R^n$ and for all $F \in I\!R^{m_i \times n}$ that satisfy $\overline{F}_{-i}(F) \in \mathcal{F}$. □

The remainder of this paper is organized as follows. In the next section we will discuss the two-player case. Then in section three we present the numerical algorithm to calculate the equilibria in case the system is scalar. The algorithm is illustrated in some examples and used to deal with some theoretical questions. The paper ends with some concluding remarks.

2 The two-player case

In this section we recall for the general multivariable case the two-player result. For didactical reasons the general N-player result is deferred to the Appendix. That is, we consider

$$\dot{x}(t) = (A + B_1F_1 + B_2F_2)x(t) + Ew(t),\ x(0) = x_0, \tag{7}$$

with $(A, [B_1\ \ B_2])$ stabilizable, $(F_1, F_2) \in \mathcal{F}$ and

$$J_i(F_1, F_2, w, x_0) = \int_0^\infty \{x^T(t)(Q_i + F_1^T R_{i1} F_1 + F_2^T R_{i2} F_2)x(t) - w^T(t)V_i w(t)\}dt. \tag{8}$$

Here the matrices Q_i, R_{ij} and V_i are symmetric, $R_{ii} > 0$, $V_i > 0$, and

$$\mathcal{F} := \{(F_1, F_2)|A + B_1F_1 + B_2F_2 \text{ is stable}\}.$$

For this game we want to determine all soft-constrained Nash equilibria. That is, we like to find all $(\bar{F}_1, \bar{F}_2) \in \mathcal{F}$ such that

$$\sup_{w \in L_2^q(0,\infty)} J_1(\bar{F}_1, \bar{F}_2, w, x_0) \leq \sup_{w \in L_2^q(0,\infty)} J_1(F_1, \bar{F}_2, w, x_0),\ \text{for all } (F_1, \bar{F}_2) \in \mathcal{F} \tag{9}$$

and

$$\sup_{w \in L_2^q(0,\infty)} J_2(\bar{F}_1, \bar{F}_2, w, x_0) \leq \sup_{w \in L_2^q(0,\infty)} J_2(\bar{F}_1, F_2, w, x_0),\ \text{for all } (\bar{F}_1, F_2) \in \mathcal{F}, \tag{10}$$

for all $x_0 \in I\!R^m$.

Using the shorthand notation

$$S_i := B_i R_{ii}^{-1} B_i^T, S_{ij} := B_i R_{ii}^{-1} R_{ji} R_{ii}^{-1} B_i^T, i \neq j,\ \text{and } M_i := E_i V_i^{-1} E_i^T,$$

we have the next result from van den Broek et al. [5].

Theorem 2.1 *Consider the differential game defined by (7) and (9–10). Assume there exist real symmetric $n \times n$ matrices X_i, $i = 1, 2$, and real symmetric $n \times n$ matrices Y_i, $i = 1, 2$, such that*

$$-(A - S_2X_2)^TX_1 - X_1(A - S_2X_2) + X_1S_1X_1 - Q_1 - X_2S_{21}X_2 - X_1M_1X_1 = 0, \tag{11}$$

$$-(A - S_1X_1)^TX_2 - X_2(A - S_1X_1) + X_2S_2X_2 - Q_2 - X_1S_{12}X_1 - X_2M_2X_2 = 0, \tag{12}$$

$$A - S_1X_1 - S_2X_2 + M_1X_1 \text{ and } A - S_1X_1 - S_2X_2 + M_2X_2 \text{ are stable,} \tag{13}$$

$$A - S_1X_1 - S_2X_2 \text{ is stable} \tag{14}$$

$$-(A - S_2X_2)^TY_1 - Y_1(A - S_2X_2) + Y_1S_1Y_1 - Q_1 - X_2S_{21}X_2 \le 0, \tag{15}$$

$$-(A - S_1X_1)^TY_2 - Y_2(A - S_1X_1) + Y_2S_2Y_2 - Q_2 - X_1S_{12}X_1 \le 0. \tag{16}$$

Define $\overline{F} = (\overline{F}_1, \overline{F}_2)$ by

$$\overline{F}_i := -R_{ii}^{-1}B_i^TX_i, \; i = 1, 2. \tag{17}$$

Then $\overline{F} \in \mathcal{F}$, and $\overline{F}$ is a soft-constrained Nash equilibrium. Furthermore

$$\overline{J}_i^{SC}(\overline{F}_1, \overline{F}_2, x_0) = x_0^TX_ix_0, \; i = 1, 2. \tag{18}$$

Conversely, if $(\bar{F}_1, \bar{F}_2)$ is a soft-constrained Nash equilibrium, the equations (11–14) have a set of real symmetric solutions (X_1, X_2).

Remark 2.2 Notice that if $Q_i \ge 0$, $i = 1, 2$ and $S_{ij} \ge 0$, $i, j = 1, 2$, the matrix inequalities (15–16) are trivially satisfied with $Y_i = 0$, $i = 1, 2$. So, under these conditions the differential game defined by (7) and (9–10) has a soft-constrained Nash equilibrium if and only if the equations (11–14) have a set of real symmetric $n \times n$ matrices X_i, $i = 1, 2$. □

3 A scalar numerical algorithm

From Theorem 2.1 we have that the equations (11–14) play a crucial role in the question whether the game (7–8) will have a soft-constrained Nash equilibrium. As is shown in this theorem any soft-constrained Nash equilibrium has to satisfy these equations. So, the question arises under which conditions (11–14) will have one or more solutions and, if possible, to calculate this (these) solution(s). This is a difficult open question. Fortunately, for the scalar case, one can devise an algorithm to calculate all soft-constrained Nash equilibria. This algorithm will be discussed in this section.

We will consider the general scalar N-player case under the simplifying assumptions that $b_i \neq 0$ and players have no direct interest in eachothers control actions (i.e. $S_{ij} = 0$, $i \neq j$). We will use throughout lower case notation to stress the fact that we are dealing with the scalar case. The set of equations (11–14) become (see Appendix)

$$-2(a - \sum_{j \neq i}^{N} s_jx_j)x_i + (s_i - m_i)x_i^2 - q_i = 0, \; i = 1, \cdots, N, \tag{19}$$

$$a - \sum_{j=1}^{N} s_jx_j < 0, \text{ and} \tag{20}$$

$$a - \sum_{j=1}^{N} s_jx_j + m_ix_i < 0, \; i = 1, \cdots, N. \tag{21}$$

Some elementary calculation shows that (19) can be rewritten as

$$2(a - \sum_{j=1}^{N} s_j x_j)x_i + (s_i + m_i)x_i^2 + q_i = 0,\ i = 1, \cdots, N. \tag{22}$$

For notational convenience next introduce, for $i = 1, \cdots, N$ and Ω some index set of the numbers $\{1, \cdots, N\}$[1], the variables

$$\tau_i := (s_i + m_i)q_i,\ \tau_{max} := \max_i \tau_i,\ \rho_i := \frac{s_i}{s_i + m_i},\ \gamma_i := -1 + 2\rho_i = \frac{s_i - m_i}{s_i + m_i}, \tag{23}$$
$$\gamma_\Omega := -1 + 2\sum_{i\in\Omega} \rho_i, y_i := (s_i + m_i)x_i,\ \text{and}\ y_{N+1} := -a_{cl} := -(a - \sum_{i=1}^{N} s_i x_i).$$

Moreover, assume (without loss of generality) that $\tau_1 \geq \cdots \geq \tau_N$. Multiplication of (22) by $s_i + m_i$ shows then that (19–21) has a solution if and only if

$$y_i^2 - 2y_{N+1}y_i + \tau_i = 0,\ i = 1, \cdots, N, \tag{24}$$
$$y_{N+1} = -a + \sum_{i=1}^{N} \rho_i y_i, \tag{25}$$
$$y_{N+1} + \frac{m_i}{m_i + s_i} y_i < 0,\ i = 1, \cdots, N, \tag{26}$$
$$\tag{27}$$

has a set of real solutions $y_i,\ i = 1, \cdots, N$, and $y_{N+1} > 0$. Following the analysis of [6] we conclude

Lemma 3.1
The set of equations (24,25) has a solution such that $y_{N+1} > 0$ if and only if there exist $t_i \in \{-1, 1\},\ i = 1, \cdots, N$, such that the equation

$$(-1 + \sum_{i=1}^{N} \rho_i)y_{N+1} + t_1\rho_i\sqrt{y_{N+1}^2 - \tau_1} + \cdots + t_N\rho_N\sqrt{y_{N+1}^2 - \tau_N} = a \tag{28}$$

has a solution $y_{N+1} > 0$. In fact all solutions of (24,25) are obtained by considering all possible sequences $(t_1, \cdots, t_N)$ in (28).
Obviously, a necessary condition for (28) to have a solution is that $y_{N+1}^2 \geq \tau_1$. □

Following [6] we next define recursively for $n = 1, \cdots, N-1$ the functions:

$$f_i^{n+1}(x) \ := \ f_i^n(x) + \rho_n x - \rho_n\sqrt{x^2 - \tau_{n+1}},\ i = 1, \cdots, 2^n, \tag{29}$$
$$f_{i+2^n}^{n+1}(x) \ := \ f_i^n(x) + \rho_n x + \rho_n\sqrt{x^2 - \sigma_{n+1}},\ i = 1, \cdots, 2^n, \tag{30}$$

with

$$f_1^1(x) := (-1 + \rho_1)x - \rho_1\sqrt{x^2 - \tau_1}\ \text{and}\ f_2^1(x) := (-1 + \rho_1)x + \rho_1\sqrt{x^2 - \tau_1}. \tag{31}$$

[1] So, for $N = 2$, Ω is either $\{1\},\ \{2\}$ or $\{1, 2\}$.

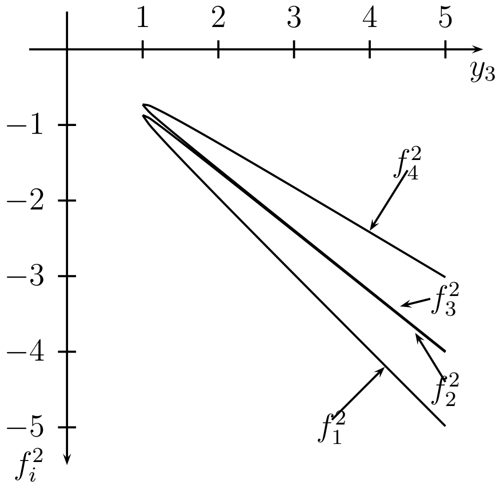

Figure 1: Curves f_i^2 in Example 3.4.

Each function f_i^N, $i = 1, \cdots, 2^N - 1$, corresponds to a function obtained from the left-hand side of (28) by making a specific choice of t_j, $j = 1, \cdots, N$, and substituting x for y_{N+1}. From Lemma 3.1 it is obvious then that (24,25) has a solution if and only if $f_i^N(x) = a$ has a solution for some $i \in \{1, \cdots, 2^N\}$. Or, stated differently, (24,25) has a solution if and only if the next function has a root

$$\Pi_{i=1}^{2^N}(f_i^N(x) - a) = 0. \tag{32}$$

Denoting the function on the left-hand side of this equation, $\Pi_{i=1}^{2^N}(f_i^N(x) - a)$, by $f(x)$ we obtain, using the same analysis as in [6], the next theorem.

Theorem 3.2 *y_i is a solution of (24,25) if and only if y_{N+1} is a zero of $f(x)$ and there exist $t_i \in \{-1, 1\}$, such that $y_i = y_{N+1} + t_i\rho_i\sqrt{y_{N+1}^2 - \tau_i}$. Moreover, $f(x)$ is a polynomial of degree 2^N.* □

An immediate consequence of this theorem is

Corollary 3.3 *The N-player scalar game has at most 2^N soft-constrained Nash equilibria.* □

In [6], Theorem 6, it was shown that the N-player scalar (undisturbed) linear quadratic differential game always has at most $2^N - 1$ feedback Nash equilibria. The next example shows that we can not draw a similar conclusion here, solely based on (24,25).

Example 3.4 Consider the two-player scalar game with $a = -2$, $s_i = 1$, $m_i = 9$, $i = 1, 2$, $q_1 = 0.1$ and $q_2 = 0.05$. For this case we plotted the four curves f_i^2 in Figure 1. From this graph we see that all curves are monotonically decreasing and they all have an intersection point with -2 for a value $y_3 > 1$. Consequently, the set of equations (24,25) has four solutions. □

Next, we develop a numerical algorithm to find all solutions of (24,25) similar to the algorithm presented in [7] to obtain all solutions of the undisturbed game. For didactical reasons again, we first consider the two-player case. Let p_1, p_2 be a (possibly complex) solution of (24,25). Denote the negative of the resulting closed-loop system parameter by

$$\lambda := -a + \rho_1 p_1 + \rho_2 p_2. \tag{33}$$

Then,

$$p_1^2 - 2\lambda p_1 + \tau_1 = 0, \tag{34}$$

and

$$p_2^2 - 2\lambda p_2 + \tau_2 = 0. \tag{35}$$

Consequently using the definition of λ and (34), respectively, we have that

$$\begin{aligned} p_1\lambda &= -p_1 a + \rho_1 p_1^2 + \rho_2 p_1 p_2 \\ &= -p_1 a + \rho_1(2\lambda p_1 - \tau_1) + \rho_2 p_1 p_2. \end{aligned}$$

From this we deduce that

$$\frac{s_1 - m_1}{s_1 + m_1} p_1 \lambda = \rho_1 \tau_1 + a p_1 - \rho_2 p_1 p_2. \tag{36}$$

In a similar way we have using the definition of λ and (35), respectively, that

$$\begin{aligned} p_2\lambda &= -p_2 a + \rho_1 p_1 p_2 + \rho_2 p_2^2 \\ &= -p_2 a + \rho_1 p_1 p_2 + \rho_2(2\lambda p_2 - \tau_2). \end{aligned}$$

Which gives rise to

$$\frac{s_2 - m_2}{s_2 + m_2} p_2 \lambda = \rho_2 \tau_2 + a p_2 - \rho_1 p_1 p_2. \tag{37}$$

Finally, using the definition of λ and both (34) and (35), respectively, we obtain

$$\begin{aligned} p_1 p_2 \lambda &= -p_1 p_2 a + \rho_1 p_1^2 p_2 + \rho_2 p_1 p_2^2 \\ &= -p_1 p_2 a + 2(\rho_1 + \rho_2)\lambda p_1 p_2 - \rho_1 \tau_1 p_2 - \rho_2 \tau_2 p_1. \end{aligned}$$

Which yields

$$(2(\rho_1 + \rho_2) - 1) p_1 p_2 \lambda = \rho_2 \tau_2 p_1 + \rho_1 \tau_1 p_2 + a p_1 p_2. \tag{38}$$

So, using the notation (23), with

$$\tilde{M} := \begin{bmatrix} -a & \rho_1 & \rho_2 & 0 \\ \frac{\rho_1 \tau_1}{\gamma_1} & \frac{a}{\gamma_1} & 0 & -\frac{\rho_2}{\gamma_1} \\ \frac{\rho_2 \tau_2}{\gamma_2} & 0 & \frac{a}{\gamma_2} & -\frac{\rho_1}{\gamma_2} \\ 0 & \frac{\rho_2 \tau_2}{\gamma_{12}} & \frac{\rho_1 \tau_1}{\gamma_{12}} & \frac{a}{\gamma_{12}} \end{bmatrix} \tag{39}$$

we conclude from (33,36,37,38) that, provided $\gamma_i \neq 0 \;\; i = 1, 2, 3$, every solution p_1, p_2 of (24,25) satisfies the equation

$$\tilde{M} \begin{bmatrix} 1 \\ p_1 \\ p_2 \\ p_1 p_2 \end{bmatrix} = \lambda \begin{bmatrix} 1 \\ p_1 \\ p_2 \\ p_1 p_2 \end{bmatrix}. \tag{40}$$

Using the fact that $p_i = (s_i + m_i) x_i$ an analogous reasoning as in [7], Theorem 7, gives the next lemma.

Lemma 3.5
1) Assume that (x_1, x_2) *solves (19,20) and* $\gamma_i \neq 0,\ i = 1, 2, 3$. *Then* $\lambda := -a + \sum_{i=1}^{2} s_i x_i > 0$ *is an eigenvalue of the matrix*

$$M := \begin{bmatrix} -a & s_1 & s_2 & 0 \\ \frac{\rho_1 q_1}{\gamma_1} & \frac{a}{\gamma_1} & 0 & -\frac{s_2}{\gamma_1} \\ \frac{\rho_2 q_2}{\gamma_2} & 0 & \frac{a}{\gamma_2} & -\frac{s_1}{\gamma_2} \\ 0 & \frac{\rho_2 q_2}{\gamma_{12}} & \frac{\rho_1 q_1}{\gamma_{12}} & \frac{a}{\gamma_{12}} \end{bmatrix}. \tag{41}$$

Furthermore, $[1, x_1, x_2, x_1 x_2]^T$ *is a corresponding eigenvector and* $\lambda^2 \geq \tau_{max}$.
2) Assume that $[1, x_1, x_2, x_3]^T$ *is an eigenvector corresponding to a positive eigenvalue* λ *of* M, *satisfying* $\lambda^2 \geq \tau_{max}$, *and that the eigenspace corresponding with* τ *has dimension one. Then,* (x_1, x_2) *solves (19,20).* □

From Lemma 3.1 and Lemma 3.5 we have then the next numerical algorithm.

Algorithm 3.6 Let $s_i := \frac{b_i^2}{r_i}$ and $m_i := \frac{e^2}{v_i}$. Assume that for every index set $\Omega \subset \{1, \cdots, N\}$, $\gamma_\Omega \neq 0$. Then, the following algorithm calculates all solutions of (19,20)

Step 1 Calculate matrix M in (41) and $\tau := \max_i (s_i + m_i) q_i$.

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

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

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

3.2) If $n_i = 1$ then

3.2.1) calculate an eigenvector z corresponding with λ_i of M. Denote the entries of z by $(z_0, z_1, z_2, \cdots)$. Calculate $x_j := \frac{z_j}{z_0}$. Then, $(x_1, \cdots, x_N)$ solve (19,20). Return to step 3.

If $n_i > 1$ then

3.2.2) Calculate $\tau_i := s_i q_i$.

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

i) calculate

$$y_j := \lambda_i + t_j \frac{s_i}{s_i + m_i} \sqrt{\lambda_i^2 - \sigma_j},\ j = 1, \cdots, N$$

ii) If $\lambda_i = -a + \sum_{j=1,\cdots,N} y_j$ then calculate $x_j := \frac{y_j}{s_j + m_j}$. Then, $(x_1, \cdots, x_N)$ solves (19,20).

Step 4 End of the algoritm. □

Example 3.7 Consider the two-player scalar game with $a = -2$, $b_i = e = 1$, $r_i = 1$, $v_i = \frac{1}{9}$, $i = 1, 2$, $q_1 = 0.1$ and $q_2 = 0.05$.

To calculate the soft-constrained Nash equilibria of this game, we first determine all solutions of (19,20). According Algorithm 3.6, we first have to determine the eigenstructure of the next matrix

$$M := \begin{bmatrix} 2 & 1 & 1 & 0 \\ -1/80 & 5/2 & 0 & 5/4 \\ -1/160 & 0 & 5/2 & 5/4 \\ 0 & -1/120 & -1/60 & 10/3 \end{bmatrix}.$$

Using Matlab, we find the eigenvalues $\{2.0389, 2.4866, 2.5132, 3.2946\}$. Since all eigenvalues are larger than $\tau = 1$, we have to process Step 3 of the algorithm for all these eigenvalues. Since all eigenvalues have a geometric multiplicity of one, we find four solutions satisfying (19,20). From the corresponding eigenspaces we obtain then the solutions tabulated below (with $a_{cl} = a - s_1x1 - s_2x_2 = -$eigenvalue):

eigenvalue	(x_1, x_2)	$a_{cl} + m_1x_1$	$a_{cl} + m_2x_2$
2.0389	(0.0262,0.0127)	-1.8030	-1.9250
2.4866	(0.4763,0.0103)	1.8003	-2.3942
2.5132	(0.0208,0.4925)	-2.3265	1.9192
3.2946	(0.6434,0.6512)	2.4958	2.5666

From the last two columns of this table we see that only the first solution satisfies the additional conditions (21). Since $q_i > 0$, and thus (15,16) are satisfied with $y_i = 0$, we conclude that this game has one soft-constrained Nash equilibrium. The with this equilibrium $(0.0262, 0.0127)$ corresponding equilibrium actions are

$$u_1^*(t) = -0.0262x(t) \text{ and } u_2^*(t) = -0.0127x(t).$$

Assuming that the initial state of the system is x_0, the worst-case expected cost by the players are

$$J_1^* = 0.0262x_0^2 \text{ and } J_2^* = 0.0127x_0^2,$$

respectively. □

Remark 3.8 In case k of the γ_i parameters are zero, we obtain k linear equations in the variables $(1, p_1, p_2, p_1p_2)$. Under some regularity conditions, k of these variables can then be explicitly solved as a function of the remaining $2^N - k$ variables. The solutions of the remaining $2^N - k$ equations can then be obtained using a similar eigenstructure algorithm.

As an example consider the case that in the above described two-player case $\gamma_1 = 0$ (and $\gamma_j \neq 0,\ j = 2, 3$). So, equations (33,36,37,38) reduce to

$$\lambda := -a + \rho_1 p_1 + \rho_2 p_2, \tag{42}$$

$$0 = \rho_1\tau_1 + ap_1 - \rho_2 p_1 p_2 \tag{43}$$

$$\gamma_2 p_2 \lambda = \rho_2\tau_2 + ap_2 - \rho_1 p_1 p_2 \tag{44}$$

$$\gamma_{12} p_1 p_2 \lambda = \rho_2\tau_2 p_1 + \rho_1\tau_1 p_2 + ap_1p_2. \tag{45}$$

From equation (43) we can then solve, e.g., if $a \neq 0$,

$$p_1 = \frac{-\rho_1\tau_1}{a} + \frac{\rho_2}{a} p_1 p_2. \tag{46}$$

Substitution of this into the remaining three equations (42, 44,45) yields

$$\begin{aligned} \lambda &:= -a - \frac{\rho_1^2\tau_1}{a} + \rho_2 p_2 + \frac{\rho_1\rho_2}{a} p_1 p_2, & (47) \\ \gamma_2 p_2 \lambda &= \rho_2\tau_2 + a p_2 - \rho_1 p_1 p_2 & (48) \\ \gamma_{12} p_1 p_2 \lambda &= \frac{-\rho_1\rho_2\tau_1\tau_2}{a} + \rho_1\tau_1 p_2 + (\frac{\rho_2^2\tau_2}{a} + a) p_1 p_2. & (49) \end{aligned}$$

Or, stated differently,

$$\begin{bmatrix} -a - \frac{\rho_1^2\tau_1}{a} & \rho_2 & \frac{\rho_1\rho_2}{a} \\ \frac{\rho_2\tau_2}{\gamma_2} & \frac{a}{\gamma_2} & -\frac{\rho_1}{\gamma_2} \\ \frac{-\rho_1\rho_2\tau_1\tau_2}{\gamma_{12}a} & \frac{\rho_1\tau_1}{\gamma_{12}} & \frac{(\frac{\rho_2^2\tau_2}{a}+a)}{\gamma_{12}} \end{bmatrix} \begin{bmatrix} 1 \\ p_2 \\ p_1 p_2 \end{bmatrix} = \lambda \begin{bmatrix} 1 \\ p_2 \\ p_1 p_2 \end{bmatrix}.$$

By solving this eigenvalue problem, in a similar way as described in Algorithm 3.6, one can determine the solutions p_2 and $p_1 p_2$ of the set of equations (47–49). Substitution of this result into (46) yields then p_1. $\square$

For the general N-player case we proceed as in [7].
Let p_i, $i = 1, \cdots, N$ be a solution of (24,25). Denote the negative of the resulting closed-loop system parameter by

$$\lambda := -a + \sum_i \rho_i p_i. \tag{50}$$

Then,

$$p_i^2 - 2\lambda p_i + \tau_i = 0,\ i = 1, \cdots, N. \tag{51}$$

Next we derive, again, for each index set $\Omega \subset \{1, \cdots, N\}$ a linear equation (linear in terms of products of p_i variables (Πp_i)). This gives us in addition to (50) another $2^N - 1$ linear equations. These equations, together with (50), determine our matrix M. In case Ω contains only 1 number we have, using the definition of λ and (51), respectively

$$\begin{aligned} p_j\lambda &= p_j(-a + \sum_{i=1}^{N} \rho_i p_i) = -a p_j + \rho_j p_j^2 + p_j \sum_{i \neq j} \rho_i p_i \\ &= -a p_j + 2\lambda\rho_j p_j - \rho_j\tau_j + p_j \sum_{i \neq j} \rho_i p_i. \end{aligned}$$

From which we deduce that

$$p_j\lambda = \frac{\rho_j\tau_j}{\gamma_j} + \frac{a}{\gamma_j} p_j - p_j \sum_{i \neq j} \frac{\rho_i}{\gamma_j} p_i,\ j = 1, \cdots, N.$$

Next consider the general case $\Pi_{j\in\Omega} p_j \lambda$. For notational convenience we use the notation Ω_{-i} to denote the set of all numbers that are in Ω except number i. Then,

$$\Pi_{j\in\Omega}\ p_j\lambda = -a\Pi_{j\in\Omega}\ p_j + 2\lambda \sum_{i\in\Omega} \rho_i \Pi_{j\in\Omega}\ p_j - \sum_{i\in\Omega} \rho_i\tau_i \Pi_{j\in\Omega_{-i}}\ p_j + \sum_{i\notin\Omega} \Pi_{j\in\Omega}\ p_j \rho_i p_i.$$

Therefore, with $\gamma_\Omega = -1 + 2\sum_{i\in\Omega} \rho_i$, we conclude that

$$\Pi_{j\in\Omega}\ p_j\lambda = \frac{1}{\gamma_\Omega}\{a\Pi_{j\in\Omega}\ p_j + \sum_{i\in\Omega} \rho_i\tau_i\Pi_{j\in\Omega_{-i}}\ p_j + \sum_{i\notin\Omega} \Pi_{j\in\Omega}\ p_j\rho_i p_i\}. \tag{52}$$

Equations (50) and (52) determine the matrix $\tilde{M}$. That is, introducing

$$p := [1, p_1, \cdots, p_N, p_1p_2, \cdots, p_{N-1}p_N, \cdots, \Pi_{i=1}^N p_i]^T$$

we have that $\tilde{M}p = \lambda p$. Since $p_i = (s_i + m_i)x_i$ and $\tau_i = (s_i + m_i)q_i$, matrix M is then easily obtained from $\tilde{M}$ by rewriting p as $p = Dx$, where $x := [1, x_1, \cdots, x_N, x_1x_2, \cdots, x_{N-1}x_N, \cdots, \Pi_{i=1}^N x_i]^T$ and D is a diagonal matrix defined by $D := \text{diag}\{1, s_1 + m_1, \cdots, s_N + m_N, (s_1 + m_1)(s_2 + m_2), \cdots, (s_{N-1} + m_{N-1})(s_N + m_N), \cdots, \Pi_{i=1}^N(s_i + m_i)\}$. Obviously, $M = D^{-1}\tilde{M}D$. Below we elaborated the case for $N = 3$.

Example 3.9 Consider the three-player case. With $p := [1, p_1, p_2, p_3, p_1p_2, p_1p_3, p_2p_3, p_1p_2p_3]^T$,

$$\begin{aligned} D = \text{diag} \quad & \{1, s_1 + m_1, s_2 + m_2, s_3 + m_3, (s_1 + m_1)(s_2 + m_2), (s_1 + m_1)(s_3 + m_3), \\ & (s_2 + m_2)(s_3 + m_3), (s_1 + m_1)(s_2 + m_2)(s_3 + m_3)\}, \end{aligned}$$

and

$$\tilde{M} = \begin{bmatrix} -a & \rho_1 & \rho_2 & \rho_3 & 0 & 0 & 0 & 0 \\ \frac{\rho_1\tau_1}{\gamma_1} & \frac{a}{\gamma_1} & 0 & 0 & -\frac{\rho_2}{\gamma_1} & -\frac{\rho_3}{\gamma_1} & 0 & 0 \\ \frac{\rho_2\tau_2}{\gamma_2} & 0 & \frac{a}{\gamma_2} & 0 & -\frac{\rho_1}{\gamma_2} & 0 & -\frac{\rho_3}{\gamma_2} & 0 \\ \frac{\rho_3\tau_3}{\gamma_3} & 0 & 0 & \frac{a}{\gamma_3} & 0 & -\frac{\rho_1}{\gamma_3} & -\frac{\rho_2}{\gamma_3} & 0 \\ 0 & \frac{\rho_2\tau_2}{\gamma_{12}} & \frac{\rho_1\tau_1}{\gamma_{12}} & 0 & \frac{a}{\gamma_{12}} & 0 & 0 & -\frac{\rho_3}{\gamma_{12}} \\ 0 & \frac{\rho_3\tau_3}{\gamma_{13}} & 0 & \frac{\rho_1\tau_1}{\gamma_{13}} & 0 & \frac{a}{\gamma_{13}} & 0 & -\frac{\rho_2}{\gamma_{13}} \\ 0 & 0 & \frac{\rho_3\tau_3}{\gamma_{23}} & \frac{\rho_2\sigma_2}{\gamma_{23}} & 0 & 0 & \frac{a}{\gamma_{23}} & -\frac{\rho_1}{\gamma_{23}} \\ 0 & 0 & 0 & 0 & \frac{\rho_3\tau_3}{\gamma_{123}} & \frac{\rho_2\tau_2}{\gamma_{123}} & \frac{\rho_1\tau_1}{\gamma_{123}} & \frac{a}{\gamma_{123}} \end{bmatrix}.$$

Which yields,

$$M = \begin{bmatrix} -a & s_1 & s_2 & s_3 & 0 & 0 & 0 & 0 \\ \frac{\rho_1 q_1}{\gamma_1} & \frac{a}{\gamma_1} & 0 & 0 & -\frac{s_2}{\gamma_1} & -\frac{s_3}{\gamma_1} & 0 & 0 \\ \frac{\rho_2 q_2}{\gamma_2} & 0 & \frac{a}{\gamma_2} & 0 & -\frac{s_1}{\gamma_2} & 0 & -\frac{s_3}{\gamma_2} & 0 \\ \frac{\rho_3 q_3}{\gamma_3} & 0 & 0 & \frac{a}{\gamma_3} & 0 & -\frac{s_1}{\gamma_3} & -\frac{s_2}{\gamma_3} & 0 \\ 0 & \frac{\rho_2 q_2}{\gamma_{12}} & \frac{\rho_1 q_1}{\gamma_{12}} & 0 & \frac{a}{\gamma_{12}} & 0 & 0 & -\frac{s_3}{\gamma_{12}} \\ 0 & \frac{\rho_3 q_3}{\gamma_{13}} & 0 & \frac{\rho_1 q_1}{\gamma_{13}} & 0 & \frac{a}{\gamma_{13}} & 0 & -\frac{s_2}{\gamma_{13}} \\ 0 & 0 & \frac{\rho_3 q_3}{\gamma_{23}} & \frac{\rho_2 q_2}{\gamma_{23}} & 0 & 0 & \frac{a}{\gamma_{23}} & -\frac{s_1}{\gamma_{23}} \\ 0 & 0 & 0 & 0 & \frac{\rho_3 q_3}{\gamma_{123}} & \frac{\rho_2 q_2}{\gamma_{123}} & \frac{\rho_1 q_1}{\gamma_{123}} & \frac{a}{\gamma_{123}} \end{bmatrix}.$$

Using this matrix M in Algorithm 3.6 one can determine all solutions of the three-player scalar equations (19,20). $\square$

Example 3.10 Consider a monetary union policy game with two countries using their fiscal policy and one common Central Bank using its monetary policy to stabilize output and prices (see e.g. van Aarle et al. [1] for more details). Assume that both countries do not expect severe external shocks on the economy, whereas the Central Bank is somewhat less optimistic in this respect. We model this by considering the game:

$$\dot{s}(t) = -s(t) - f_1(t) + f_2(t) + \frac{1}{2}i_E(t) + w(t), \; s(0) = s_0,$$

with

$$\begin{aligned} J_1 &:= \int_0^\infty \{2s^2(t) + f_1^2(t) - 4w^2(t)\}dt, \\ J_2 &:= \int_0^\infty \{2s^2(t) + 2f_2^2(t) - 4w^2(t)\}dt, \end{aligned}$$

and

$$J_E := \int_0^\infty \{s^2(t) + 3i_E^2(t) - 2w^2(t)\}dt.$$

Here s is the competitiveness of country 2 vis-a-vis country 1, f_i the real fiscal deficit and i_E the real interest rate. With these parameters, $\rho_1 = 4/5$, $\rho_2 = 2/3$, $\rho_3 = 1/7$. Consequently, $\gamma_1 = 3/5$, $\gamma_2 = 1/3$, $\gamma_{12} = -5/7$, $\gamma_{12} = -1 + 2\rho_1 + 2\rho_2 = 29/15$, $\gamma_{13} = -1 + 2\rho_1 + 2\rho_3 = 31/35$, $\gamma_{23} = -1 + 2\rho_2 + 2\rho_3 = 13/21$, and $\gamma_{123} = -1 + 2\rho_1 + 2\rho_2 + 3\rho_3 = 233/105$. To calculate the soft-constrained Nash equilibria of this game, we first determine all solutions of (19,20). According Algorithm 3.6, we first have to determine the eigenstructure of the next matrix

$$M := \begin{bmatrix} 1 & 1 & 1/2 & 1/12 & 0 & 0 & 0 & 0 \\ 8/3 & -5/3 & 0 & 0 & -5/6 & -5/36 & 0 & 0 \\ 4 & 0 & -3 & 0 & -3 & 0 & -1/4 & 0 \\ -1/5 & 0 & 0 & 7/5 & 0 & 7/5 & 7/10 & 0 \\ 0 & 20/29 & 24/29 & 0 & -15/29 & 0 & 0 & -5/116 \\ 0 & 5/31 & 0 & 56/31 & 0 & -35/31 & 0 & -35/62 \\ 0 & 0 & 3/13 & 28/13 & 0 & 0 & -21/13 & -21/13 \\ 0 & 0 & 0 & 0 & 15/233 & 140/233 & 168/233 & -105/233 \end{bmatrix}.$$

Using Matlab, we find the eigenvalues

$$\{-2.1369, -1.7173, -1.9576 \pm 0.2654i, -1.267 \pm 0.5041i, 1.9543, 2.37\}.$$

Since the square of every positive eigenvalue is larger than $\tau = 5/2$ $(= \max\{5/2, 3/2, 7/12\})$, we immediately conclude from Algorithm 3.6 that the equations (19,20) have two solutions.

From the corresponding eigenspaces we obtain the solutions tabulated below (with $a_{cl} = a - s_1x_1 - s_2x_2 - s_3x_3$):

eigenvalue	(x_1, x_2, x_3)	$a_{cl} + m_1x_1$	$a_{cl} + m_2x_2$	$a_{cl} + m_3x_3$
2.37	(0.4836,0.4546,7.909)	-2.2491	-2.2564	1.5845
1.9543	(0.6445,0.5752,0.2664)	-1.7932	-1.8105	-1.8211

From the last three columns of this table we see that only the second solution satisfies the additional conditions (21). Since $q_i > 0$, and thus (15,16) are satisfied with $y_i = 0$, we conclude that this game has one soft-constrained Nash equilibrium. The with this equilibrium $(x_1^*, x_2^*, x_3^*) := (0.6445, 0.5752, 0.2664)$ corresponding equilibrium actions are

$$f_1^*(t) = x_1^* s(t),\ f_2^*(t) = -\frac{1}{2} x_2^* s(t) \text{ and } i_E^*(t) = -\frac{1}{6} x_3^* s(t).$$

Assuming that the initial state of the system is x_0, the worst-case expected cost by the players are

$$J_1^* = 0.6445 s_0^2, J_2^* = 0.5752 s_0^2 \text{ and } J_E^* = 0.2664 s_0^2,$$

respectively.
Using [7, Algorithm 8] we obtain for the undisturbed case the next equilibrium strategies.

$$f_1^*(t) = k_1 s(t),\ f_2^*(t) = -\frac{1}{2} k_2 s(t),\ \text{ and } i_E^*(t) = -\frac{1}{6} k_E s(t),$$

where

$$k_1 = \frac{v(2)}{v(1)} = 0.6202,\ k_2 = \frac{v(3)}{v(1)} = 0.5611,\ k_E = \frac{v(4)}{v(1)} = 0.2616.$$

This results in the next closed-loop system and cost

$$\dot{s}(t) = -1.9225 s(t) \text{ and } J_i = k_i s_0^2,\ i = 1, 2, E.$$

Taking a more closer look at the equilibrium actions we see that all players use more control efforts than in the undisturbed case. The ratio of the increase in control efforts used by the fiscal player 1, fiscal player 2 and the Central Bank is approximately 6 : 4 : 3. The expected increase in worst-case cost by these three players is approximately 3.7%, 2.5% and 1.8%, respectively. So we see that though, at first sight, it seems that the Central Bank is the most risk-averse player in this game, due to the model structure the Bank will suffer least from an actual realization of a worst-case scenario. Also, in coping with this uncertainty, the Bank deviates least from its original equilibrium action. Finally, we observe that in case the players take uncertainty into account, the implemented equilibrium policies yield a closed-loop system which adjustes faster towards its equilibrium value $s = 0$. □

4 Properties of the scalar two-player case

In this section we analyze the consequences of taking deterministic noise into consideration in some more detail for the two-player case. We list some properties pointwise below.

1. The first point we like to make is that the incorporation of noise by players into their decision making may result in the fact that a situation of no equilibrium changes into a situation in which an equilibrium does exist. Take, e.g., $q_i = -1$; $b_i = r_i = v_i = e = 1$ and $a = -\frac{3}{2}$. For these parameters the undisturbed game has no equilibrium (see e.g. [7, Algorithm 8]). By, e.g., a direct substitution of these parameters, $x_i = -\frac{1}{2}$ and $y_i = -1$ in (19–21) and (15,16)), we see that all

conditions of Theorem 2.1 are satisfied. So the disturbed game does have an equilibrium.
2. The opposite effect as described in item 1. can also occur. That is a game in which first an equilibrium occurred may result in a game without an equilibrium if players take noise into consideration into their decision making. Take e.g. $a=-0.2$, $s_i=q_i=1,\ i=1,2$, $m_1=0.01$ and $m_2=1.5$. Using both algorithms again, we see that the undisturbed game has an equilibrium, whereas the (slightly) disturbed game has no equilibrium.
3. Taking the approach in [5] one can also analyse the effect of a unilateral variation in the attitude towards noise on the equilibrium outcome of the game.
To that end, rewrite $m_i =: \alpha_i s_i,\ i=1,2$. Then the Riccati equations (19) become

$$\begin{aligned} 2(a-s_1x_1-s_2x_2)x_1+s_1(1+\alpha_1)x_1^2+q_1 &= 0, \text{ and} && (53)\\ 2(a-s_1x_1-s_2x_2)x_2+s_2(1+\alpha_2)x_2^2+q_2 &= 0. && (54)\end{aligned}$$

Assuming that the equilibrium (x_1^*,x_2^*) can be described locally as a function $h(\alpha_1,\alpha_2)$, using the implicit function theorem, we get from (53,54) that

$$h'=\frac{-1}{d}\begin{bmatrix} -p_2 & s_2x_1^* \\ s_1x_2^* & -p_1\end{bmatrix}\begin{bmatrix} s_1{x_1^*}^2 & 0 \\ 0 & s_2{x_2^*}^2\end{bmatrix}$$

where $p_i := -(a-s_1x_1^*-s_2x_2^*+m_ix_i^*)>0$ (see (21)) and $d=2(p_1p_2-s_1s_2x_1^*x_2^*)$. From this we observe in particular that the consequences of a unilateral deviation by player 1 on the solutions (x_1^*,x_2^*) is

$$\frac{\partial x_1^*}{\partial\alpha_1}=\frac{p_2s_1{x_1^*}^2}{d} \text{ and } \frac{\partial x_2^*}{\partial\alpha_1}=\frac{-s_1^2{x_1^*}^2x_2^*}{d}.$$

Depending on the sign of x_1^* and d we see that various reactions are possible. If both d and x_2^* are positive, the effect of an increased uncertainty attitude by player 1 is that x_1^* increases and x_2^* decreases (take, e.g., $a=q_i=s_i=1$, $m_1=1.2$ and $m_2=0.5$). Since $u_i^*(t)=-\frac{b_i}{r_i}x_i^*x(t)$, this implies that the response by player 1 is to use more control efforts, whereas player 2 uses less control efforts. An opposite reaction is obtained by both players in case d is negative and x_2^* positive (which happens, e.g., if we choose $a=s_i=q_i=1$ and $m_i=1.1$). In case $x_2^*<0$ we see that both players will react in the same direction by either both increasing their control efforts ($d>0$) (as is the case in the example we considered in item 1.) or lowering their control efforts ($d<0$).
4. From Gershgorin's circle criterion it is clear from M that if a is very negative; the parameters ρ_i, q_i and s_i not too large; and γ_i not too close to zero Algorithm 3.6 will provide exactly one solution.
5. In case both players expect much noise and either q_1 or q_2 is strictly positive, there will be no equilibrium (unless a is very stable). This follows from the fact that if $m_i\to\infty,\ i=1,2$, matrix M converges to

$$\begin{bmatrix} -a & s_1 & s_2 & 0\\ 0 & -a & 0 & s_2\\ 0 & 0 & -a & s_1\\ 0 & 0 & 0 & -a\end{bmatrix}.$$

So, the eigenvalues of M all approach $-a$. From Step 3.1 of Algorithm 3.6 we conclude then that a^2 should be at least larger than $(s_i+m_i)q_i,\ i=1,2$.

6. If the players expect almost no noise (i.e. m_i is almost zero), matrix M converges to the matrix that has to be analyzed for the undisturbed game.
7. In case just one player expects a large impact of noise and the other player is modest in his expectations, matrix M converges to

$$\begin{bmatrix} -a & s_1 & s_2 & 0 \\ 0 & -a & 0 & s_2 \\ \frac{s_2 q_2}{s_2 - m_2} & 0 & \frac{a}{\gamma_2} & \frac{-s_1}{\gamma_2} \\ 0 & \frac{s_2 q_2}{s_2 - m_2} & 0 & \frac{a}{\gamma_2} \end{bmatrix}.$$

The characteristic polynomial of this matrix is

$$(\lambda^2 - a(\frac{1}{\gamma_2} - 1)\lambda - (\frac{a^2}{\gamma_2} - \frac{s_2^2 q_2}{s_2 - m_2}))^2.$$

From this we infer that only in case $a > 0$ and s_2 a little bit larger than m_2 Algorithm 3.6 may have an appropriate solution.

5 Concluding Remarks

In this paper we considered the problem to calculate the feedback Nash equilibria in a system that is corrupted by deterministic noise. Basically, we followed the approach taken by Engwerda in [7] to develop the numerical algorithm. The number of equilibrium points can be analyzed by considering the eigenstructure of a $2^N \times 2^N$ matrix M, where N denotes the number of involved players. The algorithm is very efficient in case the eigenspaces of M have dimension one and can be easily implemented in Matlab. Unfortunately, the generalization for the multivariable case is unclear. The advantage of this approach is that matrix M might be useful in further theoretical developments such as, e.g., determining parametric conditions under which there will cease to exist equilibria or, from an opposite point of view, the introduction of noise into the game will bring on the occurence of equilibria.
A disadvantage of the numerical approach taken here is that the size of matrix M we have to analyze grows exponentially if the number of players increases. On the other hand, matrix M has some tri-diagonal structure, which might be exploited in case the number of players grows to develop efficient numerical tools. However, it seems that e.g. the interval method as developed by van Hentenryck and coauthors (see e.g. [10]) might numerically be more efficient then. A point which has to be elaborated in using the interval method is the choice of the initial interval that contains all equilibria. A nice feature of the interval method is that, in principle, it can be used to calculate all equilibria in the multivariable case too. However, the choice of the initial interval containing all equilibria is a problem that has to be managed in this case.

Appendix

Below we describe the general N-player result of Theorem 2.1.

Theorem 5.1 *Consider the differential game defined by (1) and (4–5). Assume there exist N real symmetric $n \times n$ matrices X_i and N real symmetric $n \times n$ matrices Y_i such that*

$$-(A - \sum_{j \neq i}^{N} S_j X_j)^T X_i - X_i (A - \sum_{j \neq i}^{N} S_j X_j) + X_i S_i X_i - Q_i - \sum_{j \neq i}^{N} X_j S_{ij} X_j - X_i M_i X_i = 0,$$

$$A - \sum_{j=1}^{N} S_j X_j + M_i X_i \text{ is stable for } i = 1, \dots, N$$

$$A - \sum_{j=1}^{N} S_j X_j \text{ is stable}$$

$$-(A - \sum_{j \neq i}^{N} S_j X_j)^T Y_i - Y_i (A - \sum_{j \neq i}^{N} S_j X_j) + Y_i S_i Y_i - Q_i - \sum_{j \neq i}^{N} X_j S_{ij} X_j \leq 0.$$

Define the N-tuple $\overline{F} = (\overline{F}_1, \dots, \overline{F}_N)$ by

$$\overline{F}_i := -R_{ii}^{-1} B_i^T X_i. \tag{55}$$

Then $\overline{F} \in \mathcal{F}$, and this N-tuple is a soft-constrained Nash equilibrium. Furthermore

$$\overline{J}_i^{SC}(\overline{F}_1, \dots, \overline{F}_N, x_0) = x_0^T X_i x_0. \tag{56}$$

References

[1] Aarle B. van, Engwerda J.C. and Plasmans J., 2002, Monetary and fiscal policy interaction in the EMU: a dynamic game approach, *Annals of Operations Research*, Vol.109, pp.229-264.

[2] Başar T., 2003, Paradigms for robustness in controller and filter design, in: Neck (Ed.) *Modeling and Control of Economic Systems*, Elsevier, pp.1-13.

[3] Başar T. and Bernhard P., 1995, *H_∞-Optimal Control and Related Minimax Design Problems*, Birkhäuser, Boston.

[4] Başar T. and Olsder G.J., 1999, *Dynamic Noncooperative Game Theory*, SIAM, Philadelphia.

[5] Broek W.A. van den, Engwerda J.C. and Schumacher J.M., 2003, Robust equilibria in indefinite linear-quadratic differential games, *Journal of Optimization Theory and Applications*, 565-595.

[6] Engwerda J.C., 2000, The solution set of the N-player scalar feedback Nash algebraic Riccati equations, *IEEE Trans. Automat. Contr.*, Vol.45, pp.2363-2368.

[7] Engwerda J.C., 2003, Solving the scalar feedback Nash algebraic Riccati equations: an eigenvector approach, *IEEE Trans. Automat. Contr.*, Vol.48, pp.847-853.

[8] Francis B.A., 1987, *A course in H_∞ Control Theory*, Lecture Notes in Control and Information Sciences Vol. 88, Springer-Verlag, Berlin.

[9] Hansen L.P., Sargent T.J. and Tallarini Jr. T.D., 1999, Robust permanent income and pricing, *Review of Economic Studies*, Vol.66, pp.873-907.

[10] 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.

[11] Kun G., 2001, *Stabilizability, Controllability, and Optimal Strategies of Linear and Nonlinear Dynamical Games*, PhD. Thesis, RWTH-Aachen, Germany.

[12] Lancaster P. and Rodman L., 1995, *Algebraic Riccati Equations*, Clarendon Press, Oxford.

[13] Zhou K., J.C. Doyle, and K. Glover, 1996, *Robust and Optimal Control*, Prentice Hall, New Jersey.