A structure-preserving doubling algorithm for continuous-time algebraic Riccati equations

Article *in* Linear Algebra and its Applications · February 2005

DOI: 10.1016/j.laa.2004.10.010

CITATIONS
106

READS
144

3 authors, including:

Eric Chu
Monash University (Australia)
131 PUBLICATIONS **2,056** CITATIONS
SEE PROFILE

Wen-Wei Lin
National Chiao Tung University
224 PUBLICATIONS **3,490** CITATIONS
SEE PROFILE

Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Linear Algebra and its Applications 396 (2005) 55–80
LINEAR ALGEBRA AND ITS APPLICATIONS
www.elsevier.com/locate/laa

A structure-preserving doubling algorithm for continuous-time algebraic Riccati equations

E.K.-W. Chu [a,*], H.-Y. Fan [b], W.-W. Lin [b]

[a] *School of Mathematical Sciences, Monash University, Building 28, VIC 3800, Australia*
[b] *Department of Finance, Yuanpei Institute of Science and Technology, Hsinchu 300, Taiwan*

Received 12 June 2003; accepted 6 October 2004

Submitted by R. Byers

Abstract

Continuous-time algebraic Riccati equations (CAREs) can be transformed, *à la* Cayley, to discrete-time algebraic Riccati equations (DAREs). The efficient structure-preserving doubling algorithm (SDA) for DAREs, from [E.K.-W. Chu, H.-Y. Fan, W.-W. Lin, A structure-preserving doubling algorithm for periodic discrete-time algebraic Riccati equations, preprint 2002-28, NCTS, National Tsing Hua University, Hsinchu 300, Taiwan, 2003; E.K.-W. Chu, H.-Y. Fan, W.-W. Lin, C.-S. Wang, A structure-preserving doubling algorithm for periodic discrete-time algebraic Riccati equations, preprint 2002-18, NCTS, National Tsing Hua University, Hsinchu 300, Taiwan, 2003], can then be applied. In this paper, we develop the structure-preserving doubling algorithm from a new point of view and show its quadratic convergence under assumptions which are weaker than stabilizability and detectability, as well as practical issues involved in the application of the SDA to CAREs. A modified version of the SDA, developed for DAREs with a "doubly symmetric" structure, is also presented. Extensive numerical results show that our approach is efficient and competitive.

AMS classification: 93B50; 93B52; 93C05; 93D15

Keywords: Cayley transform; Continuous-time algebraic Riccati equation; Doubling algorithm; Matrix sign function; Structure-preserving

* Corresponding author.
E-mail addresses: eric.chu@sci.monash.edu.au (E.K.-W. Chu), hyfan@mail.yust.edu.tw (H.-Y. Fan), wwlin@am.nthu.edu.tw (W.-W. Lin).

0024-3795/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.laa.2004.10.010

1. Introduction

In this paper we investigate a structure-preserving doubling algorithm [24,37] for the computation of the symmetric positive semi-definite (s.p.s.d.) solution X (i.e. $X \geqslant 0$) to the continuous-time algebraic Riccati equation (CARE):

$$-XGX + A^{\mathrm{T}}X + XA + H = 0, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$, $X \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{m\times m}$ is symmetric positive definite (or s.p.d.; i.e. $R > 0$), $G = BR^{-1}B^{\mathrm{T}} \geqslant 0$ and $H = C^{\mathrm{T}}C \geqslant 0$ with $B \in \mathbb{R}^{n\times m}$ and $C^{\mathrm{T}} \in \mathbb{R}^{n\times p}$ being of full column rank.

Eq. (1) arises frequently in solving the continuous-time linear optimal control problem:

$$\min_{u} J = \frac{1}{2}\int_0^{\infty} \left(x^{\mathrm{T}}C^{\mathrm{T}}Cx + u^{\mathrm{T}}Ru\right) \mathrm{d}t \quad \text{subject to } \dot{x} = Ax + Bu. \tag{2}$$

The optimal feedback control u^* for (2) is given by

$$u^* = -R^{-1}B^{\mathrm{T}}Xx, \tag{4}$$

where X is the s.p.s.d. solution to the CARE (1). We assume that the pair (A, B) is stabilizable (S) (i.e. if $w^{\mathrm{T}}B = 0$ and $w^{\mathrm{T}}A = \lambda w^{\mathrm{T}}$ for some $\lambda \in \mathbb{C}$, then $\mathrm{Re}(\lambda) < 0$ or $w = 0$) and that the pair (A, C) is detectable (D) (i.e. $(A^{\mathrm{T}}, C^{\mathrm{T}})$ is stabilizable). Under assumptions (S) and (D), the CARE (1) has been proved to possess a unique s.p.s.d. solution [39].

Consider the $2n \times 2n$ Hamiltonian matrix $\mathscr{H}$ associated with the CARE (1):

$$\mathscr{H} = \begin{bmatrix} A & -G \\ -H & -A^{\mathrm{T}} \end{bmatrix}, \tag{5}$$

which satisfies

$$\mathscr{H}J = -J\mathscr{H}^{\mathrm{T}}, \quad J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$$

with I_n denoting the identity matrix of order n. By (5), the CARE (1) can be written as

$$\mathscr{H}\begin{bmatrix} I \\ X \end{bmatrix} = \begin{bmatrix} I \\ X \end{bmatrix}\Phi, \tag{6}$$

where $\Phi \in \mathbb{R}^{n\times n}$ and the spectrum $\sigma(\Phi)$ is on the stable left half plane $\mathbb{C}_-$. Under assumptions (S) and (D), the Hamiltonian matrix $\mathscr{H}$ has exactly n eigenvalues on $\mathbb{C}_-$. If the columns of $[X_1^{\mathrm{T}}, X_2^{\mathrm{T}}]^{\mathrm{T}}$ span the stable invariant subspace of $\mathscr{H}$, then X_1 is nonsingular and $X = X_2X_1^{-1} \geqslant 0$ solves the CARE (1) (see, e.g., [39,44]).

A numerically backward stable algorithm `care`, proposed by Laub [39], computes X by applying the QR algorithm with reordering [4,16,48] to the eigenvalue problem $\mathscr{H}x = \lambda x$. Unfortunately, the QR algorithm preserves neither the Hamiltonian

structure of $\mathscr{H}$ nor the associated splitting of eigenvalues. A structure-preserving algorithm has been proposed by Ammar and Mehrmann [1] which utilizes orthogonal symplectic transformations in computing a basis for the stable invariant subspace of $\mathscr{H}$. A stable symplectic orthogonal method has been suggested by Byers [19] but applied only to systems with single input or output. Many iterative methods have been suggested for solving CAREs over the past 20 years. Newton's method has been applied in extensive literature [28,31,38,41,47]. A defect correction method for modifying an approximate solution has also been proposed by Mehrmann and Tan [43]. These methods require a good starting approximate solution, and can therefore be regarded as iterative refinement methods, to be combined with other direct methods (see Bunse-Gerstner et al. [17,18] or Mehrmann [41] for details). The structure-preserving matrix sign function methods (MSGM) [7,11–14,20,21,27,33,46] have been extended by Barraud [8,9] and Gardiner and Laub [29].

A class of methods, referred to as the doubling algorithms (DA), has attracted much interests in the 70s and 80s (see [2] and the references therein). These methods originate from the fixed-point iteration derived from the discrete-time algebraic Riccati equation (DARE):

$$X_{k+1} = \widehat{A}^{\mathrm{T}} X_k (I + \widehat{G} X_k)^{-1} \widehat{A} + \widehat{H}.$$

Instead of producing the sequence $\{X_k\}$, doubling algorithms produce $\{X_{2^k}\}$. CAREs can be tackled after being transformed to DAREs via the Cayley transform. However, the convergence of the algorithm was proven only when $\widehat{A}$ is nonsingular [2], and for $(\widehat{A}, \widehat{G}, \widehat{H})$ which is stabilizable and detectable [36]. DAs were largely forgotten in the past decade. Recently, DAs have been revived for (periodic) DAREs, because of a better theoretical understanding. Stronger convergence results have been proved for $(\widehat{A}, \widehat{G}, \widehat{H})$ under weaker assumptions than stabilizability and detectability [24]. Superior numerical results, in comparison to state-of-the-art methods on a wide range of test problems, have been obtained because of the stronger structure-preserving properties and the superior operations count.

In this paper, we propose a doubling algorithm for CAREs. The CAREs are transformed to DAREs, with the corresponding Hamiltonian matrix transformed into a symplectic matrix pair by the Cayley transform. Nice convergence properties are inherited from the structure-preserving doubling algorithm (SDA) [24] applied to the corresponding DARE. The SDA preserves matrix pairs in SSF which is a stronger property than symplecticity. In the CARE setting, the matrix sign function methods preserve the Hamiltonian structure in $\mathscr{H}$ while the SDA preserves, in each iterative step, the associated symplectic matrix pair $(\widehat{\mathscr{N}}, \widehat{\mathscr{L}})$ in SSF. Although under the influence of numerical errors, the matrix pairs through the SDA retain their stabilizability, detectability as well as eigenstructures (with exactly half of the spectrum being stable; see details in [24]). This stronger structure-preserving property is its main strength and the reason of its accuracy. In Section 4, a modified version of the SDA (SDA_m) is developed, for "doubly symmetric" DAREs, where $\widehat{A}, \widehat{G} = \widehat{H}$ are symmetric and persymmetric. The SDA_m preserves the symplectic and doubly

symmetric structures of the DARE, resulting in better accuracy than the SDA. We have extensively tested the SDA against the MSGM and `care`. Numerical results showed that the doubling algorithm for CAREs is competitive and promising.

Finally, it is important to stress that matrix sign functions can be applied to more general Hamiltonian matrices in other applications, such as those from H_∞ control with G and H being indefinite. A scaling strategy [21] may also accelerate its convergence. Also, the SDA requires the transformation of the CARE by the Cayley transform, which requires the estimation of the parameter γ (see Section 3).

2. SDA and matrix sign function method

In this section we propose a structure-preserving doubling algorithm (SDA) for solving the CARE (1) based on the doubling algorithms in [24,37]. In addition, the well-known structure-preserving matrix sign function methods [7,11–14,20,21,27, 33,46] are also reviewed from the point of view of preserving Hamiltonian structure.

Let $\mathbf{H}$ be the set of $2n \times 2n$ Hamiltonian matrices, i.e.,

$$\mathbf{H} = \left\{ \mathscr{H} \,\middle|\, \mathscr{H} = \begin{bmatrix} A & -G \\ -H & -A^{\mathrm{T}} \end{bmatrix};\ A, H, G \in \mathbb{R}^{n\times n};\ H, G \geqslant 0 \right\}. \tag{7}$$

Note that if $\mathscr{H} \in \mathbf{H}$ then $\mathscr{H}J = -J\mathscr{H}^{\mathrm{T}}$. We call a $2n \times 2n$ matrix pair $(\mathscr{N}, \mathscr{L})$ symplectic if $\mathscr{N}J\mathscr{N}^{\mathrm{T}} = \mathscr{L}J\mathscr{L}^{\mathrm{T}}$. Let $\mathbf{S}$ be the set of $2n \times 2n$ symplectic matrix pairs in the standard symplectic form (SSF):

$$\mathbf{S} = \left\{ (\widehat{\mathscr{N}}, \widehat{\mathscr{L}}) \,\middle|\, \widehat{\mathscr{L}} = \begin{bmatrix} I & \widehat{G} \\ 0 & \widehat{A}^{\mathrm{T}} \end{bmatrix},\ \widehat{\mathscr{N}} = \begin{bmatrix} \widehat{A} & 0 \\ -\widehat{H} & I \end{bmatrix};\ \widehat{A}, \widehat{H}, \widehat{G} \in \mathbb{R}^{n\times n};\ \widehat{G}, \widehat{H} \geqslant 0 \right\}. \tag{8}$$

It is easily seen that symplecticity is weaker than symplecticity in SSF. Our proposed algorithm preserves the stronger structure and gives rise to better numerical performance.

We shall show how the CARE (1), associated with the corresponding Hamiltonian matrix

$$\mathscr{H} \equiv \begin{bmatrix} A & -G \\ -H & -A^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^{\mathrm{T}} \\ -C^{\mathrm{T}}C & -A^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{2n\times 2n}$$

can be transformed to an equivalent DARE.

By using the Cayley transform with some appropriate $\gamma > 0$, the Hamiltonian matrix $\mathscr{H}$ can be transformed to a symplectic matrix pair $(\mathscr{N}, \mathscr{L}) \equiv (\mathscr{H} + \gamma I, \mathscr{H} - \gamma I)$ [41,42]. In the following, we construct an equivalence transformation from $(\mathscr{N}, \mathscr{L})$ to a symplectic matrix pair $(\widehat{\mathscr{N}}, \widehat{\mathscr{L}}) \in \mathbf{S}$.

Let

$$A_\gamma \equiv A - \gamma I, \quad \bar{A}_\gamma \equiv A + \gamma I.$$

Starting from

$$\mathscr{N} = \begin{bmatrix} \bar{A}_\gamma & -G \\ -H & -A_\gamma^{\mathrm{T}} \end{bmatrix}, \quad \mathscr{L} = \begin{bmatrix} A_\gamma & -G \\ -H & -\bar{A}_\gamma^{\mathrm{T}} \end{bmatrix},$$

we choose a $\gamma > 0$ such that the matrices A_γ and $A_\gamma + GA_\gamma^{-\mathrm{T}}H$ are well-conditioned (see Section 3 later for details). To transform the symplectic matrix pair $(\mathscr{N}, \mathscr{L})$ to $(\widehat{\mathscr{N}}, \widehat{\mathscr{L}}) \in \mathbf{S}$, let

$$T_1 \equiv \begin{bmatrix} A_\gamma^{-1} & 0 \\ HA_\gamma^{-1} & I \end{bmatrix}, \quad T_2 \equiv \begin{bmatrix} I & 0 \\ 0 & \big(-HA_\gamma^{-1}G - A_\gamma^{\mathrm{T}}\big)^{-1} \end{bmatrix},$$
$$T_3 \equiv \begin{bmatrix} I & A_\gamma^{-1}G \\ 0 & I \end{bmatrix}.$$

Simple calculations produce

$$\begin{aligned} \widehat{\mathscr{N}} &= \begin{bmatrix} \widehat{A} & 0 \\ -\widehat{H} & I \end{bmatrix} = T_3T_2T_1\mathscr{N} \\ &= T_3T_2 \begin{bmatrix} A_\gamma^{-1}\bar{A}_\gamma & -A_\gamma^{-1}G \\ HA_\gamma^{-1}\bar{A}_\gamma - H & -HA_\gamma^{-1}G - A_\gamma^{\mathrm{T}} \end{bmatrix} \\ &= T_3 \begin{bmatrix} A_\gamma^{-1}\bar{A}_\gamma & -A_\gamma^{-1}G \\ \big(-HA_\gamma^{-1}G - A_\gamma^{\mathrm{T}}\big)^{-1}\big(HA_\gamma^{-1}\bar{A}_\gamma - H\big) & I \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} \widehat{\mathscr{L}} &= \begin{bmatrix} I & \widehat{G} \\ 0 & \widehat{A}^{\mathrm{T}} \end{bmatrix} = T_3T_2T_1\mathscr{L} \\ &= T_3T_2 \begin{bmatrix} I & -A_\gamma^{-1}G \\ 0 & -HA_\gamma^{-1}G - \bar{A}_\gamma^{\mathrm{T}} \end{bmatrix} \\ &= T_3 \begin{bmatrix} I & -A_\gamma^{-1}G \\ 0 & \big(-HA_\gamma^{-1}G - A_\gamma^{\mathrm{T}}\big)^{-1}\big(-HA_\gamma^{-1}G - \bar{A}_\gamma^{\mathrm{T}}\big) \end{bmatrix}, \end{aligned}$$

where

$$\begin{aligned} \widehat{A} &= \big(\bar{A}_\gamma + GA_\gamma^{-\mathrm{T}}H\big)\big(A_\gamma + GA_\gamma^{-\mathrm{T}}H\big)^{-1}, \\ \widehat{G} &= -A_\gamma^{-1}G + A_\gamma^{-1}G\big(A_\gamma^{\mathrm{T}} + HA_\gamma^{-1}G\big)^{-1}\big(\bar{A}_\gamma^{\mathrm{T}} + HA_\gamma^{-1}G\big), \\ \widehat{H} &= \big(A_\gamma^{\mathrm{T}} + HA_\gamma^{-1}G\big)^{-1}\big(HA_\gamma^{-1}\bar{A}_\gamma - H\big). \end{aligned}$$

Note that $\mathscr{L}^{-1}\mathscr{N} = \widehat{\mathscr{L}}^{-1}\widehat{\mathscr{N}}$. Since $\bar{A}_\gamma = A_\gamma + 2\gamma I$, it follows that:

$$\widehat{A} = I + 2\gamma\big(A_\gamma + GA_\gamma^{-\mathrm{T}}H\big)^{-1}, \tag{9}$$

$$\widehat{G} = 2\gamma A_\gamma^{-1} G\big(A_\gamma^{\mathrm{T}} + H A_\gamma^{-1} G\big)^{-1} = 2\gamma A_\gamma^{-1} G A_\gamma^{-\mathrm{T}}\big(I + H A_\gamma^{-1} G A_\gamma^{-\mathrm{T}}\big)^{-1}, \tag{10}$$

$$\widehat{H} = 2\gamma\big(A_\gamma^{\mathrm{T}} + H A_\gamma^{-1} G\big)^{-1} H A_\gamma^{-1} = 2\gamma\big(I + A_\gamma^{-\mathrm{T}} H A_\gamma^{-1} G\big)^{-1} A_\gamma^{-\mathrm{T}} H A_\gamma^{-1}. \tag{11}$$

From (10), (11) and Lemma A.1 in Appendix A, we know that the matrices $\widehat{G}$ and $\widehat{H}$ are positive semi-definite. We thus obtain the desired symplectic matrix pair in SSF, i.e.,

$$(\widehat{\mathscr{N}}, \widehat{\mathscr{L}}) \equiv \left(\begin{bmatrix} \widehat{A} & 0 \\ -\widehat{H} & I \end{bmatrix}, \begin{bmatrix} I & \widehat{G} \\ 0 & \widehat{A}^{\mathrm{T}} \end{bmatrix}\right) \in \mathbf{S},$$

where $\widehat{A}$, $\widehat{G}$ and $\widehat{H}$ are given by (9)–(11). The DARE associated with the symplectic matrix pair $(\widehat{\mathscr{N}}, \widehat{\mathscr{L}})$ in SSF is

$$X = \widehat{A}^{\mathrm{T}} X (I + \widehat{G} X)^{-1} \widehat{A} + \widehat{H} \tag{12}$$

on which the efficient SDA [24] can be applied. Note that X is the unique s.p.s.d. solution to the above DARE as well as the CARE (1). Moreover, in Theorems 1 and 2 of [37], the pairs $(\widehat{A}, \widehat{B})$ and $(\widehat{A}, \widehat{C})$ are proven to be stabilizable and detectable, respectively, where the matrices $\widehat{G} = \widehat{B}\widehat{B}^{\mathrm{T}}$ and $\widehat{H} = \widehat{C}^{\mathrm{T}}\widehat{C}$ are full rank decompositions (FRD).

Using (9)–(11) to transform the CARE (1) to an equivalent DARE (12) with the associated symplectic matrix pair $(\widehat{\mathscr{N}}, \widehat{\mathscr{L}})$ in SSF, the SDA in [24] can then be modified to the following algorithm for CAREs: (with Im denoting the imaginary axis).

2.1. *Structure-preserving doubling algorithm (SDA):*

`Input`: $\mathscr{H} = \begin{bmatrix} A & -G \\ -H & -A^{\mathrm{T}} \end{bmatrix} \in \mathbf{H}$ with $\sigma(\mathscr{H}) \cap \mathrm{Im} = \emptyset$; ϵ

`Output`: the stabilizing solution $X = X^{\mathrm{T}} \geqslant 0$ to the CARE (1).

`Find` an appropriate value $\hat{\gamma} > 0$.

`Compute` $\widehat{A}_0 \leftarrow I + 2\hat{\gamma}(A_{\hat{\gamma}} + G A_{\hat{\gamma}}^{-\mathrm{T}} H)^{-1}$, $\widehat{G}_0 \leftarrow 2\hat{\gamma} A_{\hat{\gamma}}^{-1} G (A_{\hat{\gamma}}^{\mathrm{T}} + H A_{\hat{\gamma}}^{-1} G)^{-1}$,
$\widehat{H}_0 \leftarrow 2\hat{\gamma}(A_{\hat{\gamma}}^{\mathrm{T}} + H A_{\hat{\gamma}}^{-1} G)^{-1} H A_{\hat{\gamma}}^{-1}$, $j \leftarrow 0$;

`Do` until convergence:

`Compute` $\widehat{A}_{j+1} \leftarrow \widehat{A}_j (I + \widehat{G}_j \widehat{H}_j)^{-1} \widehat{A}_j$, $\widehat{G}_{j+1} \leftarrow \widehat{G}_j + \widehat{A}_j \widehat{G}_j (I + \widehat{H}_j \widehat{G}_j)^{-1} \widehat{A}_j^{\mathrm{T}}$,
$\widehat{H}_{j+1} \leftarrow \widehat{H}_j + \widehat{A}_j^{\mathrm{T}} (I + \widehat{H}_j \widehat{G}_j)^{-1} \widehat{H}_j \widehat{A}_j$, $j \leftarrow j + 1$;

`If` $\|\widehat{H}_j - \widehat{H}_{j-1}\| \leqslant \epsilon \|\widehat{H}_j\|$, `Stop`;

`End`

`Set` $X \leftarrow \widehat{H}_j$.

2.2. Convergence of SDA

Let $\widehat{\mathscr{N}} = \begin{bmatrix} \widehat{A} & 0 \\ -\widehat{H} & I \end{bmatrix}$, $\widehat{\mathscr{L}} = \begin{bmatrix} I & \widehat{G} \\ 0 & \widehat{A}^{\mathrm{T}} \end{bmatrix}$, where $\widehat{G} = \widehat{G}^{\mathrm{T}}$, $\widehat{H} = \widehat{H}^{\mathrm{T}}$. Suppose $\widehat{\mathscr{N}} - \lambda\widehat{\mathscr{L}}$ has no eigenvalues on the unit circle and there exist nonsingular Q, Z such that

$$Q\widehat{\mathscr{N}}Z = \begin{bmatrix} J_{\mathrm{s}} & 0 \\ 0 & I \end{bmatrix}, \quad Q\widehat{\mathscr{L}}Z = \begin{bmatrix} I & 0 \\ 0 & J_{\mathrm{s}} \end{bmatrix}, \tag{13}$$

where the spectrum $\lambda(J_{\mathrm{s}}) \in O_{\mathrm{s}} \equiv \{\lambda : |\lambda| < 1\}$. In the following we quote the convergence results for the SDA algorithm from [24,25].

Theorem 2.1 [24]. *Let* $\widehat{\mathscr{N}} = \begin{bmatrix} \widehat{A} & 0 \\ -\widehat{H} & I \end{bmatrix}$ *and* $\widehat{\mathscr{L}} = \begin{bmatrix} I & \widehat{G} \\ 0 & \widehat{A}^{\mathrm{T}} \end{bmatrix}$, *where* $\widehat{G} = \widehat{G}^{\mathrm{T}}$, $\widehat{H} = \widehat{H}^{\mathrm{T}}$. *Suppose* $\widehat{\mathscr{N}} - \lambda\widehat{\mathscr{L}}$ *has no eigenvalues on the unit circle and there exist nonsingular* Q, Z *such that* (13) *holds. Denote* $Z = \begin{bmatrix} Z_1 & Z_3 \\ Z_2 & Z_4 \end{bmatrix}$, $Z_i \in \mathbb{R}^{n\times n}$ *for* $i = 1, 2, 3, 4$. *If* Z_1 *and* Z_4 *are invertible, then the sequences* $\{\widehat{A}_j,\ \widehat{H}_j,\ \widehat{G}_j\}$ *computed by the SDA algorithm satisfy*

(i) $\|\widehat{A}_j\| = \mathrm{O}(\|J_{\mathrm{s}}^{2^j}\|) \to 0$ *as* $j \to \infty$,
(ii) $\widehat{H}_j \to X$, *where* X *solves the DARE* (12) :

$$X = \widehat{A}^{\mathrm{T}}X(I + \widehat{G}X)^{-1}\widehat{A} + \widehat{H},$$

(iii) $\widehat{G}_j \to Y$, *where* Y *solves the dual DARE*

$$Y = \widehat{A}Y(I + \widehat{H}Y)^{-1}\widehat{A}^{\mathrm{T}} + \widehat{G}. \tag{14}$$

Moreover, the convergence rate in (i)–(iii) *above is* $\mathrm{O}\left(|\lambda_n|^{2^j}\right)$, *where* $|\lambda_1| \leqslant \cdots \leqslant |\lambda_n| < 1 < |\lambda_n|^{-1} \leqslant \cdots \leqslant |\lambda_1|^{-1}$ *with* $\lambda_i,\ \lambda_i^{-1}$ *being the eigenvalues of* $\widehat{\mathscr{N}} - \lambda\widehat{\mathscr{L}}$ (*including* 0 *and* ∞).

The following lemma proves that the stabilizability and detectability properties are preserved by the SDA throughout its iterative process. From Lemma A.1 and the SDA algorithm, the matrices $\hat{G}_j$ and $\hat{H}_j$ are positive semi-definite for each $j \geqslant 1$.

Lemma 2.2 [24]. *The stabilizability of* $(\widehat{A}, \widehat{B})$ *implies that* $(\widehat{A}_j, \widehat{B}_j)$ *is stabilizable, where* $\widehat{G}_j = \widehat{B}_j\widehat{B}_j^{\mathrm{T}} \geqslant 0$ *is a FRD of* $\widehat{G}_j$ *for each* $j \geqslant 1$. *The detectability of* $(\widehat{A}, \widehat{C})$ *implies that* $(\widehat{A}_j, \widehat{C}_j)$ *is detectable, where* $\widehat{H}_j = \widehat{C}_j^{\mathrm{T}}\widehat{C}_j \geqslant 0$ *is a FRD of* $\widehat{H}_j$ *for each* $j \geqslant 1$.

Theorem 2.3 [24]. *Let* $\widehat{\mathscr{N}} = \begin{bmatrix} \widehat{A} & 0 \\ -\widehat{H} & I \end{bmatrix}$ *and* $\widehat{\mathscr{L}} = \begin{bmatrix} I & \widehat{G} \\ 0 & \widehat{A}^{\mathrm{T}} \end{bmatrix}$, *where the matrices* $\widehat{G} = \widehat{B}\widehat{B}^{\mathrm{T}} \geqslant 0$ (*FRD*) *and* $\widehat{H} = \widehat{C}^{\mathrm{T}}\widehat{C} \geqslant 0$ (*FRD*). *Assume that* $(\widehat{A}, \widehat{B})$ *is stabilizable*

and $(\widehat{A}, \widehat{C})$ *is detectable. Then the sequences* $\{\widehat{A}_j, \widehat{H}_j, \widehat{G}_j\}$ *computed by the SDA satisfy* (i), (ii), (iii) *as in Theorem* 2.1.

Remarks. Theorem 2.1 directly proves, under the assumptions that $\widehat{\mathcal{N}} - \lambda\widehat{\mathcal{L}}$ have no unit modulo eigenvalues and Z_1, Z_4 are invertible, that the sequences $\{\widehat{A}_j, \widehat{H}_j, \widehat{G}_j\}$ generated by the SDA converge to zero and the unique s.p.s.d. solutions of the DAREs in (12) and (14), respectively. Lemma 2.2 shows the preservation of stabilizability and detectability of the iterates $(\widehat{A}_j, \widehat{G}_j, \widehat{H}_j)$ generated by the SDA. Furthermore, in Theorem 2.3, we see that the assumptions in Theorem 2.1 are weaker than conditions (S) and (D). This distinction of preserving the symplectic structure in SSF, as well as the difference in operation counts, are responsible for the superior performance of the SDA.

On the other hand, for a given $\mathcal{H} = \begin{bmatrix} A & -G \\ -H & -A^{\mathrm{T}} \end{bmatrix} \in \mathbf{H}$ with $\sigma(\mathcal{H}) \cap \mathrm{Im} = \emptyset$, the matrix sign function of $\mathcal{H}$ can also be used to develop a structure-preserving method for computing the stabilizing solution of CARE (1). A thorough discussion and the details of practical implementation are given in [21,41]. The main MSGM algorithm is described as follows. Other modified versions can be found in [5,8,9,22,29] and references therein.

Matrix sign function algorithm: [7,11–14,20,21,27,33,46]

`Input`: $\mathcal{H} = \begin{bmatrix} A & -G \\ -H & -A^{\mathrm{T}} \end{bmatrix} \in \mathbf{H}$ with $\sigma(\mathcal{H}) \cap \mathrm{Im} = \emptyset$; ϵ.

`Output`: the stabilizing solution $X = X^{\mathrm{T}} \geqslant 0$ to the CARE (1).

`Let` $\mathcal{H}_0 \leftarrow \mathcal{H}$, $j \leftarrow 0$.

`Do` until convergence:

  `Compute` $\mathcal{H}_{j+1} \leftarrow \frac{1}{2}(\mathcal{H}_j + \mathcal{H}_j^{-1})$, $j \leftarrow j + 1$;

  If $\|\mathcal{H}_j - \mathcal{H}_{j-1}\| \leqslant \epsilon \|\mathcal{H}_j\|$, `Stop`;

`End`

$\mathrm{sgn}(\mathcal{H}) \leftarrow \mathcal{H}_j$;

`Solve` $(I - \mathrm{sgn}(\mathcal{H}))\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = 0$;

`Compute` $X \leftarrow X_2 X_1^{-1}$.

Remarks

(i) Notice that $\begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ spans the stable invariant subspace of the $\mathcal{H}$.

(ii) Both the SDA and the matrix sign function algorithm require $\frac{32}{3}n^3$ flops for each iterative step.

(iii) When working with the Hamiltonian matrix $\mathcal{H}$, a more efficient and structure-preserving version of the classical matrix sign function iteration can be derived

by working only with symmetric matrices $J\mathscr{H}$. Details may be consulted in [21,41].

3. Practical implementation of SDA

3.1. Selection of $\hat{\gamma}$

Here we first derived the forward error bounds of matrices $\widehat{A}_0 \equiv \widehat{A}$, $\widehat{G}_0 \equiv \widehat{G}$ and $\widehat{H}_0 \equiv \widehat{H}$ given in (9)–(11), respectively. According to these forward errors, we can design a numerical scheme to determine an appropriate value $\hat{\gamma} > 0$. In the following roundoff analysis, we use $\mathrm{fl}(\cdot)$ to denote computed floating point values. The quantity $\mathbf{u}$ is the *unit roundoff* (or machine precision), which is typically of order 10^{-8} or 10^{-16} in single and double precision computer arithmetic, respectively. When A and B are $m \times n$ real matrices, the matrix $B := |A|$ if $b_{ij} = |a_{ij}|$ for all i, j, and $A \leqslant B$ if $a_{ij} \leqslant b_{ij}$ for all i, j. The 1-, ∞- and Frobenius matrix norms are denoted by $\|\cdot\|_1$, $\|\cdot\|_\infty$ and $\|\cdot\|_{\mathrm{F}}$, respectively.

We assume that the LU factorizations of A_γ and $W_\gamma \equiv A_\gamma + GA_\gamma^{-\mathrm{T}}H$ are computed by Gaussian elimination with partial pivoting (GEPP). We write these computed LU factors as L_A, U_A, L_W and U_W, respectively. Recall that

$$A_\gamma + \Delta A_\gamma = L_A U_A, \quad |\Delta A_\gamma| \leqslant \gamma_n |L_A|\,|U_A|, \tag{15}$$

$$W_\gamma + \Delta W_\gamma = L_W U_W, \quad |\Delta W_\gamma| \leqslant \gamma_n |L_W|\,|U_W| \tag{16}$$

with $\gamma_n := n\mathbf{u}/(1 - n\mathbf{u})$ (see, e.g., [32–Theorem 9.3]). Then we have

$$\mathrm{fl}(W_\gamma^{-1}) = W_\gamma^{-1} + E_1, \quad |E_1| \leqslant c_n\mathbf{u}\,|W_\gamma^{-1}|\,|L_W|\,|U_W|\,|\mathrm{fl}(W_\gamma^{-1})|, \tag{17}$$

where c_n is a modest constant. From (17), the forward error bound in evaluating $\widehat{A}$ in (9) is

$$\begin{aligned}&\mathrm{fl}(\widehat{A}) = \widehat{A} + E_2,\\ &|E_2| \leqslant 4\gamma c_n\mathbf{u}\,|W_\gamma^{-1}|\,|L_W|\,|U_W|\,|\mathrm{fl}(W_\gamma^{-1})| + \mathbf{u}\,|\widehat{A}| + \mathrm{O}(\mathbf{u}^2).\end{aligned} \tag{18}$$

Furthermore, from (15), we have

$$\begin{aligned}&\widehat{G}_\gamma \equiv \mathrm{fl}(2\gamma A_\gamma^{-1}G) = 2\gamma A_\gamma^{-1}G + E_3,\\ &|E_3| \leqslant 2\gamma c_n\mathbf{u}\,|A_\gamma^{-1}|\,|L_A|\,|U_A|\,|\widehat{G}_\gamma|,\end{aligned} \tag{19}$$

hence the forward error bound in evaluating $\widehat{G}$ in (10) is

$$\begin{aligned}&\mathrm{fl}(\widehat{G}) = \widehat{G} + E_4,\\ &|E_4| \leqslant 2\gamma c_n\mathbf{u}\,|A_\gamma^{-1}|\,|L_A|\,|U_A|\,|\widehat{G}_\gamma|\,|W_\gamma^{-1}|^{\mathrm{T}}\\ &\qquad + c_n\mathbf{u}\,|\mathrm{fl}(\widehat{G})|\,|U_W^{\mathrm{T}}|\,|L_W^{\mathrm{T}}|\,|W_\gamma^{-1}|^{\mathrm{T}}.\end{aligned} \tag{20}$$

Finally, from (15), we have

$$\widehat{H}_\gamma \equiv \mathrm{fl}(2\gamma H A_\gamma^{-1}) = 2\gamma H A_\gamma^{-1} + E_5,$$
$$|E_5| \leqslant 2\gamma c_n \mathbf{u}\, |\widehat{H}_\gamma|\, |L_A|\, |U_A|\, |A_\gamma^{-1}| \tag{21}$$

and the forward error bound in evaluating $\widehat{H}$ in (11) is

$$\mathrm{fl}(\widehat{H}) = \widehat{H} + E_6,$$
$$|E_6| \leqslant 2\gamma c_n \mathbf{u}\, |W_\gamma^{-1}|^{\mathrm{T}} |\widehat{H}_\gamma|\, |L_A|\, |U_A|\, |A_\gamma^{-1}| + c_n \mathbf{u}\, |W_\gamma^{-1}|^{\mathrm{T}} |U_W^{\mathrm{T}}|\, |L_W^{\mathrm{T}}|\, |\mathrm{fl}(\widehat{H})|. \tag{22}$$

For GEPP, we have in practice $\| \, |L_A|\, |U_A| \, \|_\infty \approx \|A_\gamma\|_\infty$ and $\| \, |L_W|\, |U_W| \, \|_\infty \approx \|W_\gamma\|_\infty$, and it follows from (18), (20) and (22) that:

$$\|\mathrm{fl}(\widehat{A}) - \widehat{A}\|_\infty \lesssim 4c_n \mathbf{u}\gamma \kappa_\infty(W_\gamma)\, \|\mathrm{fl}(W_\gamma^{-1})\|_\infty + \mathbf{u}\, \|\widehat{A}\|_\infty + \mathrm{O}(\mathbf{u}^2), \tag{23}$$
$$\|\mathrm{fl}(\widehat{G}) - \widehat{G}\|_\infty \lesssim 2c_n \mathbf{u}\gamma \kappa_\infty(A_\gamma)\, \|W_\gamma^{-1}\|_1 \|\widehat{G}_\gamma\|_\infty + c_n \mathbf{u} \kappa_1(W_\gamma)\, \|\mathrm{fl}(\widehat{G})\|_\infty, \tag{24}$$
$$\|\mathrm{fl}(\widehat{H}) - \widehat{H}\|_\infty \leqslant 2c_n \mathbf{u}\gamma \kappa_\infty(A_\gamma)\, \|W_\gamma^{-1}\|_1 \|\widehat{H}_\gamma\|_\infty + c_n \mathbf{u} \kappa_1(W_\gamma)\, \|\mathrm{fl}(\widehat{H})\|_\infty, \tag{25}$$

where $\kappa_1(W_\gamma) \equiv \|W_\gamma\|_1 \|W_\gamma^{-1}\|_1$, $\kappa_\infty(W_\gamma) \equiv \|W_\gamma\|_\infty \|W_\gamma^{-1}\|_\infty$ and $\kappa_\infty(A_\gamma) \equiv \|A_\gamma\|_\infty \|A_\gamma^{-1}\|_\infty$.

In order to control the forward error bounds of $\widehat{A}$, $\widehat{G}$ and $\widehat{H}$, we consider the following min–max optimization problem, to determine an optimal value $\hat{\gamma} > 0$:

$$\min_{\gamma > 0} F(\gamma) \equiv \max_{i=1,2,3} \{f_i(\gamma)\}, \tag{26}$$

where $f_1(\gamma) := \gamma \kappa_\infty(W_\gamma)$, $f_2(\gamma) := \gamma \kappa_\infty(A_\gamma)$ and $f_3(\gamma) := \kappa_1(W_\gamma)$. Since the condition numbers $\kappa_\infty(W_\gamma)$, $\kappa_\infty(A_\gamma)$ and $\kappa_1(W_\gamma)$ approach 1 as $\gamma \to \infty$, it follows that $F(\gamma)$ becomes unbounded as $\gamma \to \infty$. Extensive numerical experiments on randomly generated matrices indicate that $F(\gamma)$ is a strictly convex function in the neighborhood of the optimal $\hat{\gamma}$ where the global minimum of $F(\gamma)$ occurs. For illustration, we report a sample of graphs of $f_1(\gamma)$, $f_2(\gamma)$, $f_3(\gamma)$ and $F(\gamma)$ in Figs. 1 and 2. From Theorem 2.1, we know that if γ approaches 0 and ∞, the symplectic matrix pair $(\widehat{\mathcal{N}}, \widehat{\mathcal{L}})$ has eigenvalues close to one, leading to very slow convergence of the SDA. This can be avoided through the min-max optimization problem (26).

We can apply the Fibonacci search method to compute an approximate value of $\hat{\gamma}$, see, e.g., [10–p. 272]. Our experience indicates that three to five iterations of Fibonacci search are adequate to obtain a suboptimal yet acceptable approximation to $\hat{\gamma}$.

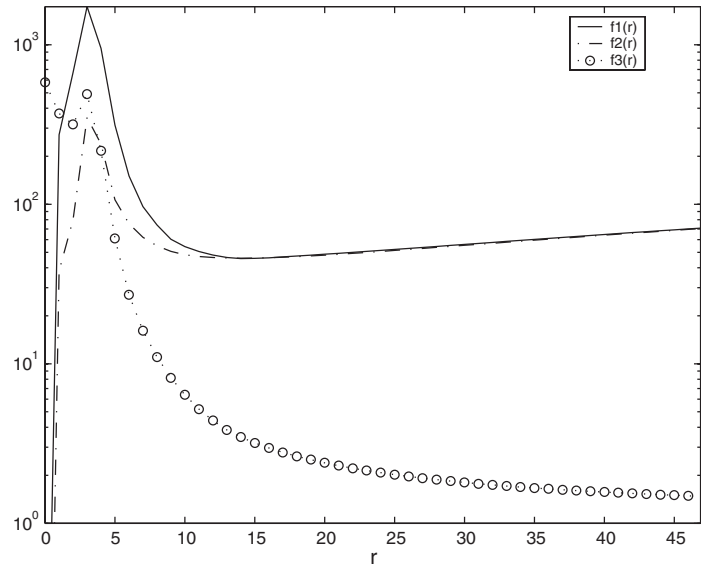

Fig. 1. The graphs of functions f_1, f_2 and f_3.

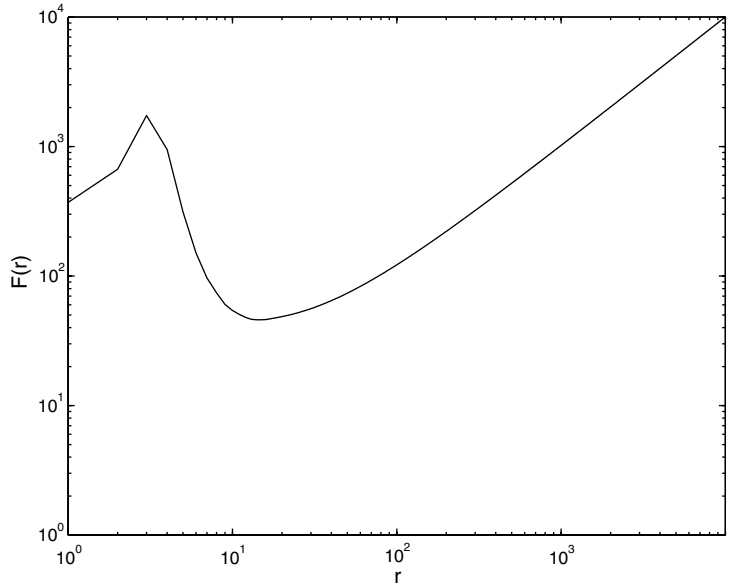

Fig. 2. The graph of $F(\gamma)$.

3.2. Symmetry of $\widehat{G}_0$ and $\widehat{H}_0$

If the matrices G and H are of low-rank, say $G = gg^{\mathrm{T}} \geqslant 0$ and $H = h^{\mathrm{T}}h \geqslant 0$, then so are $\widehat{G}_0$ and $\widehat{H}_0$. Indeed, by using the Sherman–Morrison–Woodbury formula (SMWF) twice, it can be seen that

$$\begin{aligned}
\widehat{G}_0 &= 2\gamma A_\gamma^{-1} G\big(A_\gamma^{\mathrm{T}} + HA_\gamma^{-1}G\big)^{-1} \\
&= 2\gamma A_\gamma^{-1} gg^{\mathrm{T}}\big(A_\gamma^{\mathrm{T}} + hh^{\mathrm{T}}A_\gamma^{-1}gg^{\mathrm{T}}\big)^{-1} \\
&= 2\gamma\big[A_\gamma^{-1}gg^{\mathrm{T}}\big(A_\gamma^{-\mathrm{T}} - A_\gamma^{-\mathrm{T}}hh^{\mathrm{T}}\big(I + A_\gamma^{-1}gg^{\mathrm{T}}A_\gamma^{-\mathrm{T}}hh^{\mathrm{T}}\big)^{-1} \\
&\qquad \times A_\gamma^{-1}gg^{\mathrm{T}}A_\gamma^{-\mathrm{T}}\big)\big] \\
&= 2\gamma\big\{\big(A_\gamma^{-1}g\big)\big[I - \big(A_\gamma^{-1}g\big)^{\mathrm{T}}hh^{\mathrm{T}}\big(I + \big(A_\gamma^{-1}g\big)\big(A_\gamma^{-1}g\big)^{\mathrm{T}}hh^{\mathrm{T}}\big)^{-1} \\
&\qquad \times \big(A_\gamma^{-1}g\big)\big]\big(A_\gamma^{-1}g\big)^{\mathrm{T}}\big\} \\
&= 2\gamma\big\{\big(A_\gamma^{-1}g\big)\big(I + (A_\gamma^{-1}g)^{\mathrm{T}}hh^{\mathrm{T}}\big(A_\gamma^{-1}g\big)\big)^{-1}\big(A_\gamma^{-1}g\big)^{\mathrm{T}}\big\} \\
&= 2\gamma\big\{\big(A_\gamma^{-1}g\big)\big(K_g^{\mathrm{T}}K_g\big)^{-1}\big(A_\gamma^{-1}g\big)^{\mathrm{T}}\big\} \quad \text{(Cholesky decomposition)} \\
&= 2\gamma\big(A_\gamma^{-1}gK_g^{-1}\big)\big(A_\gamma^{-1}gK_g^{-1}\big)^{\mathrm{T}}.
\end{aligned}$$

Similarly, by applying the same techniques, we also have

$$\begin{aligned}
\widehat{H}_0 &= 2\gamma\big(A_\gamma^{\mathrm{T}} + HA_\gamma^{-1}G\big)^{-1}HA_\gamma^{-1} \\
&= 2\gamma\big(A_\gamma^{\mathrm{T}} + h^{\mathrm{T}}hA_\gamma^{-1}gg^{\mathrm{T}}\big)^{-1}h^{\mathrm{T}}hA_\gamma^{-1} \\
&= 2\gamma\big[A_\gamma^{-\mathrm{T}} - A_\gamma^{-\mathrm{T}}h^{\mathrm{T}}hA_\gamma^{-1}\big(I + gg^{\mathrm{T}}A_\gamma^{-\mathrm{T}}h^{\mathrm{T}}hA_\gamma^{-1}\big)^{-1}gg^{\mathrm{T}}A_\gamma^{-T}\big]h^{\mathrm{T}}hA_\gamma^{-1} \\
&= 2\gamma\big\{(hA_\gamma^{-1})^{\mathrm{T}}\big[I - (hA_\gamma^{-1})\big(I + gg^{\mathrm{T}}A_\gamma^{-\mathrm{T}}h^{\mathrm{T}}hA_\gamma^{-1}\big)^{-1} \\
&\qquad \times gg^{\mathrm{T}}(hA_\gamma^{-1})^{\mathrm{T}}\big](hA_\gamma^{-1})\big\} \\
&= 2\gamma\big\{(hA_\gamma^{-1})^{\mathrm{T}}\big(I + (hA_\gamma^{-1})gg^{\mathrm{T}}(hA_\gamma^{-1})^{\mathrm{T}}\big)^{-1}(hA_\gamma^{-1})\big\} \\
&= 2\gamma\big\{(hA_\gamma^{-1})^{\mathrm{T}}\big(K_hK_h^{\mathrm{T}}\big)^{-1}(hA_\gamma^{-1})\big\} \quad \text{(Cholesky decomposition)} \\
&= 2\gamma(K_h^{-1}hA_\gamma^{-1})^{\mathrm{T}}(K_h^{-1}hA_\gamma^{-1}).
\end{aligned}$$

3.3. Computation of $\widehat{A}_j$, $\widehat{G}_j$ and $\widehat{H}_j$

We now propose a structured and efficient procedure for the computation of $\widehat{A}_j$, $\widehat{G}_j$ and $\widehat{H}_j$ in the SDA algorithm, respectively, where $\widehat{G}_0 = \widehat{B}_0\widehat{B}_0^{\mathrm{T}} \geqslant 0$, $\widehat{H}_0 = \widehat{C}_0^{\mathrm{T}}\widehat{C}_0 \geqslant 0$ are FRDs. For $j = 0, 1, 2, \ldots$, we let $W_j \equiv (I + \widehat{G}_j\widehat{H}_j)^{-1}$. It is easily seen that $\widehat{H}_jW_j = W_j^{\mathrm{T}}\widehat{H}_j$ and $\widehat{G}_jW_j^{\mathrm{T}} = W_j\widehat{G}_j$ are s.p.s.d. for each $j \geqslant 1$. By the SMWF we can derive the formulae

$$W_j = \big(I + \widehat{G}_j\widehat{H}_j\big)^{-1} = I - \widehat{B}_j\big(I + \widehat{B}_j^{\mathrm{T}}\widehat{H}_j\widehat{B}_j\big)^{-1}\widehat{B}_j^{\mathrm{T}}\widehat{H}_j, \tag{27}$$

$$\widehat{G}_j W_j^{\mathrm{T}} = \widehat{G}_j - \widehat{G}_j\widehat{C}_j^{\mathrm{T}}\big(I + \widehat{C}_j\widehat{G}_j\widehat{C}_j^{\mathrm{T}}\big)^{-1}\widehat{C}_j\widehat{G}_j = \widehat{B}_j\big(I + \widehat{B}_j^{\mathrm{T}}\widehat{H}_j\widehat{B}_j\big)^{-1}\widehat{B}_j^{\mathrm{T}}, \tag{28}$$

$$W_j^{\mathrm{T}}\widehat{H}_j = \widehat{H}_j - \widehat{H}_j\widehat{B}_j\big(I + \widehat{B}_j^{\mathrm{T}}\widehat{H}_j\widehat{B}_j\big)^{-1}\widehat{B}_j^{\mathrm{T}}\widehat{H}_j = \widehat{C}_j^{\mathrm{T}}\big(I + \widehat{C}_j\widehat{G}_j\widehat{C}_j^{\mathrm{T}}\big)^{-1}\widehat{C}_j. \tag{29}$$

When the matrices B and C start with low ranks in (1), we can improve the efficiency of our computation further by the following compression process. Compute the Cholesky decomposition of the s.p.d. matrices $W_{G,j} \equiv \big(I + \widehat{B}_j^{\mathrm{T}}\widehat{H}_j\widehat{B}_j\big) = K_{B,j}^{\mathrm{T}}K_{B,j}$ and $W_{H,j} \equiv \big(I + \widehat{C}_j\widehat{G}_j\widehat{C}_j^{\mathrm{T}}\big) = K_{C,j}K_{C,j}^{\mathrm{T}}$, respectively. For $j = 0, 1, 2, \ldots$, application of (27)–(29) leads to

$$\widehat{A}_{j+1} = \widehat{A}_j^2 - \widehat{A}_j\widehat{B}_j\big(I + \widehat{B}_j^{\mathrm{T}}\widehat{H}_j\widehat{B}_j\big)^{-1}\widehat{B}_j^{\mathrm{T}}\widehat{H}_j\widehat{A}_j, \tag{30}$$

$$\begin{aligned}\widehat{G}_{j+1} &= \widehat{G}_j + \widehat{A}_j\widehat{B}_j\big(I + \widehat{B}_j^{\mathrm{T}}\widehat{H}_j\widehat{B}_j\big)^{-1}\widehat{B}_j^{\mathrm{T}}\widehat{A}_j^{\mathrm{T}} \\ &= \big[\widehat{B}_j, \quad \widehat{A}_j\widehat{B}_jK_{B,j}^{-1}\big]\begin{bmatrix}\widehat{B}_j^{\mathrm{T}} \\ K_{B,j}^{-\mathrm{T}}\widehat{B}_j^{\mathrm{T}}\widehat{A}_j^{\mathrm{T}}\end{bmatrix} \equiv \widehat{B}_{j+1}\widehat{B}_{j+1}^{\mathrm{T}} \geqslant 0 \quad \text{(FRD)}\end{aligned} \tag{31}$$

and

$$\begin{aligned}\widehat{H}_{j+1} &= \widehat{H}_j + \widehat{A}_j^{\mathrm{T}}\widehat{C}_j^{\mathrm{T}}\big(I + \widehat{C}_j\widehat{G}_j\widehat{C}_j^{\mathrm{T}}\big)^{-1}\widehat{C}_j\widehat{A}_j \\ &= \big[\widehat{C}_j^{\mathrm{T}}, \quad \widehat{A}_j^{\mathrm{T}}\widehat{C}_j^{\mathrm{T}}K_{C,j}^{-\mathrm{T}}\big]\begin{bmatrix}\widehat{C}_j \\ K_{C,j}^{-1}\widehat{C}_j\widehat{A}_j\end{bmatrix} \equiv \widehat{C}_{j+1}^{\mathrm{T}}\widehat{C}_{j+1} \geqslant 0 \quad \text{(FRD)},\end{aligned} \tag{32}$$

where $\widehat{B}_{j+1}$ and $\widehat{C}_{j+1}^{\mathrm{T}}$ are the full column rank compressions of $\big[\widehat{B}_j, \ \widehat{A}_j\widehat{B}_jK_{B,j}^{-1}\big]$ and $\big[\widehat{C}_j^{\mathrm{T}}, \ \widehat{A}_j^{\mathrm{T}}\widehat{C}_j^{\mathrm{T}}K_{C,j}^{-\mathrm{T}}\big]$, respectively. In general, $\mathrm{rank}(\widehat{B}_{j+1}) > \mathrm{rank}(\widehat{B}_j)$ and $\mathrm{rank}(\widehat{C}_{j+1}) > \mathrm{rank}(\widehat{C}_j)$, and the compression process becomes unprofitable when the ranks of $\widehat{B}_{j+1}$ and $\widehat{C}_{j+1}$ approach n.

3.4. Error analysis of SDA

We consider the forward error bounds of the computed matrices $\widehat{A}_{j+1}$, $\widehat{G}_{j+1}$ and $\widehat{H}_{j+1}$ in the SDA algorithm for one iterative step j. Since $K_{B,j}$ and $K_{C,j}$ are the computed Cholesky factors of matrices $W_{G,j}$ and $W_{H,j}$, respectively, it follows that:

$$\begin{aligned}&\widehat{K}_B \equiv \mathrm{fl}(K_{B,j}^{-\mathrm{T}}\widehat{B}_j^{\mathrm{T}}) = K_{B,j}^{-\mathrm{T}}\widehat{B}_j^{\mathrm{T}} + \Delta E_1, \\ &\quad |\Delta E_1| \leqslant c_1\mathbf{u}\,|K_{B,j}^{-\mathrm{T}}|\,|K_{B,j}^{\mathrm{T}}|\,|\widehat{K}_B|\end{aligned} \tag{33}$$

and

$$\begin{aligned}&\widehat{K}_C \equiv \mathrm{fl}(K_{C,j}^{-1}\widehat{C}_j) = K_{C,j}^{-1}\widehat{C}_j + \Delta\widetilde{E}_1, \\ &\quad |\Delta\widetilde{E}_1| \leqslant \tilde{c}_1\mathbf{u}\,|K_{C,j}^{-1}|\,|K_{C,j}|\,|\widehat{K}_C|,\end{aligned} \tag{34}$$

where c_1 and $\tilde{c}_1$ are modest constants. Therefore, we have

$$\begin{aligned} \mathrm{fl}(K_{B,j}^{-\mathrm{T}}\widehat{B}_j^{\mathrm{T}}\widehat{A}_j^{\mathrm{T}}) = \mathrm{fl}(\widehat{K}_B\widehat{A}_j^{\mathrm{T}}) = K_{B,j}^{-\mathrm{T}}\widehat{B}_j^{\mathrm{T}}\widehat{A}_j^{\mathrm{T}} + \Delta E_2, \\ |\Delta E_2| \leqslant c_2\mathbf{u}\,|K_{B,j}^{-\mathrm{T}}|\,|K_{B,j}^{\mathrm{T}}|\,|\widehat{K}_B|\,|\widehat{A}_j^{\mathrm{T}}| \end{aligned} \tag{35}$$

and

$$\begin{aligned} \mathrm{fl}(K_{C,j}^{-1}\widehat{C}_j\widehat{A}_j) = \mathrm{fl}(\widehat{K}_C\widehat{A}_j) = K_{C,j}^{-1}\widehat{C}_j\widehat{A}_j + \Delta\widetilde{E}_2, \\ |\Delta\widetilde{E}_2| \leqslant \tilde{c}_2\mathbf{u}\,|K_{C,j}^{-1}|\,|K_{C,j}|\,|\widehat{K}_C|\,|\widehat{A}_j|, \end{aligned} \tag{36}$$

where c_2 and $\tilde{c}_2$ are modest constants.

If $\mathrm{rank}\big(\widehat{B}_j^{\mathrm{T}}\big) = \ell$, then from Theorem 18.4 of [32] and (31), there exist an orthogonal matrix $Q_B \in \mathbb{R}^{2\ell\times 2\ell}$ and a computed upper triangular matrix $\mathrm{fl}\big(\widehat{B}_{j+1}^{\mathrm{T}}\big)$ with full row rank, such that

$$\begin{bmatrix} \widehat{B}_j^{\mathrm{T}} \\ \mathrm{fl}\big(\widehat{K}_B\widehat{A}_j^{\mathrm{T}}\big) \end{bmatrix} + \begin{bmatrix} \Delta B_1 \\ \Delta B_2 \end{bmatrix} = Q_B \begin{bmatrix} \mathrm{fl}\big(\widehat{B}_{j+1}^{\mathrm{T}}\big) \\ 0 \end{bmatrix}, \tag{37}$$

where $|\Delta B_j| \leqslant c_3\mathbf{u}\; G_\ell\big(|\widehat{B}_j^{\mathrm{T}}| + |K_{B,j}^{-\mathrm{T}}|\;|\widehat{B}_j^{\mathrm{T}}|\;|\widehat{A}_j^{\mathrm{T}}|\big)$ for $j = 1, 2$, with c_3 being a modest constant and $\|G_\ell\|_{\mathrm{F}} = \frac{1}{2}$.

From (35) and (37), we have

$$\begin{bmatrix} \widehat{B}_j^{\mathrm{T}} \\ K_{B,j}^{-\mathrm{T}}\widehat{B}_j^{\mathrm{T}}\widehat{A}_j^{\mathrm{T}} \end{bmatrix} + \begin{bmatrix} \Delta B_1 \\ \Delta\widetilde{B}_2 \end{bmatrix} = Q_B \begin{bmatrix} \mathrm{fl}\big(\widehat{B}_{j+1}^{\mathrm{T}}\big) \\ 0 \end{bmatrix}, \tag{38}$$

where $|\Delta\widetilde{B}_2| \leqslant c_2\mathbf{u}\,|K_{B,j}^{-\mathrm{T}}|\,|K_{B,j}^{\mathrm{T}}|\,|\widehat{K}_B|\,|\widehat{A}_j^{\mathrm{T}}| + c_3\mathbf{u}\; G_\ell\big(|\widehat{B}_j^{\mathrm{T}}| + |K_{B,j}^{-\mathrm{T}}|\,|\widehat{B}_j^{\mathrm{T}}|\,|\widehat{A}_j^{\mathrm{T}}|\big)$. Pre-multiplying both sides of (38) by Q_B^{T}, it follows that:

$$\begin{bmatrix} \widehat{B}_{j+1}^{\mathrm{T}} \\ 0 \end{bmatrix} + Q_B^{\mathrm{T}} \begin{bmatrix} \Delta B_1 \\ \Delta\widetilde{B}_2 \end{bmatrix} = \begin{bmatrix} \mathrm{fl}\big(\widehat{B}_{j+1}^{\mathrm{T}}\big) \\ 0 \end{bmatrix} \tag{39}$$

and we deduce that

$$\|\mathrm{fl}\big(\widehat{B}_{j+1}^{\mathrm{T}}\big) - \widehat{B}_{j+1}^{\mathrm{T}}\|_{\mathrm{F}} \leqslant c_4\mathbf{u}\,\|\widehat{B}_j\|_{\mathrm{F}} + c_5\mathbf{u}\,\kappa_{\mathrm{s}}\big(K_{B,j}^{\mathrm{T}}\big)\,\|\widehat{K}_b\|_{\mathrm{F}}\|\widehat{A}_j\|_{\mathrm{F}}, \tag{40}$$

where c_4 and c_5 are modest constants, and $\kappa_{\mathrm{s}}\big(K_{B,j}^{\mathrm{T}}\big) \equiv \|\,|K_{B,j}^{-\mathrm{T}}|\;|K_{B,j}^{\mathrm{T}}|\,\|_\infty$ is the Skeel condition number of $K_{B,j}^{\mathrm{T}}$. Furthermore, applying a similar argument with the help of (36), we can derive that

$$\|\mathrm{fl}(\widehat{C}_{j+1}) - \widehat{C}_{j+1}\|_{\mathrm{F}} \leqslant c_6\mathbf{u}\,\|\widehat{C}_j\|_{\mathrm{F}} + c_7\mathbf{u}\,\kappa_{\mathrm{s}}(K_{C,j})\,\|\widehat{K}_C\|_{\mathrm{F}}\|\widehat{A}_j\|_{\mathrm{F}}, \tag{41}$$

where c_6 and c_7 are modest constants.

On the other hand, it follows from (33) that:

$$\begin{aligned} \mathrm{fl}\big(K_{B,j}^{-\mathrm{T}}\widehat{B}_j^{\mathrm{T}}\widehat{H}_j\widehat{A}_j\big) = K_{B,j}^{-\mathrm{T}}\widehat{B}_j^{\mathrm{T}}\widehat{H}_j\widehat{A}_j + \Delta E_3, \\ |\Delta E_3| \leqslant c_8\mathbf{u}\,|K_{B,j}^{-\mathrm{T}}|\,|K_{B,j}^{\mathrm{T}}|\,|\widehat{K}_B|\,|\widehat{H}_j|\,|\widehat{A}_j|, \end{aligned} \tag{42}$$

where c_8 is a modest constant. From (35) and (42), the forward error bound of computing $\widehat{A}_{j+1}$ is

$$\begin{aligned}&\|\mathrm{fl}(\widehat{A}_{j+1}) - \widehat{A}_{j+1}\|_{\mathrm{F}}\\ &\quad\leqslant c_9\mathbf{u}\,\|\widehat{A}_j\|_{\mathrm{F}}^2 + c_{10}\mathbf{u}\,\kappa_{\mathrm{s}}\big(K_{B,j}^{\mathrm{T}}\big)\,\|\widehat{B}_j\|_{\mathrm{F}}^2\|\widehat{K}_{B,j}^{-1}\|_{\mathrm{F}}^2\|\widehat{H}_j\|_{\mathrm{F}}\|\widehat{A}_j\|_{\mathrm{F}}^2,\end{aligned} \tag{43}$$

where c_9 and c_{10} are modest constants.

When the Skeel condition numbers $\kappa_{\mathrm{s}}\big(K_{B,j}^{\mathrm{T}}\big)$ and $\kappa_{\mathrm{s}}(K_{C,j})$ in (40) and (41) are bounded from above by acceptable numbers, the accumulation of error will be dampened by the fast rate of convergence at the final stage of the iterative process. Danger, if any, lies in the early stage of the process before the $\lambda_n^{2^j}$ convergence factor dominates. It is unlikely to have any ill-effect, as the accumulated error in the matrix additions and multiplications should be of magnitude around a small multiple of the machine accuracy.

As the SSF properties are preserved in the SDA, any error will be a structured one, only pushing the iteration towards a solution of a neighboring SSF system. Thus the algorithm is stable in this sense, when the errors are not too large and when stabilizability and detectability are maintained. For large js, as $\widehat{A}_j \to 0$, $\widehat{G}_j$ and $\widehat{H}_j$ converge to the unique s.p.s.d. solutions of (14) and (12), respectively. Danger again will only comes at the initial stage of the iteration. Corresponding checks may be prudent in the algorithm.

4. SDA_m

A matrix A is persymmetric when A is symmetric with respect to the main antidiagonal [30–p. 193]. When the DARE transformed from the CARE (1) has the additional property that the initial data $\widehat{A}_0$, $\widehat{G}_0 = \widehat{H}_0 \in \mathbb{R}^{2\ell\times 2\ell}$ are symmetric and persymmetric, the additional structure can be preserved in a modified version of the SDA (SDA_m). For simplicity, we consider only when $\gamma = 1$. This doubly symmetric type of DAREs appear in the Examples 10 and 17 of Section 5 (originally from [15]).

For convenience, in the SDA, we denote for $j = 1, 2, \ldots$

$$\begin{aligned}&A \equiv \widehat{A}_j, \quad G \equiv \widehat{G}_j = \widehat{H}_j,\\ &A_+ \equiv \widehat{A}_{j+1}, \quad G_+ \equiv \widehat{G}_{j+1} = \widehat{H}_{j+1}.\end{aligned} \tag{44}$$

Since A, $G = H$ are symmetric and persymmetric of even order, we write

$$A = \begin{bmatrix} a_1 & a_2\zeta \\ \zeta a_2 & \zeta a_1\zeta \end{bmatrix}, \quad G = \begin{bmatrix} g_1 & g_2\zeta \\ \zeta g_2 & \zeta g_1\zeta \end{bmatrix}, \tag{45}$$

where a_1, a_2, g_1 and $g_2 \in \mathbb{R}^{\ell\times\ell}$ are symmetric and $\zeta = [e_\ell, \ldots, e_1]$ with e_j being the jth column of I_ℓ. In the SDA, we shall show that $\widehat{A}$, $\widehat{G}$ and $\widehat{H}$ are also symmetric and persymmetric with $\widehat{G} = \widehat{H}$, with

$$A_+ = A(I + G^2)^{-1}A = \begin{bmatrix} \widehat{a}_1 & \widehat{a}_2\zeta \\ \zeta\widehat{a}_2 & \zeta\widehat{a}_1\zeta \end{bmatrix}, \tag{46}$$

$$G_+ = G + AG(I + G^2)^{-1}A^{\mathrm{T}} = \begin{bmatrix} \widehat{g}_1 & \widehat{g}_2\zeta \\ \zeta\widehat{g}_2 & \zeta\widehat{g}_1\zeta \end{bmatrix}. \tag{47}$$

Let

$$q_1 \equiv g_1 + g_2, \quad q_2 \equiv g_1 - g_2, \quad \alpha_1 \equiv a_1 + a_2, \quad \alpha_2 \equiv a_1 - a_2. \tag{48}$$

Simple manipulation leads to

$$\widehat{a}_1 = \frac{1}{2}\left[\alpha_1(I + q_1^2)^{-1}\alpha_1 + \alpha_2(I + q_2^2)^{-1}\alpha_2\right], \tag{49}$$

$$\widehat{a}_2 = \frac{1}{2}\left[\alpha_1(I + q_1^2)^{-1}\alpha_1 - \alpha_2(I + q_2^2)^{-1}\alpha_2\right], \tag{50}$$

$$\widehat{g}_1 = g_1 + \frac{1}{2}\left[\alpha_1(I + q_1^2)^{-1}q_1\alpha_1 + \alpha_2(I + q_2^2)^{-1}q_2\alpha_2\right], \tag{51}$$

$$\widehat{g}_2 = g_2 + \frac{1}{2}\left[\alpha_1(I + q_1^2)^{-1}q_1\alpha_1 - \alpha_2(I + q_2^2)^{-1}q_2\alpha_2\right]. \tag{52}$$

Furthermore let

$$q_1 = [U_1, V_1]\begin{bmatrix} \Sigma_1 & 0 \\ 0 & -\Gamma_1 \end{bmatrix}\begin{bmatrix} U_1^{\mathrm{T}} \\ V_1^{\mathrm{T}} \end{bmatrix}, \quad q_2 = [U_2, V_2]\begin{bmatrix} \Sigma_2 & 0 \\ 0 & -\Gamma_2 \end{bmatrix}\begin{bmatrix} U_2^{\mathrm{T}} \\ V_2^{\mathrm{T}} \end{bmatrix} \tag{53}$$

be the spectral decompositions of q_1 and q_2, respectively, with Σ_1, Γ_1, Σ_2 and Γ_2 being nonnegative diagonal matrices. Then $\widehat{a}_1$, $\widehat{a}_2$, $\widehat{g}_1$ and $\widehat{g}_2$ in (49)–(52) can be computed by the following symmetric forms:

$$\begin{aligned} &\xi_1 \equiv \alpha_1 U_1(I + \Sigma_1^2)^{-1}U_1^{\mathrm{T}}\alpha_1 - \alpha_1 V_1(I + \Gamma_1^2)^{-1}V_1^{\mathrm{T}}\alpha_1, \\ &\xi_2 \equiv \alpha_2 U_2(I + \Sigma_2^2)^{-1}U_2^{\mathrm{T}}\alpha_2 - \alpha_2 V_2(I + \Gamma_2^2)^{-1}V_2^{\mathrm{T}}\alpha_2, \\ &\widehat{a}_1 = \frac{1}{2}\{\xi_1 + \xi_2\}, \quad \widehat{a}_2 = \frac{1}{2}\{\xi_1 - \xi_2\}; \end{aligned} \tag{54}$$

$$\begin{aligned} &\eta_1 \equiv \alpha_1 U_1(I + \Sigma_1^2)^{-1}\Sigma_1 U_1^{\mathrm{T}}\alpha_1 - \alpha_1 V_1(I + \Gamma_1^2)^{-1}\Gamma_1 V_1^{\mathrm{T}}\alpha_1, \\ &\eta_2 \equiv \alpha_2 U_2(I + \Sigma_2^2)^{-1}\Sigma_2 U_2^{\mathrm{T}}\alpha_2 - \alpha_2 V_2(I + \Gamma_2^2)^{-1}\Gamma_2 V_2^{\mathrm{T}}\alpha_2, \\ &\widehat{g}_1 = g_1 + \frac{1}{2}\{\eta_1 + \eta_2\}, \quad \widehat{g}_2 = g_2 + \frac{1}{2}\{\eta_1 - \eta_2\}. \end{aligned} \tag{55}$$

The SDA_m computes $\widehat{A}$, $\widehat{G}$ in (46) and (47) using the symmetric forms (54) and (55) and considerably improves the accuracy of Examples 10 and 17 in the next section.

5. Numerical examples

For the tables in the following examples, data for various methods are lists in columns with obvious headings. The heading "care" is for the care command in MATLAB [40], "MSGM" is for the matrix sign function method [21], and "SDA" (or "SDA_m") stands for our SDA (or SDA_m) algorithm. There is no iteration numbers to report for care and an '$*$' in the tables indicates a failure of convergence in obtaining a solution. In the graphs, "ratio_care" and "ratio_MSGM" are the ratio of the CPU-times for care and MSGM to that of the SDA, respectively. For the comparison of residuals, the "normalized" residual (NRes) formula is applied in the numerical examples, i.e.,

$$\mathrm{NRes} \equiv \frac{\|A^{\mathrm{T}}\tilde{X} + \tilde{X}A^{\mathrm{T}} - \tilde{X}G\tilde{X} + H\|}{\|A^{\mathrm{T}}\tilde{X}\| + \|\tilde{X}A^{\mathrm{T}}\| + \|\tilde{X}G\tilde{X}\| + \|H\|},$$

where $\widetilde{X}$ is an approximate solution and $\|\cdot\|$ denotes the 2-norm for matrices.

Some numerical examples from [15] involved very large data sets, which have not been repeated here. Twenty examples were presented in [23]. We retain the numbering of examples in [23], comment upon all of them but present only five representative ones in this paper.

In the MSGM, the scaling strategy suggested in [21] was implemented. For a fairer comparison, similar convergence criteria were used in all the methods and the solutions were not refined.

All computations were performed using MATLAB/Version 6.0 on a Compaq/DS20 workstation. The machine precision is 2.22×10^{-16}.

Example 5. The example is identical to Example 5 of [15], which has been presented originally in [45]. This is a 9th-order continuous state space model of a tubular ammonia reactor. The actual system matrices are

$$A = \begin{bmatrix} -4.019 & 5.12 & 0 & 0 & -2.082 & 0 & 0 & 0 & 0.87 \\ -0.346 & 0.986 & 0 & 0 & -2.34 & 0 & 0 & 0 & 0.97 \\ -7.909 & 15.407 & -4.096 & 0 & -6.45 & 0 & 0 & 0 & 2.68 \\ -21.816 & 35.606 & -0.339 & -3.87 & -17.8 & 0 & 0 & 0 & 7.39 \\ -60.196 & 98.188 & -7.907 & 0.34 & -53.008 & 0 & 0 & 0 & 20.4 \\ 0 & 0 & 0 & 0 & 94.0 & -147.2 & 0 & 53.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 94.0 & -147.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 12.8 & 0 & -31.6 & 0 \\ 0 & 0 & 0 & 0 & 12.8 & 0 & 0 & 18.8 & -31.6 \end{bmatrix},$$

$$B^{\mathrm{T}} = \begin{bmatrix} 0.010 & 0.003 & 0.009 & 0.024 & 0.068 & 0 & 0 & 0 & 0 \\ -0.011 & -0.021 & -0.059 & -0.162 & -0.445 & 0 & 0 & 0 & 0 \\ -0.151 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$H = I_9, \quad R = I_3.$$

The numerical results are given in Table 1.

Table 1
Results for Example 5

	SDA	MSGM	care
NRes	1.68×10^{-15}	1.73×10^{-13}	4.64×10^{-14}
Iter. no.	9	8	–

Table 2
Results for Example 6

	SDA	MSGM	care
NRes	5.78×10^{-13}	3.11×10^{-8}	1.91×10^{-12}
Iter. no.	10	9	–

Example 6. The example is identical to Example 6 of [15], which has been presented originally in [26]. This control problem for a J-100 jet engine is a special case of a multivariable servomechanism problem. To save space, we shall not list the system matrices here. We report the numerical results in Table 2.

Example 10. The example is identical to Example 10 of [15], which has been presented originally in [6]. Here, the system matrices are

$$A = \begin{bmatrix} \varepsilon + 1 & 1 \\ 1 & \varepsilon + 1 \end{bmatrix}, \quad G = I_2, \quad H = \begin{bmatrix} \varepsilon^2 & 0 \\ 0 & \varepsilon^2 \end{bmatrix}.$$

The exact stabilizing solution X is given by

$$\begin{aligned} x_{11} = x_{22} &= \frac{1}{2}\left[2(\varepsilon + 1) + \sqrt{2(\varepsilon + 1)^2 + 2} + \sqrt{2}\varepsilon\right], \\ x_{12} = x_{21} &= x_{11}/[x_{11} - (\varepsilon + 1)]. \end{aligned} \tag{56}$$

The corresponding DARE is doubly symmetric and the SDA_m was applied (see details in Section 4). The numerical results with $\varepsilon = 1, 10^{-3}, 10^{-5}$ and 10^{-7} are given in Table 3.

Example 11. The example is identical to Example 11 of [15], which has been presented originally in [35]. This example represents an algebraic Riccati equation arising from a H_∞-control problem [49]. Let

$$A = \begin{bmatrix} 3 - \varepsilon & 1 \\ 4 & 2 - \varepsilon \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad R = 1, \quad H = \begin{bmatrix} 4\varepsilon - 11 & 2\varepsilon - 5 \\ 2\varepsilon - 5 & 2\varepsilon - 2 \end{bmatrix}.$$

The matrix

$$X = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

Table 3
Results for Example 10

		SDA	SDA_m	MSGM	care
$\varepsilon = 1$	NRes	0.00×10^{0}	0.00×10^{0}	4.69×10^{-16}	9.36×10^{-17}
	Rel. err.	1.96×10^{-16}	1.96×10^{-16}	8.80×10^{-16}	3.83×10^{-16}
	Iter. no.	4	4	2	–
$\varepsilon = 10^{-3}$	NRes	1.58×10^{-14}	1.43×10^{-16}	1.11×10^{-13}	9.20×10^{-17}
	Rel. err.	1.82×10^{-11}	2.22×10^{-16}	2.22×10^{-13}	4.08×10^{-16}
	Iter. no.	16	13	12	–
$\varepsilon = 10^{-5}$	NRes	2.28×10^{-12}	1.11×10^{-16}	1.07×10^{-11}	5.53×10^{-17}
	Rel. err.	7.16×10^{-7}	1.76×10^{-16}	2.14×10^{-11}	2.60×10^{-16}
	Iter. no.	22	19	18	–
$\varepsilon = 10^{-7}$	NRes	1.49×10^{-10}	1.32×10^{-16}	3.31×10^{-9}	2.06×10^{-17}
	Rel. err.	6.04×10^{-8}	4.44×10^{-16}	6.63×10^{-9}	1.36×10^{-16}
	Iter. no.	12	26	20	–

Table 4
Results for Example 11

		SDA	MSGM	care
$\varepsilon = 1$	NRes	0.00×10^{0}	1.69×10^{-16}	1.97×10^{-16}
	Rel. err.	1.26×10^{-16}	1.25×10^{-15}	9.68×10^{-16}
	Iter. no.	5	2	–
$\varepsilon = 0$	NRes	3.06×10^{-16}	$*$	5.06×10^{-17}
	Rel. err.	2.66×10^{-9}	$*$	7.68×10^{-9}
	Iter. no.	28	$*$	–

is the stabilizing solution for $\varepsilon > 0$. For $\varepsilon = 0$, the solution X is obtained by an H-invariant *Lagrangian* subspace, i.e., a solution in the sense of H_∞-control. The numerical results with $\varepsilon = 1, 0$ are given in Table 4.

Example 12. The example is identical to Example 12 of [15], which has been presented originally in [34]. Let

$$V = I - \frac{2}{3} v v^{\mathrm{T}}, \quad v^{\mathrm{T}} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}; \quad A_0 = \varepsilon \operatorname{diag}(1, 2, 3),$$
$$H_0 = \operatorname{diag}(\varepsilon^{-1}, 1, \varepsilon);$$

we have

$$A = V A_0 V, \quad G = \varepsilon^{-1} I_3, \quad H = V H_0 V.$$

The solution is

$$X = V \operatorname{diag}(x_1, x_2, x_3) V,$$

Table 5
Results for Example 12

		SDA	MSGM	care
$\varepsilon = 1$	NRes	2.01×10^{-16}	1.78×10^{-15}	3.00×10^{-16}
	Rel. err.	4.33×10^{-16}	2.78×10^{-15}	5.03×10^{-16}
	Iter. no.	6	4	–
$\varepsilon = 10^6$	NRes	1.62×10^{-15}	2.22×10^{-4}	2.19×10^{-15}
	Rel. err.	2.58×10^{-15}	6.33×10^{-4}	4.92×10^{-15}
	Iter. no.	11	10	–

where

$$x_1 = \varepsilon^2 + \sqrt{\varepsilon^4 + 1}, \quad x_2 = 2\varepsilon^2 + \sqrt{4\varepsilon^4 + \varepsilon}, \quad x_3 = 3\varepsilon^2 + \sqrt{9\varepsilon^4 + \varepsilon^2}.$$

The numerical results with $\varepsilon = 1, 10^6$ are given in Table 5.

Example 15. The example is identical to Example 15 of [15], which has been presented originally in [39–Example 4] and [3]. This example arises from a mathematical model of position and velocity control for a string of high-speed vehicles. If N vehicles are to be controlled, the size of the system matrices will be $n = 2N - 1$, the number of control inputs will be $m = N$, and the number of outputs will be $p = N - 1$, respectively. The comparison of normalized residuals are reported in Table 6 for $N = 5, 20, 60, 100, 140$ and 180. Fig. 3 reports the comparison of CPU times for `care`, MSGN and the SDA.

Example 17. The example is identical to Example 17 of [15], which has been presented originally in [39–Example 6]. The system matrices are

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ 0 & \cdots & \cdots & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad R = r, \quad C^{\mathrm{T}} = \sqrt{q} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}.$$

It is known from [39] that $x_{1n} = \sqrt{qr}$. Therefore, we may use the relative error in x_{1n}, i.e., $\mathrm{RE} \equiv (|x_{1n} - \sqrt{qr}|)/\sqrt{qr}$, as an indicator of the accuracy of the results. The corresponding DARE is doubly symmetric and the SDA_m was applied (see details in Section 4). Table 7 reports the comparison of normalized residuals computed by SDA, SDA_m and `care` for $n = 6, 12, 18, 24, 30$. We also report the comparison of relative errors in x_{1n} computed by above three algorithms in Table 8.

Table 6
Comparison of normalized residuals for Example 15

		SDA	MSGM	care
$N = 5$	NRes	1.61×10^{-16}	8.75×10^{-15}	2.53×10^{-15}
	Iter. no.	5	6	–
$N = 20$	NRes	3.85×10^{-16}	3.55×10^{-14}	6.15×10^{-15}
	Iter. no.	5	6	–
$N = 60$	NRes	1.53×10^{-15}	2.32×10^{-13}	8.14×10^{-15}
	Iter. no.	7	8	–
$N = 100$	NRes	2.15×10^{-15}	6.62×10^{-13}	2.55×10^{-14}
	Iter. no.	8	9	–
$N = 140$	NRes	3.05×10^{-15}	6.50×10^{-12}	3.60×10^{-14}
	Iter. no.	8	9	–
$N = 180$	NRes	1.25×10^{-14}	4.64×10^{-12}	2.01×10^{-13}
	Iter. no.	9	9	–

5.1. Comments on numerical results

We have tested 20 examples in [23] to illustrate the accuracy and efficiency of the SDA applied to CAREs, in comparison to the MSGM [21] and `care` in MATLAB [40]. Some of these examples have parameters to vary their sizes or conditioning. In what follows, we shall comment upon all the examples in [23], thus retaining the old labelling of the examples:

(1) Comparing with `care` for all the examples, solutions with better or comparable accuracy were obtained using the SDA in far less time. This comparison has been difficult as `care` yields no iteration numbers and the CPU time information from MATLAB is not always accurate.
(2) The best indication of the efficiency of the SDA over `care` comes from Example 15 (with varying dimension n), where `care` required two to eight times more CPU times than the SDA. This is consistent with the findings in [24] for DAREs. Keep in mind that the SDA requires far less number of flops than `care` in each iteration, as the operations in the SDA are performed in $\mathbb{R}^{n\times n}$ whereas those for `care` are carried out in $\mathbb{R}^{2n\times 2n}$.
(3) For examples with varying conditioning, such as Examples 9–14, 17 and 18, the SDA out-performed `care` and converges to more accurate solutions in less time. For the ill-conditioned Example 20, `care` failed while the SDA succeeded without difficulty.
(4) In Example 11 (in H_∞ control), some eigenvalues were numerically on the imaginary axis and assumptions in the theory were practically violated. The stronger structure-preserving property of the SDA enabled it to produce an

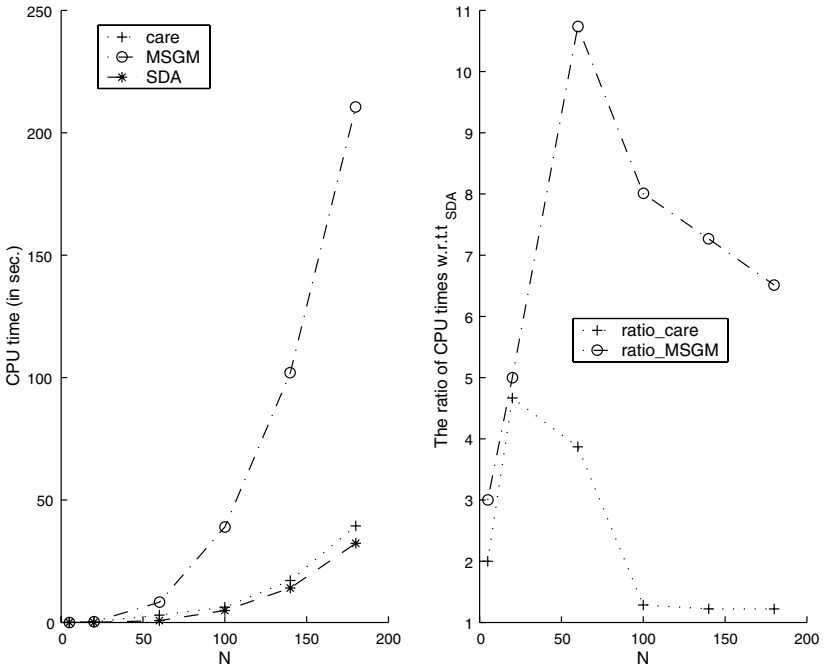

Fig. 3. Comparison of CPU times for Example 15.

accurate solution when the MSGM failed. Somehow, `care` produced a slightly less accurate solution using much more CPU time.

(5) In Examples 10 and 17, the CAREs gave rise to DAREs which were "doubly symmetric" (see Section 4 for details). The SDA_m improved the efficiency of the SDA for these examples, obtaining comparable accuracy for Example 10 while out-performing `care` for Example 17.

(6) Comparing to the MSGM for ill-conditioned problems, the SDA performed better in terms of accuracies or number of iterations. This is consistent with the fact that while both the SDA and MSGM are structure-preserving, the former preserves more structure than the latter. For some well-conditioned problems, the efficiency and accuracy of the SDA and MSGM are comparable. For a few simple small examples, the MSGM converged quickly and was superior to the SDA. Note that the work involved in an iteration for either method is similar.

(7) The MSGM, with similar operations count to SDA, was generally more efficient than `care`, especially for well-conditioned problems. For ill-conditioned problems (such as Example 10), the MSGM was sometimes less accurate than `care`.

Table 7
Comparison of normalized residuals for Example 17

	n	NRes_SDA	NRes_SDA_m	NRes_MSGM	NRes_care
$q, r = 1$	6	4.50×10^{-15}	3.56×10^{-16}	8.87×10^{-15}	1.80×10^{-14}
	12	3.63×10^{-10}	3.22×10^{-14}	9.68×10^{-12}	1.23×10^{-11}
	18	9.47×10^{-5}	1.83×10^{-11}	4.63×10^{-9}	9.46×10^{-9}
	24	2.47×10^{-2}	2.34×10^{-8}	9.88×10^{-6}	3.25×10^{-7}
	30	4.80×10^{-1}	3.52×10^{-5}	3.47×10^{-2}	7.17×10^{-4}
$q, r = 100$	6	2.59×10^{-15}	2.82×10^{-16}	1.20×10^{-11}	1.02×10^{-15}
	12	4.81×10^{-10}	2.94×10^{-14}	4.11×10^{-9}	1.58×10^{-11}
	18	4.33×10^{-5}	2.26×10^{-11}	1.78×10^{-6}	7.83×10^{-9}
	24	7.24×10^{-1}	2.90×10^{-8}	1.37×10^{-2}	1.50×10^{-5}
	30	3.07×10^{-1}	1.45×10^{-5}	2.94×10^{-1}	4.38×10^{-3}

Table 8
Comparison of relative errors in x_{1n} for Example 17

	n	RE_SDA	RE_SDA_m	RE_MSGM	RE_care
$q, r = 1$	6	1.94×10^{-14}	1.11×10^{-15}	2.22×10^{-14}	5.68×10^{-14}
	12	3.99×10^{-10}	1.68×10^{-13}	4.92×10^{-11}	5.61×10^{-11}
	18	8.73×10^{-5}	6.37×10^{-11}	2.35×10^{-8}	2.68×10^{-8}
	24	3.09×10^{-1}	6.39×10^{-8}	3.29×10^{-5}	6.16×10^{-6}
	30	6.49×10^{-1}	1.57×10^{-4}	1.40×10^{-1}	8.37×10^{-3}
$q, r = 100$	6	2.13×10^{-15}	9.95×10^{-16}	3.27×10^{-11}	7.67×10^{-15}
	12	6.04×10^{-10}	1.83×10^{-13}	2.30×10^{-8}	6.13×10^{-11}
	18	5.87×10^{-4}	1.16×10^{-10}	2.95×10^{-5}	2.70×10^{-8}
	24	2.02×10^{-1}	1.32×10^{-7}	2.39×10^{-2}	5.20×10^{-5}
	30	4.60×10^{-1}	5.67×10^{-5}	2.98×10^{-1}	1.71×10^{-2}

6. Conclusions

Solving CAREs as DAREs, after applying the Cayley transform, has previously been investigated by many. Recent developments and better understanding of doubling algorithms, especially the structure-preserving properties and efficiency of the SDA [24], give this old approach a new lease of life. In addition, we have studied how the parameter γ in the Cayley transform can be chosen optimally. A Fibonacci search for choosing γ was suggested in Section 3, together with the details of other issues involved in the practical implementation of the SDA. We have also developed the SDA_m which preserves the structure of some doubly symmetric DAREs. Extensive numerical results show that this approach of solving CAREs using the SDA is efficient and competitive, especially for ill-conditioned problems.

Appendix A

Lemma A.1. *If* $\Phi^{\mathrm{T}} = \Phi \geqslant 0$ *and* $\Psi^{\mathrm{T}} = \Psi \geqslant 0$, *then* $\Phi(I + \Phi\Psi)^{-1} \geqslant 0$ *and* $(I + \Psi\Phi)^{-1} \geqslant 0$.

Proof

It suffices to prove that $\Phi(I + \Psi\Phi)^{-1} \geqslant 0$. Notice that if Φ is positive definite, then the matrix $\Phi(I + \Psi\Phi)^{-1} = (\Phi^{-1} + \Psi)^{-1}$ is also positive definite. Now, since the matrix $\Psi^{\mathrm{T}} = \Phi \geqslant 0$, we know that $\Phi + \epsilon I$ is positive definite for $\epsilon \geqslant 0$, and hence we have

$$(\Phi + \epsilon I)[I + \Psi(\Phi + \epsilon I)]^{-1} > 0. \tag{A.1}$$

Let $\epsilon \to 0$, we obtain the desired result. Similarly, it can be shown that $(I + \Psi\Phi)^{-1}\Psi \geqslant 0$.

Acknowledgment

We would like to thank Professor Ralph Byers and the referee for their valuable comments and suggestions on the manuscript.

References

[1] G. Ammar, V. Mehrmann, On Hamiltonian and symplectic Hessenberg forms, Linear Algebra Appl. 149 (1991) 55–72.
[2] B.D.O. Anderson, Second-order convergent algorithms for the steady-state Riccati equation, Internat. J. Control 28 (1978) 295–306.
[3] M. Athans, W. Levine, A. Levis, A system for the optimal and suboptimal position and velocity control for a string of high-speed vehicles, in: Proc. Fifth Int. Analogue Computation Meetings, Lausanne, Switzerland, 1967.
[4] Z. Bai, J. Demmel, On swapping diagonal blocks in real Schur form, Linear Algebra Appl. 186 (1993) 73–95.
[5] Z. Bai, J. Demmel, Using the matrix sign function to compute invariant subspaces, SIAM J. Matrix Anal. Appl. 19 (1998) 205–225.
[6] Z. Bai, Q. Qian, Inverse free parallel method for the numerical solution of algebraic Riccati equations, in: J.G. Lewis (Ed.), Proc. Fifth SIAM Conf. Appl. Linear Algebra, Snowbird, UT, June 1994, SIAM, Philadelphia, PA, 1994, pp. 167–171.
[7] L. Balzer, Accelerated convergence of the matrix sign function, Internat. J. Control 21 (1980) 1057–1078.
[8] A.Y. Barraud, Investigation autour de la function signe d'une matrice, application á l' équation de Riccati, R.A.I.R.O. Automatique 13 (1979) 335–368.
[9] A.Y. Barraud, Produit étoile et function signe de matrice. application á l' équation de Riccati dans le cas discrete, R.A.I.R.O. Automatique, 14 (1980) 55–85.
[10] M.S. Bazaraa, H.D. Sheraii, C.M. Shetty, Nonlinear Programming, John Wiley and Sons, 1993.

[11] A.N. Beavers, E.D. Denman, Asymptotic solutions to the matrix Riccati equation, Math. Biosci. 20 (1974) 339–344.
[12] A.N. Beavers, E.D. Denman, A computational method for eigenvalues and eigenvectors of a matrix with real eigenvalues, Numer. Math. 21 (1974) 389–396.
[13] A.N. Beavers, E.D. Denman, A new similarity transformation method for eigenvalues and eigen vectors, Math. Biosci. 21 (1974) 143–169.
[14] A.N. Beavers, E.D. Denman, A new solution method for matrix quadratic equations, Math. Biosci. 20 (1974) 135–143.
[15] P. Benner, A.J. Laub, V. Mehrmann, A collection of benchmark examples for the numerical solution of algebraic Riccati equations I: Continuous-time case, Tech. Rep. SPC 95-22, Fakultät für Mathematik, TU Chemnitz-Zwickau, 09107 Chemnitz, FRG, 1995. Available from: http://www.tu-chemnitz.de/sfb393/spc95pr.html.
[16] J.H. Brandts, Matlab code for sorting real Schur forms, Numer. Linear Algebra Appl. 9 (2002) 249–261.
[17] A. Bunse-Gerstner, R. Byers, V. Mehrmann, A chart of numerical methods for structured eigenvalue problems, SIAM J. Matrix Anal. Appl. 13 (1992) 419–453.
[18] A. Bunse-Gerstner, V. Mehrmann, D. Watkins, An SR algorithm for Hamiltonian matrices, based on Gaussian elimination, Methods Oper. Res. 58 (1989) 15–26.
[19] R. Byers, A Hamiltonian QR-algorithm, SIAM J. Sci. Statist. Comput. 7 (1986) 212–229.
[20] R. Byers, Numerical stability and instability in matrix sign function based algorithms, in: C. Byrnes, A. Lindquist (Eds.), Computational and Combinatorial Methods in System Theory, North-Holland, 1986, pp. 185–200.
[21] R. Byers, Solving the algebraic Riccati equation with the matrix sign function, Linear Algebra Appl. 85 (1987) 267–279.
[22] R. Byers, C. He, V. Mehrmann, The matrix sign function method and the computation of invariant subspaces, SIAM J. Matrix Anal. Appl. 18 (1997) 615–632.
[23] E.K.-W. Chu, H.-Y. Fan, W.-W. Lin, A structure-preserving doubling algorithm for continuous-time algebraic Riccati equations, preprint 2002-28, NCTS, National Tsing Hua University, Hsinchu 300, Taiwan, 2003.
[24] E.K.-W. Chu, H.-Y. Fan, W.-W. Lin, C.-S. Wang, Structure-preserving algorithms for periodic discrete-time algebraic Riccati equations. Internat. J. Control 77(8) (2004) 767–788.
[25] E.K.-W. Chu, H.-Y. Fan, W.-W. Lin, C.-S. Wang, A structure-preserving doubling algorithm for periodic discrete-time algebraic Riccati equations, preprint 2002-18, NCTS, National Tsing Hua University, Hsinchu 300, Taiwan, 2003.
[26] E.J. Davison, W. Gesing, The systematic design of control systems for the multivariable servomechenism problem, in: M.K. Sain, J.L. Peczkowsky (Eds.), Alternatives for Linear Multivariable Control, Nat. Eng. Consortium Inc., Chicago, IL, 1978.
[27] E. Denman, R. Beavers, The matrix sign function and computations in systems, Appl. Math. Comput. 2 (1976) 63–94.
[28] L. Dieci, Some numerical considerations and Newton's method revisited for solving algebraic Riccati equations, IEEE Trans. Automat. Control 36 (1991) 608–616.
[29] J. Gardiner, A.J. Laub, A generalization of the matrix-sign-function solution to the algebraic Riccati equations, Internat. J. Control 44 (1986) 823–832.
[30] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., The Johns Hopkins University Press, 1996.
[31] S. Hammarling, Newton's method for solving the algebraic Riccati equation, NPL Rep. DITC 12/82, Nat. Phys. Lab., Teddington, Middlesex TW11 OLW, UK, 1982.
[32] N.J. Higham, Accuracy and Stability of Numerical Algorithms, Society for Industrial and Applied Mathematics, 1996.
[33] J.L. Howland, The sign matrix and the separation of matrix eigenvalues, Linear Algebra Appl. 49 (1983) 221–332.

[34] P.Hr. Petkov, N.D. Christov, M.M. Konstantinov, On the numerical properties of the Schur approach for solving the matrix Riccati equation, Systems Control Lett. 9 (1987) 197–201.
[35] G.D. Ianculescu, J. Ly, A.J. Laub, P.M. Papadopoulos, Space station freedom solar array H_∞ control. Talk at 31st IEEE Conf. on Decision and Control, Tucson, AZ, December, 1992.
[36] M. Kimura, Convergence of the doubling algorithm for the discrete-time algebraic Riccati equation, Internat. J. Systems Sci. 19 (1988) 701–711.
[37] M. Kimura, Doubling algorithm for continuous-time algebraic Riccati equation, Internat. J. Systems Sci. 20 (1989) 191–202.
[38] D. Kleinman, On an iterative technique for Riccati equation computations, IEEE Trans. Automat. Control AC-13 (1968) 114–115.
[39] A.J. Laub, A Schur method for solving algebraic Riccati equations, IEEE Trans. Automat. Control 24 (1979) 913–921.
[40] MathWorks, MATLAB user's guide (for UNIX Workstations), The Math Works, Inc., 1992.
[41] V. Mehrmann, The Autonomous Linear Quadratic Control Problem, Springer-Verlag, 1991.
[42] V. Mehrmann, A step toward a unified treatment of continuous and discrete time control problems, Linear Algebra Appl. 241–243 (1996) 749–779.
[43] V. Mehrmann, E. Tan, Defect correction methods for the solution of algebraic Riccati equations, IEEE Trans. Automat. Control AC-33 (1988) 695–698.
[44] C.C. Paige, C.F. Van Loan, A Schur decomposition for Hamiltonian matrices, Linear Algebra Appl. 41 (1981) 11–32.
[45] L. Patnaik, N. Viswanadham, I. Sarma, Computer control algorithms for a tubular ammonia reactor, IEEE Trans. Automat. Control 25 (1980) 642–651.
[46] J. Roberts, Linear model reduction and solution of the algebraic Riccati equation by the use of the sign function, Internat. J. Control 32 (1980) 667–687.
[47] N. Sandell, On Newton's method for Riccati equation solution, IEEE Trans. Automat. Control AC-19 (1974) 254–255.
[48] G.W. Stewart, HQR3 and EXCHNG: Fortran subroutines for calculating and ordering the eigenvalues of a real upper Hessenberg matrix, ACM Trans. Math. Software 2 (1976) 275–280.
[49] K. Zhou, J.C. Doyle, K. Glover, Robust and Optimal Control, Prentice-Hall, Upper Saddle River, NJ, 1996.