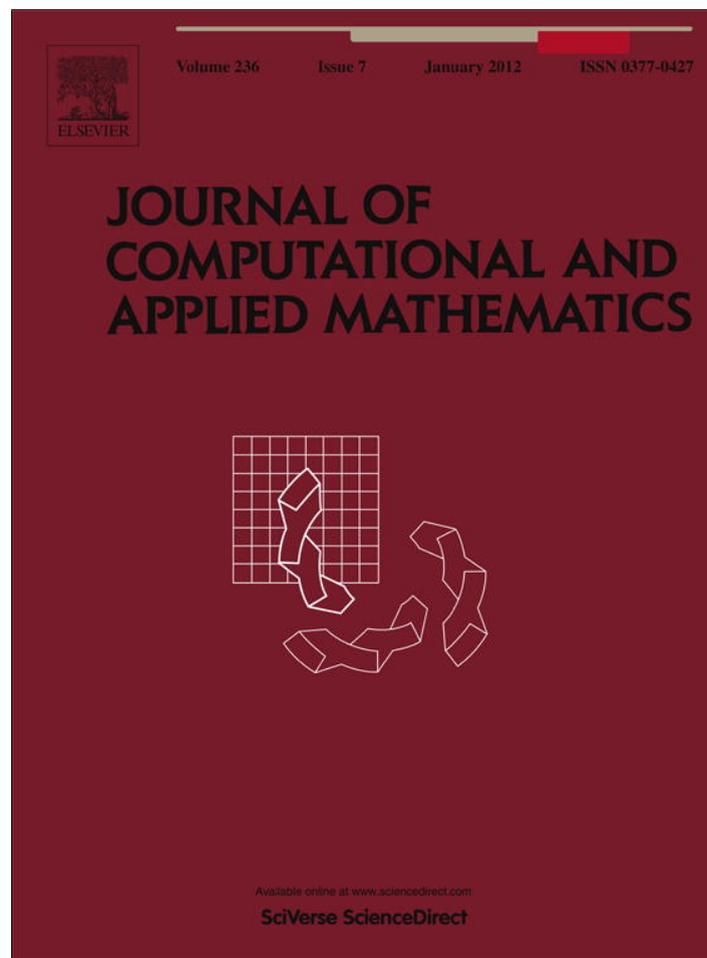


Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



(This is a sample cover image for this issue. The actual cover is not yet available at this time.)

This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Magnus integrators for solving linear-quadratic differential games

Sergio Blanes*, Enrique Ponsoda

Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, 46022 Valencia, Spain

ARTICLE INFO

Article history:

Received 22 June 2010

Received in revised form 23 December 2011

Keywords:

Differential games

Coupled Riccati differential equations

Exponential integrators

ABSTRACT

We consider Magnus integrators to solve linear-quadratic N -player differential games. These problems require to solve, backward in time, non-autonomous matrix Riccati differential equations which are coupled with the linear differential equations for the dynamic state of the game, to be integrated forward in time. We analyze different Magnus integrators which can provide either analytical or numerical approximations to the equations. They can be considered as time-averaging methods and frequently are used as exponential integrators. We show that they preserve some of the most relevant qualitative properties of the solution for the matrix Riccati differential equations as well as for the remaining equations. The analytical approximations allow us to study the problem in terms of the parameters involved. Some numerical examples are also considered which show that exponential methods are, in general, superior to standard methods.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

We consider the numerical integration of non-autonomous linear-quadratic N -player differential games. Linear-quadratic N -player differential games can be considered as optimal linear control problems and have been extensively studied from the theoretical point of view [1–7]. Optimal linear control problems appear in many different fields in engineering [1,8,9] or in quantum mechanics (see [10] and references therein). The autonomous case has been extensively studied (the structure of the solution, the stability of the system, etc.) and different numerical methods exist. The non-autonomous case is more involved and the theoretical results are more limited. Standard numerical methods do not preserve the structure of the problems and then, important properties, like the stability of the system, can be seriously damaged by the numerical scheme.

In this work, we consider schemes based on the Magnus series expansion to solve non-autonomous problems and we show that they preserve most qualitative properties of the problem. The new methods can be seen as average techniques which transform non-autonomous problems into autonomous problems with the same structure (in terms of symmetric matrices, positive definite matrices, etc.) which can be then solved by methods designed for autonomous problems. The new schemes are usually referred as exponential methods due to its formal expansion, but can also be considered as time-averaging methods.

Optimal linear control problems can be described as systems of nonlinear boundary value problems (BVPs). Numerical methods for nonlinear BVPs are usually more involved (and computationally more costly) than IVPs. It is possible, however, to reformulate the problem in a way which requires the integration of non-autonomous coupled (quasi-)linear systems of differential equations forward and backward in time. There exist highly efficient methods for numerically solving non-autonomous linear systems which have been recently developed in the literature, providing qualitatively and quantitatively accurate results at a moderate computational cost. These methods are usually referred as geometric numerical integrators

* Corresponding author.

E-mail addresses: serblaza@imm.upv.es (S. Blanes), eponsoda@imm.upv.es (E. Ponsoda).

and, among them, the exponential methods showed a high performance for linear problems [11–14]. In some cases, the coupled quasi-linear systems can be solved sequentially as a set of linear equations.

The goal of this work is to analyze methods based on the Magnus series expansion adapted for solving the N -player linear quadratic differential game. For simplicity in the presentation, most of the analysis is presented for a two-player linear differential game defined by

$$x'(t) = A(t)x(t) + B_1(t)u_1(t) + B_2(t)u_2(t), \quad x(0) = x_0, \quad 0 = t_0 \leq t \leq t_f = T, \quad (1)$$

where the unknown $x(t) \in \mathbb{R}^n$ is the dynamic state of the game at the time t , that is influenced by both players. Here $A(t) \in \mathbb{R}^{n \times n}$, $B_i(t) \in \mathbb{R}^{n \times r_i}$, and the controls $u_i(t) \in \mathbb{R}^{r_i}$, $i = 1, 2$, are the vectors of variables that can be manipulated by each player, see [5,6] for details. The quadratic cost functions associated with the player i , [7], are given by

$$2J_i = x^T(T) Q_{iT} x(T) + \int_0^T \{x^T(t) Q_i(t)x(t) + u_i^T(t) R_{ii}(t) u_i(t) + u_j^T(t) R_{ij}(t) u_j(t)\} dt, \quad (2)$$

$j = 3 - i$, $i = 1, 2$, where $Q_i(t), Q_{iT} \in \mathbb{R}^{n \times n}$, $R_{ii}(t) \in \mathbb{R}^{r_i \times r_i}$, $R_{ij}(t) \in \mathbb{R}^{r_j \times r_j}$, are symmetric (and usually non-negative) matrices, R_{ij} is symmetric and positive definite, $i = 1, 2$, and z^T denotes the transpose of z .

From [7], for a zero sum game it is necessary that $R_{ij} \neq 0$, $i \neq j$, but in a non-zero sum game it is natural to choose $R_{ij} = 0$, $i \neq j$, because in the most frequent applications the cost function of each player does not contain the other player control. In this way, the quadratic cost function J_i depends only on the control u_i , $i = 1, 2$.

Let us consider for the moment a non-cooperative non-zero sum game where each player, in order to minimize their cost function, determines his action in an independent way knowing only the initial state of the game and the model structure. For the non-cooperative differential game, the open-loop Nash controls are given in [2,3,5,7,15]

$$u_i(t) = -R_{ii}^{-1}(t) B_i^T(t) P_i(t) x(t), \quad i = 1, 2, \quad (3)$$

with $P_i(t) \in \mathbb{R}^{n \times n}$, $i = 1, 2$, satisfying the coupled Riccati differential equations (RDEs)

$$\begin{aligned} P_1' &= -Q_1(t) - A^T(t)P_1 - P_1A(t) + P_1S_1(t)P_1 + P_1S_2(t)P_2, \\ P_2' &= -Q_2(t) - A^T(t)P_2 - P_2A(t) + P_2S_2(t)P_2 + P_2S_1(t)P_1, \end{aligned} \quad (4)$$

with the final conditions, $P_1(T) = Q_{1T}$, $P_2(T) = Q_{2T}$, wherein

$$S_i(t) = B_i(t) R_{ii}^{-1}(t) B_i^T(t); \quad i = 1, 2, \quad (5)$$

is a symmetric $n \times n$ matrix. Substituting (3) in (1) we have that

$$x(t) = \Phi(t, t_0) x_0, \quad (6)$$

where the fundamental matrix (or evolution operator) $\Phi(t, t_0)$ verifies the transition equation

$$\Phi'(t, t_0) = H(t, P(t)) \Phi(t, t_0); \quad \Phi(t, t) = I_n, \quad (7)$$

with

$$H(t, P(t)) = A(t) - S_1(t)P_1(t) - S_2(t)P_2(t), \quad (8)$$

where $S_i(t)$, $i = 1, 2$, are given by (5), $P_i(t)$, $i = 1, 2$, are the solutions of (4) and I_n denotes the identity matrix in $\mathbb{R}^{n \times n}$.

The system of Eqs. (4) jointly with (7) form a coupled system of nonlinear differential equations with boundary conditions. The numerical solution of nonlinear BVPs usually requires involved and computationally costly methods with significant storage requirements. The equations of this problem can be solved, however, sequentially as a set of IVPs.

To solve this problem requires then to consider the following steps:

- (i) First, to solve from $t = T$ to $t = 0$ the coupled non-autonomous RDEs (4), where $S_i(t)$, $i = 1, 2$, are given by (5).
- (ii) To use the approximate values obtained for $P_1(t)$, $P_2(t)$ computed on a mesh, and the explicit values of $A(t)$, $S_1(t)$, $S_2(t)$ to solve the transition Eqs. (7) and (8), from $t = 0$ to $t = T$. Then, from (6), the dynamic state of the game $x(t)$ is determined.
- (iii) Finally, one can obtain the open loop Nash controls using (3) as well as the quadratic cost functions (2).

In this work we consider several Magnus integrators which can be seen as exponential integrators from the family of Lie group integrators, for homogeneous linear ordinary differential equations. They can be used to solve the coupled equations while preserving the qualitative properties of the solution, [13,16,17]. The proposed schemes can be considered as average methods, and they are valid for linear and nonlinear problems.

We consider in detail the non-zero sum game where the RDE can be written as an equivalent non-autonomous linear differential equation to be solved by Magnus integrators, and next we extend the results to the zero sum case, where the coupled RDEs cannot be written in an equivalent linear form.

2. The coupled matrix Riccati differential equations

The coupled matrix RDEs (4) have been studied by several authors (see for example [3,5,7,8,15,18,19] and the references therein) and its interest for solving differential games is frequently mentioned. Extensive treatment of the RDE can be found in the literature, both in its theoretical aspect [9,20] and its numerical treatment [21–26]. However, in general, the numerical schemes proposed are valid only for autonomous problems, do not take into account the particular structure for some problems, or do not take into account the RDEs are coupled with another differential equation. We consider, however, the non-autonomous RDE (4) in the context of the full problem, from the fact that the solutions of this coupled system appear in the equations of the state vector (7) and (8). This motivates the search of methods for solving (4) to ensure a sufficient accuracy in order not to drag too many errors in the followings steps.

For solving the coupled RDEs (4), it is usual to consider the following decomposition $P_i U = V_i$, $i = 1, 2$, with $U(t)$, $V_i(t) \in \mathbb{R}^{n \times n}$. Let us denote

$$y(t) = \begin{bmatrix} U(t) \\ V_1(t) \\ V_2(t) \end{bmatrix}; \quad K(t) = \begin{bmatrix} A(t) & -S_1(t) & -S_2(t) \\ -Q_1(t) & -A^T(t) & 0 \\ -Q_2(t) & 0 & -A^T(t) \end{bmatrix},$$

where $y(t) \in \mathbb{R}^{3n \times n}$, $K(t) \in \mathbb{R}^{3n \times 3n}$, $S_i(t)$, $i = 1, 2$, are given by (5), and the matrices $A(t)$ and $Q_i(t)$, $i = 1, 2$, are given by (1) and (2), respectively. Then, it is easy to check that $y(t)$ is the solution of the IVP

$$y'(t) = K(t)y(t); \quad y(T) = \begin{bmatrix} U(T) \\ V_1(T) \\ V_2(T) \end{bmatrix} = \begin{bmatrix} I \\ Q_{1T} \\ Q_{2T} \end{bmatrix}, \tag{9}$$

with conditions at the end of the interval, to be integrated backward in time. In [5,19], if (9) has an appropriate solution with $U(t)$ non-singular, the solution of (4) can be calculated by

$$P_i(t) = V_i(t)U(t)^{-1}, \quad i = 1, 2. \tag{10}$$

Conditions under which U^{-1} exists are known (see [8,24] and references therein). In this work we assume $U(t)$ is non-singular, otherwise $P_i(t)$ would be unbounded and the equation for the state vector would not be well defined.

2.1. The N player case

For the general case with N players, the problem to be solved is

$$x'(t) = A(t)x(t) + \sum_{i=1}^N B_i(t)u_i(t), \quad x(0) = x_0,$$

$$2J_i(u_1, \dots, u_N) = x^T(T) Q_{iT} x(T) + \int_0^T \left\{ x^T(t) Q_i(t) x(t) + u_i^T(t) R_{ii}(t) u_i(t) + \sum_{j=1, j \neq i}^N u_j^T(t) R_{ij}(t) u_j(t) \right\} dt,$$

$i = 1, \dots, N$. In the same way as in the two-player's game, we consider a non-cooperative and non-zero sum game, with $R_{ij} = 0$, $i \neq j$, and then, J_i depends only of the control u_i . Under these conditions the controls are given by

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

In this case, the coupled Riccati equations are

$$P_i' = -Q_i(t) - A^T(t)P_i - P_i A(t) + \sum_{j=1}^N P_i S_j(t) P_j \tag{11}$$

with $S_i(t) = B_i(t) R_{ii}^{-1}(t) B_i^T(t)$, $i = 1, \dots, N$. If we denote by

$$W(t) = \begin{bmatrix} P_1(t) \\ \vdots \\ P_N(t) \end{bmatrix} \in \mathbb{R}^{Nn \times n}, \quad C(t) = \begin{bmatrix} -Q_1(t) \\ \vdots \\ -Q_N(t) \end{bmatrix} \in \mathbb{R}^{Nn \times Nn}, \tag{12}$$

$$B(t) = [-S_1(t) \dots -S_N(t)] \in \mathbb{R}^{n \times Nn} \tag{13}$$

$$D(t) = \begin{bmatrix} A^T(t) & 0 & \dots & 0 \\ 0 & A^T(t) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A^T(t) \end{bmatrix} \in \mathbb{R}^{Nn \times Nn}, \tag{14}$$

then, the coupled system (11) can be written as

$$W'(t) = C(t) - D(t)W(t) - W(t)A(t) - W(t)B(t)W(t).$$

From [9,24], if we consider $y(t) \in \mathbb{R}^{(N+1)n \times n}$ the solution of the linear equation

$$y'(t) = \begin{bmatrix} A(t) & B(t) \\ C(t) & -D(t) \end{bmatrix} y(t); \quad y(T) = \begin{bmatrix} U(T) \\ V_1(T) \\ \vdots \\ V_N(T) \end{bmatrix} = \begin{bmatrix} I \\ Q_{1T} \\ \vdots \\ Q_{NT} \end{bmatrix}, \tag{15}$$

we have that $P_i(t) = V_i(t) U(t)^{-1}$.

2.2. Optimal control for linear quadratic problems

To motivate the interest of geometric integrators to approximate the solution of this problem, let us first consider the RDE in its linear form for one-player

$$\begin{bmatrix} U' \\ V' \end{bmatrix} = \begin{bmatrix} A(t) & -S(t) \\ -Q(t) & -A^T(t) \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix}. \tag{16}$$

Since Q and S are symmetric, this matrix belongs to the algebra of symplectic matrices and has a Hamiltonian structure. For instance, the Eq. (16) can be originated from the Hamiltonian equations associated to the Hamiltonian

$$H = -\frac{1}{2}p^T S(t)p + p^T A(t)q + \frac{1}{2}q^T Q(t)q,$$

where $q, p \in \mathbb{R}^n$ are the coordinates and associated momenta and the Hamiltonian equations are given by

$$q' = \nabla_p H, \quad p' = -\nabla_q H,$$

which can be considered equivalent to (16) (the evolution matrices for both problems follow the same equation).

As an illustration, let us consider the simple 2×2 problem with $A, Q, S \in \mathbb{R}$ and $Q, S \geq 0$, then

$$\begin{bmatrix} U(t) \\ V(t) \end{bmatrix} = \begin{bmatrix} C_h - \frac{A}{\eta} S_h & \frac{S}{\eta} S_h \\ \frac{Q}{\eta} S_h & C_h + \frac{A}{\eta} S_h \end{bmatrix} \begin{bmatrix} 1 \\ Q_T \end{bmatrix},$$

where $\eta = \sqrt{A^2 + QS}$, $C_h = \cosh(\eta(T - t))$ and $S_h = \sinh(\eta(T - t))$. Then $U(t) > 0, \forall t$, and

$$\lim_{T \rightarrow \infty} V(t)U(t)^{-1} = \frac{Q_T(\eta + A) + Q}{\eta - A + SQ_T}$$

which gives a valuable information on the long time evolution of the problem. The qualitative behavior is similar if the problem is explicitly time-dependent with, for example, $\lim_{t \rightarrow \infty} A(t) = A, \lim_{t \rightarrow \infty} Q(t) = Q, \lim_{t \rightarrow \infty} S(t) = S$ and $Q(t), S(t) \geq 0$. The solution of the corresponding RDE converges to the previous constant. This last problem has no analytical solution and numerical methods are usually necessary. It is then natural to look for a numerical method which preserves the structure of the problem and has similar asymptotic behavior for long time integrations (notice that for the exact solution $U(t) > 0$, and then its inverse exists).

The numerical integration of (16) by most geometric integrators can be seen as the exact solution of a perturbed problem which preserves the symplectic structure, i.e. they are the exact solution of a perturbed problem where, e.g. S and Q are replaced by slightly perturbed symmetric matrices. However, in general, the positivity of these matrices is not guaranteed so, U can be singular leading to unbounded numerical solutions. We consider Magnus integrators and we show that some of them preserve both the symplectic structure as well as the positivity, leading to bounded solutions, as the exact solution.

3. The Magnus expansion

The numerical integration of non-autonomous linear differential equations using geometric integrators is frequently done by exponential integrators. In general, they preserve the Lie group structure of the exact solution and for this reason, they are also frequently referred as Lie group methods. Geometric integrators have shown to be superior, in general, to standard numerical methods for the integration of Hamiltonian systems [13,27].

We consider in some detail the Magnus series expansion [11,28] as well as some numerical Magnus integrators since for this problem they have shown a high performance while being relatively simple to implement.

Let us consider the linear homogeneous equation

$$Y'(t) = M(t)Y(t), \quad Y(t_0) = I_n, \tag{17}$$

with $Y(t), M(t) \in \mathbb{C}^{n \times n}$, and denote the fundamental matrix solution by $\Phi(t, t_0)$. If we consider that the solution can be written in the exponential form, $\Phi(t, t_0) = \exp(\Omega(t, t_0))$, where $\Omega(t, t_0) = \sum_{n=1}^{\infty} \Omega_n(t, t_0)$, we obtain the Magnus series expansion [28] where

$$\begin{aligned} \Omega_1(t, t_0) &= \int_{t_0}^t M(t_1) dt_1, \\ \Omega_2(t, t_0) &= \frac{1}{2} \int_{t_0}^t dt_1 \int_{t_0}^{t_1} dt_2 [M(t_1), M(t_2)], \dots \end{aligned} \tag{18}$$

Here $[A, B] \equiv AB - BA$ is the matrix commutator of A and B . We denote the truncated expansion by

$$\Psi^{[k]}(t, t_0) = \exp\left(\sum_{i=1}^k \Omega_i(t, t_0)\right). \tag{19}$$

The first order approximation agrees with the first order approximation for most exponential methods like e.g. the Fer [29] or Wilcox [30] expansions, but they differ at higher orders. In general, exponential methods (and most expansions) converge for

$$\int_{t_0}^t \|M(t_1)\| dt_1 < \xi, \tag{20}$$

where ξ is a constant depending on the method. For the Magnus expansion (ME), we have that, for the 2-norm, $\xi = \pi$, being this a sufficient but not necessary condition [31] (see also [11] and the references therein).

3.1. Magnus expansion for the non-homogeneous problem

Suppose the game can be influenced by external forces

$$x'(t) = A(t)x(t) + B_1(t)u_1(t) + B_2(t)u_2(t) + z(t), \quad x(0) = x_0,$$

where $z(t) \in \mathbb{R}^n$ is the uncertainty, see [32]. This problem can be solved using most results with minor modifications. Notice that this problem is equivalent to

$$y'(t) = \bar{A}(t)y(t) + \bar{B}_1(t)u_1(t) + \bar{B}_2(t)u_2(t), \quad y(0) = y_0,$$

where $y = [x, 1]^T \in \mathbb{R}^{n+1}$ and

$$\bar{A}(t) = \begin{bmatrix} A(t) & z(t) \\ 0 & 0 \end{bmatrix}, \quad \bar{B}_1(t) = \begin{bmatrix} B_1(t) \\ 0 \end{bmatrix}, \quad \bar{B}_2(t) = \begin{bmatrix} B_2(t) \\ 0 \end{bmatrix}.$$

Then, the methods previously considered can be used.

On the other hand, we have seen that a matrix RDE can be written as a linear system. However, coupled RDEs where $R_{ij} \neq 0$ do not admit this linear form. For this reason, it is convenient to consider the generalization of the ME to the nonlinear case.

3.2. The Magnus expansion for nonlinear systems

Let us consider the nonlinear and non-autonomous equation

$$z' = f(t, z), \quad z(t_0) = z_0 \in \mathbb{R}^n, \tag{21}$$

where the evolution operator ($z(t) = \Phi^t(z_0)$) satisfies the operational equation

$$\frac{d}{dt} \Phi^t = \Phi^t L_f(t, y), \quad y = z_0, \tag{22}$$

with solution $\Phi^t(y)|_{y=z_0}$. Here, $L_f = f \cdot \nabla_y$, with $f = [f_1, f_2, \dots, f_n]^T$, is the Lie derivative (or Lie operator) associated with f , acting on differentiable functions $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ as

$$L_f F(z) = F'(z)f(z),$$

where $F'(y)$ denotes the Jacobian matrix of F (see [11] for more details).

Since L_f is a linear operator, we can then use directly the Magnus series expansion to obtain the formal solution of (22) for $t \in [t_0, T]$ as $\Phi^T = \exp(L_{\omega(z_0)})$, with $\omega = \sum_i \omega_i$. The first two terms are now

$$\omega_1(z_0) = \int_{t_0}^T f(s, z_0) ds, \tag{23}$$

$$\omega_2(z_0) = -\frac{1}{2} \int_{t_0}^T ds_1 \int_{t_0}^{s_1} ds_2 (f(s_1, z_0), f(s_2, z_0)), \tag{24}$$

where (f, g) denotes the Lie bracket with components $(f, g)_i = f \cdot \nabla_z g_i(z) - g \cdot \nabla_z f_i(z)$. Observe that for the integrals (23) and (24) z_0 , on the vector field, is frozen (on the Lie derivatives, one has to compute the derivatives with respect to z and to replace them by z_0). If the series converges, the exact solution, $z(T) = \Phi^T(z_0)$ can be approximated by the 1-flow solution of the autonomous differential equation

$$y' = \omega^{[n]}(y), \quad y(0) = z_0, \tag{25}$$

$\omega^{[n]} = \sum_i^n \omega_i$, i.e. $y(1) \simeq z(T)$ (see also [11] and references therein for more details). Notice that the ME can be considered as an average on the explicitly time-dependent functions appearing on the vector field. It still remains to solve the autonomous problem (25). A simple method (useful when an efficient algorithm is known for the problem (21) when the time is frozen) will be presented.

Example 1. As an illustrative example, let us consider the scalar nonlinear and non-autonomous equation

$$z' = \alpha(t)z^k + \beta(t)z^m, \quad z(t_0) = z_0,$$

then Eqs. (23) and (24) provide

$$\omega_1(z_0) = a z_0^k + b z_0^m, \quad \omega_2(z_0) = c z_0^{k+m-1},$$

where the constants a, b, c are given by

$$a = \int_{t_0}^T \alpha(s) ds, \quad b = \int_{t_0}^T \beta(s) ds,$$

$$c = -\frac{m-k}{2} \int_{t_0}^T ds_1 \int_{t_0}^{s_1} ds_2 (\alpha(s_1)\beta(s_2) - \beta(s_1)\alpha(s_2)).$$

Then, the solution, $y(1)$, of the autonomous (solvable) equation

$$y' = ay^k + by^m + c y^{k+m-1}, \quad y(t_0) = z_0,$$

is the approximation to $z(T)$ by the second order Magnus expansion. If $k = 1, m = 2$ this is a RDE and the Magnus approximation can be seen as the exact solution of another autonomous RDE where the time-dependent function $\alpha(t)$ is replaced by a and $\beta(t)$ by $b + c$.

3.3. Magnus integrators

In those cases where it is not feasible to obtain analytical solutions, the numerical integration can be the choice. The time interval is divided on a mesh, $t_0 < t_1 < \dots < t_{N-1} < t_N = T$, and an appropriate scheme is used on each interval. For simplicity in the presentation, we consider a constant time step, $h = (T - t_0)/N$ and $t_n = t_0 + nh, n = 0, 1, \dots, N$. In the following, we consider Magnus integrators. To this purpose, it is useful to notice that for the truncated expansion (19)

$$\Psi^{[1]}(t+h, t) = \Phi(t+h, t) + \mathcal{O}(h^3), \quad \Psi^{[2]}(t+h, t) = \Phi(t+h, t) + \mathcal{O}(h^5).$$

If the integrals in (18) cannot be computed analytically, we can approximate them by a quadrature rule, and this can be done by computing the matrix $M(t)$ on the points of a single quadrature rule [14] (see also [11,12]).

In some cases, the matrix $M(t)$ is explicitly known, but in some other cases it is only known on a mesh. Then, in order to present methods which can be easily adapted for different quadrature rules we introduce the averaged (or generalized momentum) matrices

$$M^{(i)}(h) \equiv \frac{1}{h^i} \int_{t_n}^{t_n+h} (t - t_{1/2})^i M(t) dt = \frac{1}{h^i} \int_{-h/2}^{h/2} t^i M(t + t_{1/2}) dt, \tag{26}$$

for $i = 0, 1, \dots$ with $t_{1/2} = t_n + h/2$. Then, it is clear that $\Psi^{[1]}(t+h, t) = e^{M^{(0)}(h)}$ for the second order method, but fourth-order methods can be obtained by taking

$$\Psi_1^{[2]}(t+h, t) = \exp(M^{(0)}(h) + [M^{(1)}(h), M^{(0)}(h)]). \tag{27}$$

In some cases, the commutator of sparse matrices can lead to filled matrices which are computationally costly to exponentiate. For this reason it is convenient to consider Magnus integrators which do not contain commutators. In this way, a simple fourth-order commutator-free method is given in [16,17]

$$\Psi_2^{[2]}(t+h, t) = \exp\left(\frac{1}{2}M^{(0)}(h) + 2M^{(1)}(h)\right) \exp\left(\frac{1}{2}M^{(0)}(h) - 2M^{(1)}(h)\right). \tag{28}$$

This method has shown to be highly efficient in practice. In addition, the simple structure of the method allows to prove the scheme can preserve relevant qualitative properties of the exact solution.

4. Structure preservation of the Magnus integrators

Magnus integrators can be considered as the exact solutions, for a given time interval, of autonomous problems whose global flow agrees with the exact one up to a given order of accuracy. This interpretation allows to show that most qualitative properties of the exact flow are retained by Magnus integrators

Lemma 2. *If $M(t)$ is a continuous bounded symmetric positive definite matrix for $t \in [t_n, t_n + h]$ then $M^{(0)}(h)$ and $M_{\pm}(h) \equiv \frac{1}{2}M^{(0)}(h) \pm 2M^{(1)}(h)$ are also symmetric and positive definite matrices for a sufficiently small value of h .*

Proof. The preservation of the symmetry and that $M^{(0)}(h)$ is a positive definite matrix is obvious. On the other hand, we have that for a time step, h

$$M_{\pm}(h) \equiv \frac{1}{2}M^{(0)}(h) \pm 2M^{(1)}(h) = \int_{-h/2}^{h/2} \rho(\tau)M(\tau + t_{1/2}) d\tau,$$

with $\rho_{\pm}(\tau) = \frac{1}{2} \pm 2\tau/h$, which is a linear function such that

$$\rho_{\pm} : [-h/2, h/2] \longrightarrow [-1/2, 3/2].$$

Since $M(t)$ is continuous and bounded, $M_{\pm}(h) = \frac{h}{2}M(t_n+h/2) + \mathcal{O}(h^2)$, which is also a positive definite matrix for sufficiently small h . \square

This result allows to show that the fourth-order commutator-free method (28) will preserve relevant qualitative properties in many cases.

There exist higher order commutator-free Magnus integrators given by the general composition

$$\prod_{i=1}^m e^{M^{[i]}(h)},$$

with $M^{[i]}(h)$ given by linear combinations of the momentum matrices (26) or the matrices $M(t)$ computed on a quadrature rule [33,34]. We have observed that for all methods of order greater than four exists j , such that $M^{[j]}(h) = \alpha_j h M(t_n + h/2) + \mathcal{O}(h^2)$ with $\alpha_j < 0$. Then, this lemma does not extend to the existing higher order methods. It remains as an open question if the highest order where this lemma applies, is four.

If $M(t)$ is a piecewise constant symmetric positive definite matrix, one can always split the time interval such that $M(t)$ is continuous on each interval, where the lemma applies.

This scheme allows to guarantee the structure and positive definiteness of the matrices to be integrated and then one can expect good stability properties for long time integrations.

Notice that the Magnus integrators applied to (16) can be considered as the exact solution, for a given time interval, to the autonomous problem

$$\begin{bmatrix} U' \\ V' \end{bmatrix} = \begin{bmatrix} \bar{A} & -\bar{S} \\ -\bar{Q} & -\bar{A}^T \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix}. \tag{29}$$

The first order Magnus integrator (of second order in the time step h) corresponds to the case

$$\bar{A} = A^{(0)}, \quad \bar{Q} = Q^{(0)}, \quad \bar{S} = S^{(0)},$$

while the fourth order approximation (27) corresponds to

$$\begin{aligned} \bar{A} &= A^{(0)} + [A^{(1)}, A^{(0)}] + S^{(1)}Q^{(0)} - S^{(0)}Q^{(1)}, \\ \bar{Q} &= Q^{(0)} + (Q^{(1)}A^{(0)} + A^{(0)T}Q^{(1)}) + (Q^{(0)}A^{(1)} + A^{(1)T}Q^{(1)}), \end{aligned}$$

and similarly for \bar{S} . If $\bar{\Phi}(t, t_0)$ denotes the evolution operator of the averaged Eq. (29) it is clear that

$$\det \Phi(t, t_0) = \det \bar{\Phi}(t, t_0).$$

Obviously, \bar{Q} and \bar{S} are still symmetric matrices and, if $Q(t), S(t)$ are also positive definite, \bar{Q}, \bar{S} will share this property for sufficiently small time steps. Notice that for the fourth-order commutator-free methods, this result was independent of the matrix $A(t)$.

The fourth-order commutator free Magnus approximation (28) admits a simpler interpretation. It corresponds, for one time step, h , to take the composition of flows given by

$$\begin{bmatrix} U_1' \\ V_1' \end{bmatrix} = \begin{bmatrix} \bar{A}_1 & -\bar{S}_1 \\ -\bar{Q}_1 & -\bar{A}_1^T \end{bmatrix} \begin{bmatrix} U_1 \\ V_1 \end{bmatrix}, \quad \begin{bmatrix} U_1(0) \\ V_1(0) \end{bmatrix} = \begin{bmatrix} U(t_0) \\ V(t_0) \end{bmatrix},$$

$$\begin{bmatrix} U_2' \\ V_2' \end{bmatrix} = \begin{bmatrix} \bar{A}_2 & -\bar{S}_2 \\ -\bar{Q}_2 & -\bar{A}_2^T \end{bmatrix} \begin{bmatrix} U_2 \\ V_2 \end{bmatrix}, \quad \begin{bmatrix} U_2(0) \\ V_2(0) \end{bmatrix} = \begin{bmatrix} U_1(1) \\ V_1(1) \end{bmatrix},$$

where

$$\bar{A}_1 = \frac{1}{2}A^{(0)} - 2A^{(1)}, \quad \bar{A}_2 = \frac{1}{2}A^{(0)} + 2A^{(1)},$$

and similarly for $\bar{Q}_i, \bar{S}_i, i = 1, 2$. Here $[U(t_0 + h), V(t_0 + h)]^T \simeq [U_2(1), V_2(1)]^T$.

If $Q(t)$ and $S(t)$ are symmetric and positive definite, Lemma 2 shows that, for sufficiently small time steps, $\bar{Q}_i, \bar{S}_i, i = 1, 2$, will also be symmetric and positive definite matrices. The solution of the RDE is approximated by a sequence of piecewise constant RDEs where the symmetry and positive definiteness of the solution is then guaranteed.

We also expect a high performance of the Magnus integrators for the N -player case with $N > 1$. These methods have shown to be highly accurate and stable, and this last fact can be of great interest because the storage requirements can also play an important role in this problem.

4.1. Magnus integrators for the zero sum game

From [7], for a zero sum game it is necessary that $R_{ij} \neq 0, i \neq j$. In this way, the quadratic cost functions J_i depend on both controls $u_i, i = 1, 2$. The coupled Riccati differential equations to be solved are

$$P_1' = -Q_1(t) - A^T(t)P_1 - P_1A(t) + P_1S_1(t)P_1 + P_1S_2(t)P_2 + P_2S_{22}(t)P_2,$$

$$P_2' = -Q_2(t) - A^T(t)P_2 - P_2A(t) + P_2S_2(t)P_2 + P_2S_1(t)P_1 + P_1S_{11}(t)P_1,$$

with the final conditions, $P_1(T) = Q_{1T}, P_2(T) = Q_{2T}$, wherein

$$S_{ij}(t) = B_i(t) R_{ij}^{-1}(t) B_j^T(t); \quad i = 1, 2.$$

This problem cannot be reformulated as a linear problem. It is still possible, however, to use Magnus methods to obtain an averaged equivalent problem (up to a given order of accuracy). To this goal, we have to consider the generalization of the Magnus expansion to nonlinear problems given in Section 3.2. The standard Magnus expansion requires the computation of Lie brackets, and this can be rather involved for this problem. To simplify this problem, we can use commutator-free methods where the interpretation seems more clear. They are equivalent to solve, sequentially, problems given by

$$P_{1,k}' = -Q_{1,k} - A_k^T P_{1,k} - P_{1,k} A_k + P_{1,k} S_{1,k} P_{1,k} + P_{1,k} S_{2,k} P_{2,k} + P_{2,k} S_{22,k} P_{2,k},$$

$$P_{2,k}' = -Q_{2,k} - A_k^T P_{2,k} - P_{2,k} A_k + P_{2,k} S_{2,k} P_{2,k} + P_{2,k} S_{1,k} P_{1,k} + P_{1,k} S_{11,k} P_{1,k},$$

$k = 1, 2, \dots, K$, and the initial conditions for $k = i + 1$ correspond to the final conditions obtained on the previous stage $k = i$ (this is a composition of flows). For example, the first order Magnus approximation corresponds to the case where $K = 1$ and

$$A_1 = A^{(0)}, \quad Q_{1,1} = Q_1^{(0)}, \dots$$

and the fourth order commutator-free method corresponds to $K = 2$ where

$$A_1 = \frac{1}{2}A^{(0)} - 2A^{(1)}, \quad Q_{1,1} = \frac{1}{2}Q_1^{(0)} - 2Q_1^{(1)}, \dots$$

$$A_2 = \frac{1}{2}A^{(0)} + 2A^{(1)}, \quad Q_{1,2} = \frac{1}{2}Q_1^{(0)} + 2Q_1^{(1)}, \dots$$

where one has to solve the problem first for $k = 1$ with final conditions $P_{1,1}(t_f) = P_{1,f}, P_{2,1}(t_f) = P_{2,f}$ for one backward time step, (1-flow), followed by the problem with $k = 2$ and final conditions $P_{1,2}(t_f) = P_{1,1}(1), P_{2,2}(t_f) = P_{2,1}(1)$.

5. On the implementation of the algorithms

5.1. The choice of the quadrature rules

If the integrals in (26) have no analytical solution, we can replace them by quadrature rules of the corresponding order. Since we are considering coupled problems, we found appropriate to use the Newton–Côtes quadrature rules. Then, the trapezoidal rule for a second order scheme

$$\Psi^{[2]}(t+h, t) \equiv \exp\left(\frac{h}{2}(M(t+h) + M(t))\right) = \Phi(t+h, t) + \mathcal{O}(h^3), \tag{30}$$

is more appropriate than the midpoint rule. Notice that the midpoint rule provides $Y(t_n)$, but we have computed $M(t_n + h/2)$ and they can be the inputs for another equation, e.g. Eq. (7). We can obtain a fourth-order scheme from (27) by taking a fourth-order quadrature rule to approximate the integrals. As mentioned, it is convenient, from the computational point of view, to use evaluations of M at equispaced points. For example, using the Simpson rule we have:

$$M^{(0)}(h) \cong \frac{h}{6}(M_1 + 4M_2 + M_3), \quad M^{(1)}(h) \cong \frac{h}{12}(M_3 - M_1), \tag{31}$$

where $M_1 = M(t_n)$, $M_2 = M(t_n + \frac{h}{2})$, $M_3 = M(t_n + h)$. Notice that only two evaluations per step of $M(t)$ are necessary, since M_3 can be reused at the next integration step. In addition, using this Simpson rule, the matrix $M(t)$ is computed on a mesh which intersects with the mesh where the solutions Y_i are approximated.

On the other hand, for the N -player problem (12)–(15) when N is relatively large, the corresponding matrix $K(t)$ is sparse, and in many problems the computation of the exponential of a sparse matrix acting on a vector can be done efficiently using, e.g. Krylov spaces [35–37] or just a Taylor approximation. In practice, we suggest to consider the commutator-free fourth-order Magnus integrator (28) with (31) [11,16]

$$\Psi_2^{[4]}(t+h, t) \equiv \exp\left(\frac{h}{12}(-M_1 + 4M_2 + 3M_3)\right) \exp\left(\frac{h}{12}(3M_1 + 4M_2 - M_3)\right) \tag{32}$$

(a linear combination of sparse matrices is a sparse matrix with the same non-zero elements). To solve the whole problem, we have found it can be more efficient in some cases to use a given method for the RDEs and a different one to approximate the evolution operator for the state vector. For example, as we will see in the numerical examples, we found, in some cases with slowly varying matrices, more convenient to solve the RDEs using a second order method and to use a higher order method for the fundamental matrix associated to the state vector.

Obviously, the performance of the schemes proposed depend on the algorithm used to compute the matrix exponentials, or their action on vectors. We recommend [36,38,39], where different numerical algorithms are extensively treated. An analysis of the exponential of matrices in Lie algebras is given in [22], and an extensive software package for computing the matrix exponential is given in [37].

5.2. To solve the state equation

The dynamic state of the game, $x(t)$, can be obtained by solving the equation for the fundamental matrix (7) with

$$H(t, P(t)) = A(t) - \sum_{i=1}^N S_i(t)P_i(t).$$

For the time integration along the interval $[t_i, t_i + h]$, a numerical scheme requires to evaluate $H(t, P(t))$ at a set of nodes, $H(t_i^{(j)}, P(t_i^{(j)}))$ with $t_i^{(j)} = t_i + c_j h$, $j = 1, \dots, k$, and this has to be repeated for $i = 1, \dots, J$, where, for simplicity, we consider a constant time step, $h = T/J$. Then, the numerical integration of (15) must provide outputs at these nodes, $t_i^{(j)}$, to obtain $P_i^{(j)} = (P_{1,i}^{(j)}, \dots, P_{N,i}^{(j)})$ which are approximations to $P_m(t_i^{(j)})$, $m = 1, \dots, N$, $j = 1, \dots, k$, $i = 1, \dots, J$. This requires to compute and store a set of $k \times J \times N$ matrices of dimension $n \times n$. This is exceedingly costly for moderate values of n, N, J, k , and an alternative procedure seems necessary. It is possible, however, to consider numerical schemes which reduce drastically the storage requirements to just J matrices, corresponding to the fundamental matrices at the mesh points, t_i . To this purpose, we proceed as follows. Consider the well known property

$$\Phi(b, a) = \Phi(b, c) \Phi(c, a).$$

Then, to compute the approximations to the matrices $\Phi(t_i, t_0)$ we consider that

$$\Phi(t_i, t_0) = \Phi(t_i, T) \Phi(T, t_0) = \Phi(t_i, T) (\Phi(t_0, T))^{-1}.$$

This property allows us to run the algorithm for the Riccati equation jointly with the equation for the evolution matrix $\Phi(t_i, T)$ backward in time, in the same loop. We store these matrices and, at the end of the loop, the inverse $(\Phi(t_0, T))^{-1}$

has to be computed. In addition, the matrices $A(t)$, $S_i(t)$ used to solve the RDEs for P_i can be directly used, and they have not to be stored.

There is also an alternative procedure which corresponds to compute and store the matrices $H(t, P(t))$ at the nodes to be used to approximate $\Phi(t_{i+1}, t_i)$, or preferably, to be used to compute the recursive algorithm

$$x_{i+1} = \Phi(t_{i+1}, t_i)x_i.$$

This can be convenient in those cases where the computation of $\Phi(t_{i+1}, t_i)$ by exponentiation of matrices can be exceedingly costly, but an approximation to its action on a vector can be efficiently done [35,36].

Remark. To carry the full integration backward in time allows to use, in a simple way, variable time steps. The Magnus methods can also be used in this case [12]. In addition, it is worth to mention that, since the schemes proposed above are time-symmetric, they are also suitable to be used with standard extrapolation to rise their order.

6. Examples

Example 3. Let us consider a simple pursuit-evasion problem in space where the acceleration u_1 of the pursuer is controlled by the first player and the acceleration u_2 of the evader is controlled by the second player, see [7,40]. Let $r(t)$ be the position vector of the player one with respect the player two. Then

$$r'(t) = v(t); \quad v'(t) = u_1(t) - u_2(t), \tag{33}$$

for $t \in [0, T]$. We take $T = 1$ (a long time integration can also be considered which, after a proper scaling of the time, it would be equivalent to change the parameters of the problem). If we denote $x(t) = [r(t), v(t)]^T$, then the Eqs. (33) can be written in the form

$$x'(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_1(t) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} u_2(t); \quad x(0) = \begin{bmatrix} r(0) \\ v(0) \end{bmatrix} = x_0. \tag{34}$$

The cost functions are

$$2J_1 = x^T(1) \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} x(1) + \int_0^1 \frac{1}{c} u_1^T(t) u_1(t) dt,$$

$$2J_2 = x^T(1) \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} x(1) + \int_0^1 c u_2^T(t) u_2(t) dt,$$

where $c > 1$ is a real constant. Solving the Eq. (9) for K , it follows that

$$U(t) = \begin{bmatrix} 1 + \frac{1}{6} \left(c - \frac{1}{c}\right) (t-1)^3 & t-1 \\ \frac{1}{2} \left(c - \frac{1}{c}\right) (t-1)^2 & 1 \end{bmatrix}, \quad V_1(t) = \begin{bmatrix} 1 & 0 \\ 1-t & 0 \end{bmatrix},$$

and $V_2(t) = -V_1(t)$. Note that $\det U(t) = 1 + \frac{1}{3} \left(c - \frac{1}{c}\right) (1-t)^3 > 1$ under the condition $c > 1$, because $0 \leq t \leq 1$, and then this condition guarantees the existence of $U^{-1}(t)$ for all $t \in [0, 1]$. From (10)

$$P_1(t) = \frac{1}{\omega(t)} \begin{bmatrix} 1 & 1-t \\ 1-t & (1-t)^2 \end{bmatrix}, \quad P_2(t) = -P_1(t),$$

with $\omega(t) = 1 + \frac{1}{3} \left(c - \frac{1}{c}\right) (1-t)^3$, and the transition problem (7) can be written as follows

$$\Phi'(t) = H(t, P(t)) \Phi(t), \quad H = \begin{bmatrix} 0 & 1 \\ -\frac{(c - \frac{1}{c})(1-t)}{\omega(t)} & -\frac{(c - \frac{1}{c})(1-t)^2}{\omega(t)} \end{bmatrix}, \tag{35}$$

with $\Phi(0) = I$. Solving (35) and using (6), we can obtain the dynamic state of the game (34), and then, determine the controls. The Eq. (35) has no analytical solution, and Magnus integrators can provide analytical approximations.

If the parameter c is explicitly time-dependent, say $c(t)$, Magnus integrators can provide analytical approximations to $P_1(t)$ and $P_2(t)$ which are also symmetric non-negative and non-positive, respectively. In this case, we can use, for example, the first or second order approximations of the Magnus series expansion (18).

We can measure $\mathcal{E}^{[1]}(t) = \|H_{ex}(t) - H^{[1]}(t)\|$, where $H_{ex}(t)$ is the exact solution and $H^{[1]}(t, P(t))$ is obtained by considering the first order Magnus approximation to obtain $P_1(t)$ and $P_2(t)$, for different choices of the function, $c(t)$. In the

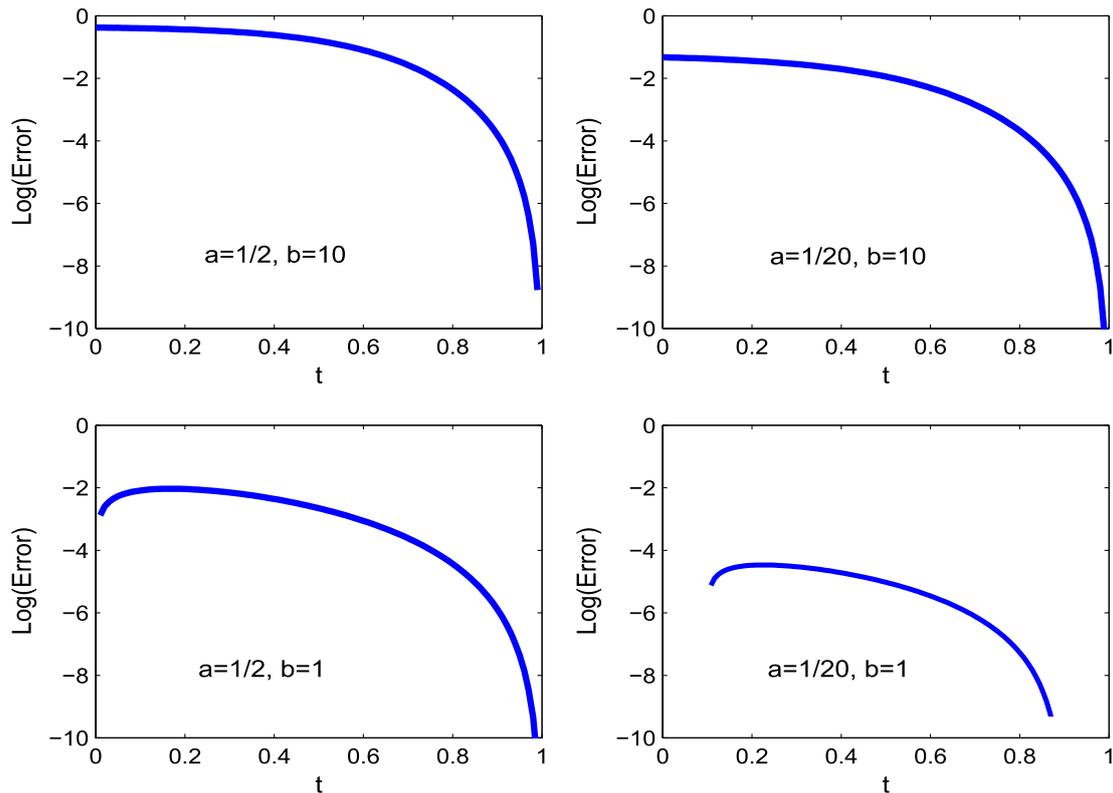


Fig. 1. Error $\mathcal{E}^{[1]}(t) = \|H_{ex}(t) - H^{[1]}(t)\|$ along the interval $t \in [0, 1]$ for the choice $c(t) = b e^{at}$ and for different values of the parameters a and b .

following we analyze the case: $c = b e^{at}$ for $b \geq 1, a > 0$. In Fig. 1 we show the error $\mathcal{E}^{[1]}(t)$ (we take $\|A\| = (\sum_{i,j} A_{i,j}^2)^{1/2}$) along the interval $t \in [0, 1]$ for different values of the parameters a and b .

Notice that, since $c(t) \geq 1$ we can take a norm such that $\|K(t)\| = c(t)$ and for a small value of a we have that $\int_{t_0}^t \|K(t_1)\| dt_1 \sim b$, and we still have accurate results for $b > \pi$ where the (sufficient but not necessary) convergence condition (20) is not satisfied. We can also observe that the accuracy increases for small values of a , i.e. when $c(t)$ is close to a constant (in this limit, the ME converges to the exact solution). Similar results are obtained taking for $c(t)$ other smooth functions.

Example 4. In [41] a problem of air pollutant emissions from two regions is proposed

$$x'(t) = -ax(t) + bu_1(t) + bu_2(t); \quad x(0) = x_0. \tag{36}$$

Here $x(t)$ is the excess of the pollutant in the atmosphere, $u_i(t), i = 1, 2$, denotes the emissions of each region and a, b are positive constants related with the intervention of the nature on environment. The cost functions to minimize are given by

$$J_i = \frac{1}{2} \int_0^T e^{-\rho t} \{c_i u_i^2(t) + d_i x^2(t)\} dt, \quad i = 1, 2, \tag{37}$$

where c_i, d_i , are positive parameters related to the costs of emission and pollution withstand respectively, and ρ is a refresh rate. With an appropriate change of constants, the problem (36) and (37) can be applied to other situations, such as financial problems for example, see [4].

In our notation, $Q_{iT} = 0, i = 1, 2, R_{ij} = 0, i \neq j, Q_i(t) = d_i e^{-\rho t}, R_{ii} = c_i e^{-\rho t}$, and $S_i = b^2 e^{\rho t} / c_i, i = 1, 2$.

We generalize this problem to the case of N regions (N players) where the Eq. (36) changes to

$$x'(t) = -ax(t) + b \sum_{i=1}^N u_i(t); \quad x(0) = x_0,$$

and the index for the remaining parameters range for $i = 1, \dots, N$. Then, following the steps indicated at the end of Section 2.1, we have:

- the Eq. (15) with the data

$$y'(t) = K(t)y(t), \quad y(T) = [1, 0, \dots, 0]^T;$$

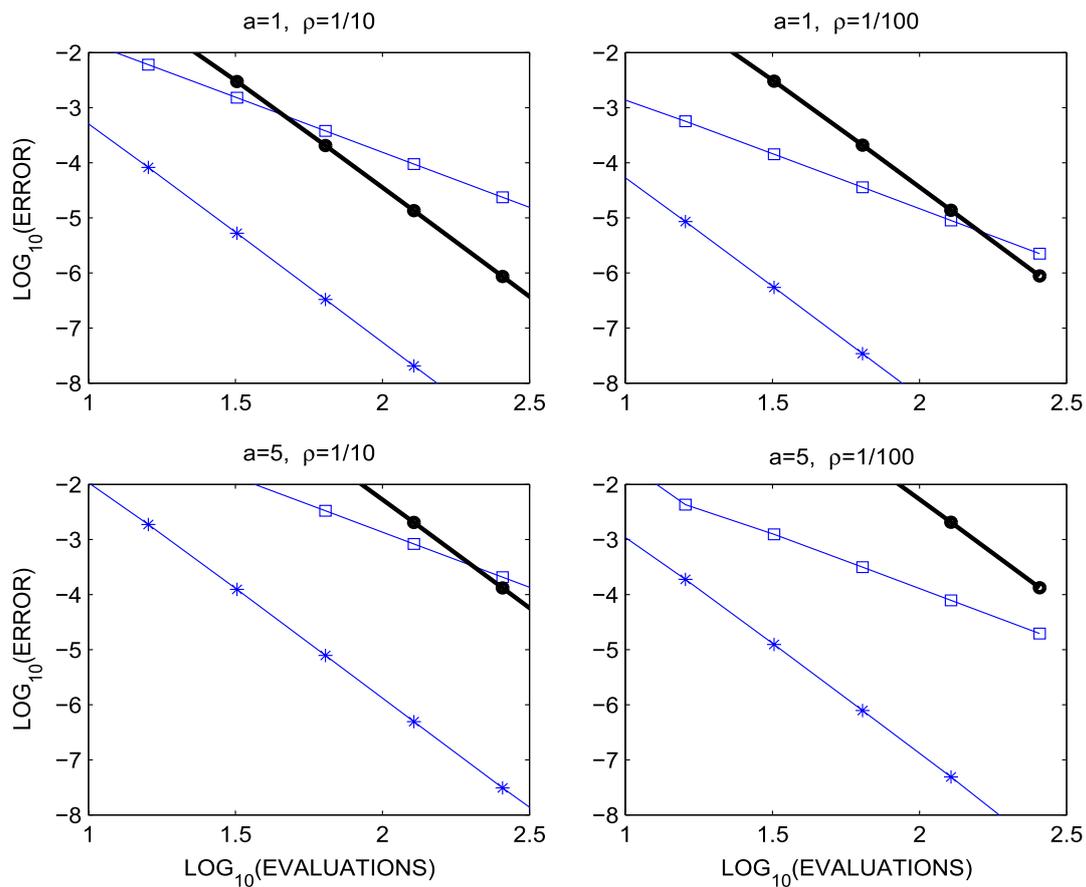


Fig. 2. Error $\|y(0) - y_{ap}(0)\|$ versus the number of evaluations of the matrix $K(t)$ (twice the number of steps) in solving the Riccati equation (38) using: the standard fourth-order Runge–Kutta method (circles joined by lines), the commutator-free fourth-order exponential method (stars joined by lines), and the second order exponential method (squares joined by lines).

$$y(t) = \begin{bmatrix} u(t) \\ v_1(t) \\ \vdots \\ v_N(t) \end{bmatrix}; \quad K(t) = \begin{bmatrix} -a & -\frac{b^2}{c_1}e^{\rho t} & \dots & -\frac{b^2}{c_N}e^{\rho t} \\ -d_1e^{-\rho t} & a & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -d_Ne^{-\rho t} & 0 & \dots & a \end{bmatrix}, \quad (38)$$

that must be solved from $t = T$ to $t = 0$.

- With $u(t)$ and $v_i(t)$ obtained in the previous step, we evaluate

$$p_i(t) = \frac{v_i(t)}{u(t)}, \quad i = 1, \dots, N,$$

when $u(t) \neq 0$.

- Solve the transition equation

$$\phi'(t) = \left(-a - b^2 e^{\rho t} \sum_{i=1}^N \frac{1}{c_i} p_i(t) \right) \phi(t), \quad \phi(0) = 1, \quad (39)$$

and calculate the dynamic state of the game: $x(t) = \phi(t) x_0$.

- Finally, we can determine the controls

$$u_i(t) = -\frac{1}{c_i} e^{\rho t} b p_i(t) \phi(t) x_0, \quad i = 1, \dots, N.$$

In our numerical experiments we take $T = 1$ and consider the case of $N = 10$ players, and the parameters used are: $c_i = i/2$, $d_i = 2/i$, $i = 1, \dots, 10$, $b = 3/2$, and we take different values for a and ρ .

As a standard numerical method we take the well known 4-stage fourth-order Runge–Kutta method. For the linear non-autonomous problem (17), this method requires only two evaluations of the matrix $K(t)$, and in this sense it is optimal.

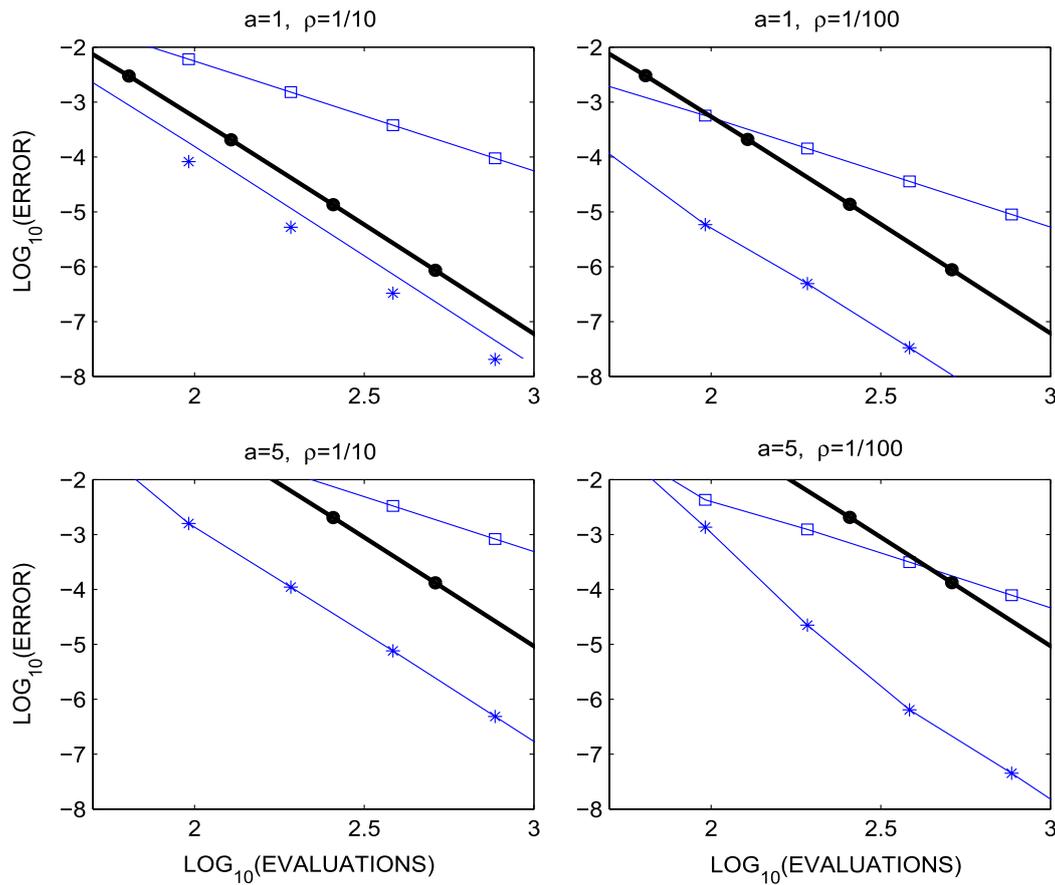


Fig. 3. Error $\|y(0) - y_{ap}(0)\|$ versus the number of vector-matrix multiplications. The exponentials are approximated by the sixth-order Taylor method and each exponential is counted as six vector-matrix products.

Another important reason is that the method requires evaluations of the matrices at equispaced time intervals. Then, we can solve (38) using a time step, h , to obtain the values of $p_i(t)$, $i = 1, \dots, N$, in an equispaced mesh, and then to use the same method for solving (39) with a time step twice larger.

We have also used the fourth-order commutator-free exponential method (32) using the same time steps as in the Runge–Kutta method, and are used for solving (39) with a time step twice larger. We have also computed (38) using the second order exponential method (30) which requires half number of evaluations of the matrix $K(t)$.

First, we measure the error in solving the linear equations associated to the Riccati equation and compute $\|y(0) - y_{ap}(0)\|$, where y_{ap} is the numerical solution obtained at the end of the backward integration. Fig. 2 illustrates the results obtained for $a = 1, 5$, and $\rho = 1/10, 1/100$. We show the results for the Runge–Kutta method (circles joined by lines), the fourth-order commutator-free exponential method (stars joined by lines), and the second order exponential method (squares joined by lines). We compare the error versus the number of evaluations of the matrix $K(t)$ or equivalently the number of steps which is twice this number. This way to measure the cost is in favor of the exponential methods because in general for the same time step, they are computationally more costly.

Since the computational cost of a method highly depends on the problem, we have repeated the same experiment with a different measure of the cost. For large systems, it is useful to consider efficient algorithm to approximate the exponential of a matrix acting on a vector. For this problem, the matrix $K(t)$ is sparse (the $(N + 1) \times (N + 1)$ matrix has only $3N + 1$ non-zero entries and a vector-matrix multiplication requires only this number of products). The exponents in the commutator-free methods have the same non-zero entries as $K(t)$ and the action of each exponential on a vector can be approximated by, for example, polynomial approximations like Krylov methods [35]. Just for illustration, we present in Fig. 3 the results obtained when each exponential is approximated by a sixth-order Taylor approximation, and the cost (EVALUATIONS) for each exponential is counted as six vector-matrix products. This measure, however, is in favor of the Runge–Kutta method because it needs to keep more vectors in memory and the number of evaluations of the matrix $K(t)$ is three times the number of evaluations for the Magnus integrators in these examples. The exponential still show a clear superiority.

Next, we take as initial conditions $x(0) = 1$ and measure the error $|x(1) - x_{ap}(1)|$ where the values of $p_i(t)$, $i = 1, \dots, 10$, are obtained from the previous computations. In Fig. 4 we show the results for the Runge–Kutta method, the fourth-order exponential method for solving (38) and (39), and the second order exponential method for solving (38) while the fourth-order exponential method is used for solving (39) (squares joined by lines). In this case, the cost is measured as the number of evaluations of $H(t, P(t))$, which is the most costly part for this problem.

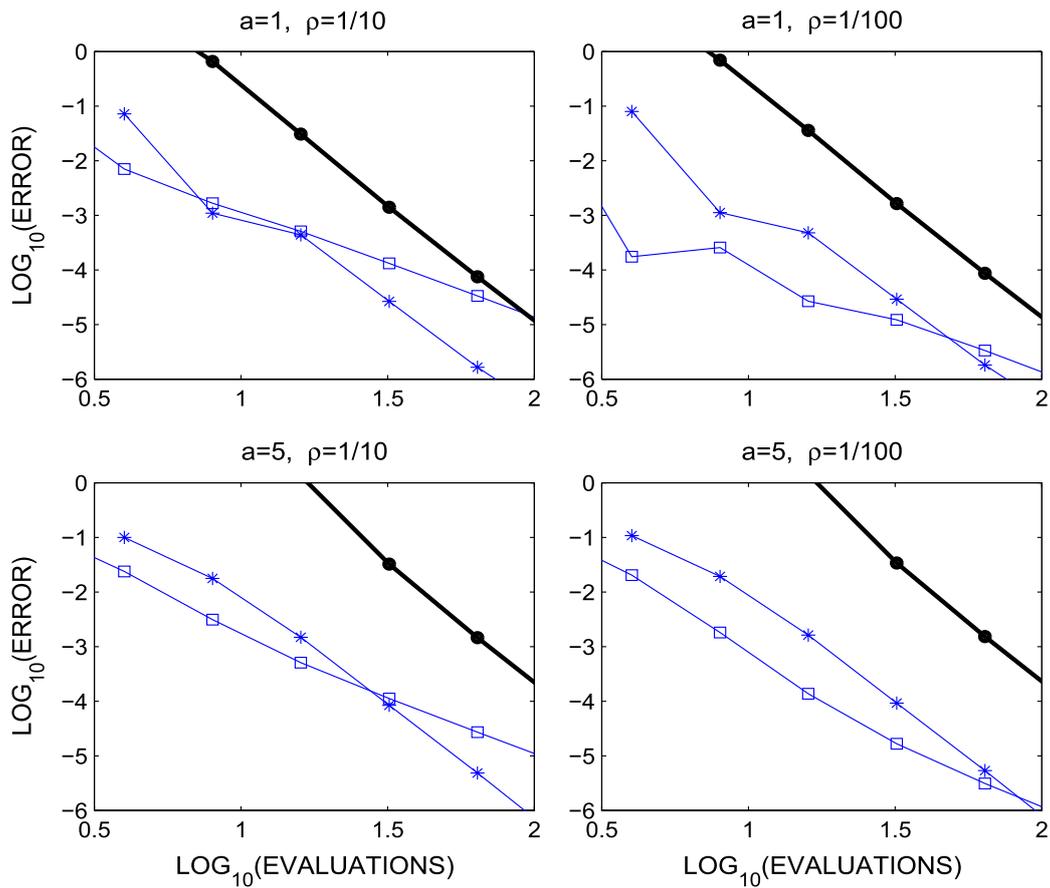


Fig. 4. Error $|x(1) - x_{ap}(1)|$ versus the number of evaluations of the matrix $K(t)$ in solving the equation for the state vector (39) using: the Runge–Kutta method, the fourth-order exponential method for solving (38) and (39), and the second order exponential method for solving (38) and while the fourth-order exponential method is used for solving (39). The curves are labeled as in the previous figure.

In all cases the exponential integrators showed a clear superiority versus the Runge–Kutta method. On the other hand, we can also see that, for relatively low accuracies and when the time-dependence of the linear matrix associated to the Riccati equations is smooth, the second order method for solving it gives enough accuracy at a lower computational cost, and then shows the highest performance for the computation of $x(t)$. We have carried out this numerical experiment for different number of players and similar results on the relative performance between exponential methods and the Runge–Kutta method were obtained.

7. Conclusions

Magnus integrators for solving N -player linear quadratic differential games are applied in this paper. The open loop Nash strategies for a non-cooperative and non-zero sum is considered in detail, and then, two initial value problems must be integrated numerically forward and backward in time: a matrix Riccati differential equation with final conditions, and a transition problem with initial value condition. Since the coupled system of matrix RDE can be reformulated as a linear equation, we propose to use Magnus integrators, which can be considered as exponential integrators for linear problems as well as time averaging methods. Exponential integrators have recently developed and analyzed, showing a high performance. In addition, they can provide analytical and numerical approximations. We show that most relevant qualitative properties are preserved by the Magnus integrators. We have also extended the results to the zero sum game where the coupled RDEs cannot be written as an equivalent linear problem.

We have presented symmetric exponential integrators which approximate the whole set of equations (as an extra bonus, they can be easily used to get higher order methods by extrapolation). These methods require to compute or approximate the exponential of matrices or their actions on vectors. The illustrative examples showed some promising features in favor of exponential integrators for a wide range of problems.

It is also important to remark that in this work we have presented a procedure to numerically integrate Riccati differential equations which are coupled with other linear equations, where standard methods for the RDE are not efficient. Riccati differential equations coupled with other linear equations appear also, for instance, in linear quadratic optimal control [42] as well as in the embedding formulation for boundary value problems [43] while in optimal quantum control it is usually required to integrate forward and backward coupled linear differential equations, and we believe that the algorithms presented in this work can also be adapted and then to be useful tools for solving numerically these problems.

Acknowledgments

The authors acknowledge the support of the Generalitat Valenciana through the project GV/2009/032. The work of SB has also been partially supported by Ministerio de Ciencia e Innovación (Spain) under the coordinated project MTM2010-18246-C03 (co-financed by the ERDF of the European Union) and the work of EP has also been partially supported by Ministerio de Ciencia e Innovación of Spain, by the project MTM2009-08587.

References

- [1] B.D.O. Anderson, J.B. Moore, *Optimal Control: Linear Quadratic Methods*, Dover, New York, 1990.
- [2] T. Basar, G.J. Olsder, *Dynamic Non-cooperative Game Theory*, second ed., SIAM, Philadelphia, 1999.
- [3] J.B. Cruz, C.I. Chen, Series Nash solution of two person non zero sum linear quadratic games, *J. Optim. Theory Appl.* 7 (1971) 240–257.
- [4] E. Dockner, S. Jorgensen, N. Van Long, *Differential Games in Economic and Management Science*, Cambridge Univ. Press, 2000.
- [5] J. Engwerda, *LQ Dynamic Optimization and Differential Games*, John Wiley and Sons, 2005.
- [6] R. Isaacs, *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit. Control and Optimization*, Courier Dover Publications, 1999.
- [7] A.W. Starr, Y.C. Ho, Non-zero sum differential games, *J. Optim. Theory Appl.* 3 (1969) 179–197.
- [8] H. Abou-Kandil, G. Freiling, V. Ionescu, G. Jank, *Matrix Riccati Equations in Control and Systems Theory*, Birkhäuser Verlag, Basel, 2003.
- [9] W.T. Reid, *Riccati Differential Equations*, Academic, New York, 1972.
- [10] C. Brif, R. Chakrabarti, H. Rabitz, Control of quantum phenomena: past, present and future, *New J. Phys.* 12 (2010) 075008 (68pp).
- [11] S. Blanes, F. Casas, J.A. Oteo, J. Ros, The Magnus expansion and some of its applications, *Phys. Rep.* 470 (2009) 151–238.
- [12] S. Blanes, F. Casas, J. Ros, High order optimized geometric integrators for linear differential equations, *BIT* 42 (2002) 262–284.
- [13] A. Iserles, H.Z. Munthe-Kaas, S.P. Nørsett, A. Zanna, Lie group methods, *Acta Numer.* 9 (2000) 215–365.
- [14] A. Iserles, S.P. Nørsett, On the solution of linear differential equations in Lie groups, *Phil. Trans. R. Soc. A* 357 (1999) 983–1019.
- [15] H. Abou-Kandil, P. Bertrand, Analytic solution for a class of linear quadratic open-loop Nash games, *Internat. J. Control* 43 (1986) 997–1002.
- [16] S. Blanes, P.C. Moan, Splitting methods for the time-dependent Schrödinger equation, *Phys. Lett. A* 265 (2000) 35–42.
- [17] S. Blanes, P.C. Moan, Splitting methods for non-autonomous differential equations, *J. Comput. Phys.* 170 (2001) 205–230.
- [18] L. Jódar, H. Abou-Kandil, A resolution method for Riccati differential systems coupled in their quadratic terms, *SIAM J. Math. Comp.* 19 (1988) 1225–1230.
- [19] L. Jódar, E. Ponsoda, R. Company, Solutions of coupled Riccati equations arising in differential games, *Control Cybernet.* 24 (2) (1995) 117–128.
- [20] S. Bittanti, A.J. Laub, J.C. Willems, *The Riccati Equation*, Springer, 1991.
- [21] S. Blanes, L. Jódar, E. Ponsoda, Approximate solutions with a priori error bounds for continuous coefficient matrix Riccati equations, *Math. Comput. Modelling* 31 (2000) 1–15.
- [22] E. Celledoni, A. Iserles, Approximating the exponential from a Lie algebra to a Lie group, *Math. Comp.* 69 (1998) 1457–1480.
- [23] L. Dieci, Numerical integration of the differential Riccati equation and some related issues, *SIAM J. Numer. Anal.* 29 (1992) 781–815.
- [24] L. Jódar, E. Ponsoda, Non-autonomous Riccati-type matrix differential equations: existence interval, construction of continuous numerical solutions and error bounds, *IMA J. Numer. Anal.* 15 (1995) 61–74.
- [25] S.S. Kenney, R.B. Leipnik, Numerical integration of the differential matrix Riccati equation, *IEEE Trans. Automat. Control* AC-30 (1985) 962–970.
- [26] J. Schiff, S. Schneider, A natural approach to the numerical integration of Riccati differential equations, *SIAM J. Sci. Comput.* 36 (1996) 1392–1413.
- [27] E. Hairer, C. Lubich, G. Wanner, *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*, second ed., in: Springer Series in Computational Mathematics, vol. 31, Springer-Verlag, 2006.
- [28] W. Magnus, On the exponential solution of differential equations for a linear operator, *Commun. Pure Appl. Math.* 7 (1954) 649–673.
- [29] F. Fer, Résolution de l'équation matricielle $U' = pU$ par produit infini d'exponentielles matricielles, *Bull. Cl. Sci. Acad. R. Belg.* 44 (1958) 818–829.
- [30] R.M. Wilcox, Exponential operators and parameter differentiation in quantum physics, *J. Math. Phys.* 8 (1967) 962–982.
- [31] F. Casas, Sufficient conditions for the convergence of the Magnus expansion, *J. Phys. A: Math. Theor.* 40 (2007) 15001–15017.
- [32] V.I. Zhukovskiy, in: V. Lakshmikantham, A.A. Martynyuk (Eds.), *Lyapunov Functions in Differential Games*, in: Series in Stability and Control: Theory, Methods and Applications, vol. 19, Taylor and Francis, Great Britain, 2003.
- [33] A. Alvermannand, H. Fehske, High-order commutator-free exponential time-propagation of driven quantum systems, *J. Comput. Phys.* 230 (2011) 5930–5956.
- [34] S. Blanes, P.C. Moan, Fourth- and sixth-order commutator-free Magnus integrators for linear and non-linear dynamical systems, *Appl. Numer. Math.* 56 (2006) 1519–1537.
- [35] M. Hochbruck, A. Ostermann, Exponential integrators, *Acta Numer.* 19 (2010) 209–286.
- [36] C.B. Moler, C.F. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, *SIAM Rev.* 45 (2003) 3–49.
- [37] R.B. Sidje, Expokit: a software package for computing matrix exponentials, *ACM Trans. Math. Software* 24 (1998) 130–156.
- [38] G.H. Golub, C.F. Van Loan, *Matrix Computations*, John Hopkins Univ. Press, Baltimore, Maryland, 1993.
- [39] P. Lancaster, M. Tismenetsky, *The Theory of Matrices*, second ed., Academic Press, London, 1985.
- [40] M.H. Foley, W.E. Schmitendorf, On a class of non-zero sum linear quadratic differential games, *J. Optim. Theory Appl.* 7 (5) (1971) 357–377.
- [41] V. Kaitala, M. Pohjola, Sustainable international agreement on greenhouse warming. A game theory study, in: Carraro, Filar (Eds.), *Control and Game Theoretic Models of the Environment*, Birkhauser, Boston, 1995, pp. 67–87.
- [42] G.D. Hu, Symplectic Runge–Kutta methods for the linear quadratic regulator problem, *Int. J. Math. Anal.* 1 (2007) 293–304.
- [43] H.B. Keller, M. Lentini, Invariant imbedding, the box scheme and an equivalence between them, *SIAM J. Numer. Anal.* 19 (1982) 942–962.