SLICOT Working Note 1999-16

DARЕX — A Collection of Benchmark Examples for Discrete-Time Algebraic Riccati Equations (Version 2.0) [1]

Jörn Abels[2] Peter Benner[3]

December 1999

[1]This document presents research results of the European Community BRITE-EURAM III Thematic Networks Programme NICONET (contract number BRRT-CT97-5040) and is distributed by the Working Group on Software WGS. *WGS secretariat*: Mrs. Ida Tassens, ESAT - Katholieke Universiteit Leuven, K. Mercierlaan 94, 3001-Leuven-Heverlee, BELGIUM. This report is also available by anonymous ftp from `wgs.esat.kuleuven.ac.be` in the directory /pub/WGS/REPORTS/ with file name SLWN1999-16.ps.Z

[2]Fachbereich 3 – Mathematik und Informatik, Universität Bremen, D–28334 Bremen, Germany; `abels@informatik.uni-bremen.de`

[3]Zentrum für Technomathematik, Fachbereich 3 – Mathematik und Informatik, Universität Bremen, D–28334 Bremen, Germany; `benner@math.uni-bremen.de`

Abstract

This is the second part of a collection of benchmark examples for the numerical solution of algebraic Riccati equations. After presenting examples for the continuous-time case in Part I (CAREX), our concern in this paper is discrete-time algebraic Riccati equations. This collection may serve for testing purposes in the construction of new numerical methods, but may also be used as a reference set for the comparison of methods. This version updates an earlier benchmark collection. Some of the examples have been extended by incorporating parameters and there have been some new additions to the collection.

0 Introduction

We present a collection of examples for discrete-time algebraic Riccati equations (DARE) of the form

$$0 = A^T XA - X - (A^T XB + S)(R + B^T XB)^{-1}(B^T XA + S^T) + Q \tag{1}$$

where $A, Q, X \in \mathbb{R}^{n\times n}$, $B, S \in \mathbb{R}^{n\times m}$, and $R = R^T \in \mathbb{R}^{m\times m}$. The matrix $Q = Q^T$ may be given in factored form $Q = C^T\tilde{Q}C$ with $C \in \mathbb{R}^{p\times n}$ and $\tilde{Q} = \tilde{Q}^T \in \mathbb{R}^{p\times p}$.

As described below, (1) can be solved using its relationship to the symplectic pencil defined by

$$L - \lambda M = \begin{bmatrix} \hat{A} & 0 \\ -\hat{Q} & I_n \end{bmatrix} - \lambda \begin{bmatrix} I_n & G \\ 0 & \hat{A}^T \end{bmatrix} \tag{2}$$

where

$$\hat{A} = A - BR^{-1}S^T, \qquad G = BR^{-1}B^T, \qquad \hat{Q} = Q - SR^{-1}S^T = C^T\tilde{Q}C - SR^{-1}S^T.$$

If $\hat{A}$ is invertible, this pencil is equivalent to the symplectic matrices

$$Z = M^{-1}L = \begin{bmatrix} \hat{A} + G\hat{A}^{-T}\hat{Q} & -G\hat{A}^{-T} \\ -\hat{A}^{-T}\hat{Q} & \hat{A}^{-T} \end{bmatrix}, \tag{3}$$

$$\text{or} \qquad \tilde{Z} = L^{-1}M = \begin{bmatrix} \hat{A}^{-1} & \hat{A}^{-1}G \\ \hat{Q}\hat{A}^{-1} & \hat{Q}\hat{A}^{-1}G + \hat{A}^T \end{bmatrix}. \tag{4}$$

The DARE (1) arises, e.g., in (a) filtering or stochastic realization problems, and (b) linear-quadratic control problems. In case (a), R is the measurement noise covariance and it is not uncommon for this kind of matrix to be singular. For (b), R is the control weighting matrix and in the discrete-time case, occasionally such a matrix can be singular, too. In these cases, the pencil formulation (2) is not possible. An *extended symplectic pencil* (ESP) can then be formed by

$$\tilde{L} - \lambda\tilde{M} = \begin{bmatrix} A & 0 & B \\ Q & -I_n & S \\ S^T & 0 & R \end{bmatrix} - \lambda \begin{bmatrix} I_n & 0 & 0 \\ 0 & -A^T & 0 \\ 0 & -B^T & 0 \end{bmatrix}. \tag{5}$$

To illustrate a problem where the DARE (1) arises, we consider the discrete-time linear-quadratic control problem (case (b) from above).

Minimize

$$J(x_0, u) = \frac{1}{2}\sum_{k=0}^{\infty} \left(y_k^T\tilde{Q}y_k + 2x_k^T Su_k + u_k^T Ru_k \right) \tag{6}$$

subject to the difference equation

$$x_{k+1} = Ax_k + Bu_k, \quad k = 0, 1, \ldots, \qquad x_0 = \xi, \tag{7}$$

$$y_k = Cx_k, \quad k = 0, 1, \ldots. \tag{8}$$

If, for example, $\tilde{Q} \geq 0$, $R > 0$, (A, B) stabilizable[1], and (A, C) detectable[2], then the solution of the optimal control problem (6)–(8) is given by the feedback law

$$u_k = -(R + B^T X B)^{-1}(A^T X B + S)^T x_k, \quad k = 0, 1, \ldots,$$

where X is the unique stabilizing positive semidefinite solution of (1) (see, e.g., [25, 33]).

One common approach to solve (1) is to compute the stable invariant subspace of the symplectic matrix Z or the stable deflating subspace of the (extended) symplectic pencil given above, i.e., the invariant/deflating subspace corresponding to the generalized eigenvalues of $L - \lambda M$, $\tilde{L} - \lambda \tilde{M}$, respectively, inside the unit circle (e.g., [22, 25, 29, 38]). If this subspace is spanned by

$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \begin{array}{l} \}n \\ \}n \end{array} \qquad \text{or} \qquad \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} \begin{array}{l} \}n \\ \}n \\ \}m \end{array}, \qquad \text{respectively,}$$

and U_1 is invertible, then $X = U_2 U_1^{-1}$ is the stabilizing solution of (1), i.e., all the eigenvalues of

$$F = A - B(R + B^T X B)^{-1}(A^T X B + S)^T \tag{9}$$

lie inside the unit circle.

At this point it should be noted that it is possible to transform a continuous-time algebraic Riccati equation (CARE) into a DARE (and vice versa) via a (generalized) Cayley transformation, i.e., the Hamiltonian matrix corresponding to the CARE is transformed into a symplectic matrix/pencil. From this symplectic pencil it is possible to derive the coefficient matrices of a corresponding DARE under certain regularity assumptions; see [26]. In this way, it is possible to obtain DARE examples from the first part of our benchmark collection. We do not use this approach here, though, and restrict ourselves to data arising naturally in a discrete-time setting and/or highlighting some of the properties of discrete-time algebraic Riccati equations.

In the sequel we will use the following notation. Let $A \in \mathbb{R}^{n \times n}$. By $\sigma(A)$ we denote the set of eigenvalues or spectrum of A. The spectral norm of a matrix is given by

$$\|A\| = \sqrt{\max\{|\lambda| \ : \ \lambda \in \sigma(A^T A)\}}$$

and the given matrix condition numbers are based upon the spectral norm,

$$\kappa(A) = \|A\| \|A^{-1}\|.$$

The *Frobenius* norm of a matrix will be denoted by $\|A\|_F$ and is given by

$$\|A\|_F = \sqrt{\sum_{i,j=1}^{n} a_{ij}^2}.$$

All norms and condition numbers given in the sequel were computed by the MATLAB[3] functions `norm` and `cond`.

[1] (A, B) is *(d-)stabilizable*, if rank $[A - \lambda I, \ B] = n$ for all λ with $|\lambda| \geq 1$.
[2] (A,C) is *(d-)detectable*, if (A^T, C^T) is (d-)stabilizable.
[3] MATLAB is a trademark of The MathWorks, Inc.

The examples are grouped in three sections. The first section contains parameter-free examples of fixed dimension while the second has parameter-dependent problems of fixed size. Section 4 contains examples of scalable size.

The coefficient matrices of the presented examples are usually given in the same form as they appear in the literature. Since in most cases $S = 0$, we omit S in all examples where this property holds.

All presented examples may be generated by the FORTRAN 77 subroutine DDAREX (see Appendix A).

The description of each example contains a table with some of the system properties. This information is summarized in Appendix B. For all parameters needed in the examples there exist default values that are also given in the tables. These default values are chosen in such a way that the collection of examples can be used as a test set for the comparison of methods. The tables contain information for one or more choices of the parameters. Underlined values indicate the default values.

For each example, we provide the condition number $\kappa(\hat{A})$ which shows if the symplectic matrix Z in (3) can be formed safely (though it is still preferable to use the pencil approach and thereby avoid a matrix inversion unless $\hat{A}^{-1}$ is "easy" to form). The column $|\lambda^C_{max}|$ indicates the absolute value of the closed-loop eigenvalue of largest modulus, i.e.,

$$|\lambda^C_{max}| = \max\{\, |\lambda| \; : \; \lambda \in \sigma(F)\, \}$$

with F as in (9). These are the (generalized) eigenvalues of the symplectic matrix (pencils) in (2), (3), and (5) inside the unit circle. Further, we give norms and condition numbers of the stabilizing solution X. For examples without an analytical solution available, we computed approximations by the generalized Schur vector method [3, 29]. If possible, these approximations were refined by Newton's method [3, 15, 25] to achieve the highest possible accuracy. We then chose the approximate solution with smallest residual norm and recomputed the solution using the optimal scaling strategy proposed in [14]. This computed solution was then used to determine the properties of the example.

The condition number of discrete-time algebraic Riccati equations has been tackled by several papers during recent years, see, e.g., [14, 19, 31, 36, 37]. For simplicity, here we use the condition number proposed in [14]. This condition number measures the sensitivity of the stabilizing DARE solution with respect to first-order perturbations by means of the Fréchet derivative of the DARE at X. In [14] it is shown that, assuming $Q \geq 0$, $R > 0$, such a condition number is given by

$$K_{DARE} = \frac{\|\,[\,Z_1,\, Z_2,\, Z_3\,]\,\|}{\|X\|_F},$$

where

$$\begin{aligned} Z_1 &= \|A\|_F P^{-1}\left(I_n \otimes F^T X + (F^T X \otimes I_n)T\right), && (10)\\ Z_2 &= -\|G\|_F P^{-1}\left(\hat{A}^T X (I_n + GX)^{-1} \otimes \hat{A}^T X (I_n + GX)^{-1}\right), && (11)\\ Z_3 &= \|Q\|_F P^{-1}. && (12) \end{aligned}$$

Here, denoting the jth unit vector by e_j, the permutation matrix T is defined by

$$T = \sum_{i,j=1}^{n} e_i e_j^T \otimes e_j e_i^T,$$

and P is the matrix representation of the *Stein* (*discrete Lyapunov*) operator

$$\Omega(Z) = Z - F^T Z F.$$

The computation of K_{DARE} therefore requires the solution of the linear equations (10)–(12). Since Kronecker products are involved, these systems get very large even for small values of n. For larger n, an inverse power iteration can be employed to estimate $\| [\, Z_1,\, Z_2,\, Z_3 \,] \|_F$ (see [14]). This approach requires in each iteration step the solution of two Stein equations corresponding to Ω.

The given benchmark collection is based on the collection described in [5, 7]. The FORTRAN 77 subroutine has been modified, a couple of bugs have been corrected, and in particular some new examples have been added. In the following sections, in case an example was already contained in [5, 7], then the first reference for the example gives the number of the example in [5].

1 Parameter-free problems of fixed size

Examples 1 and 2 of [5] have been extended and now possess a parameter. Therefore, the first example in this section is Example 3 of [5].

Example 1.1 [5, Example 3], [38, Example II]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\|X\|$	$\kappa(X)$	K_{DARE}
2	1	1		5.8	0.0	1.0	1.0	–

This example was used in [38] to demonstrate a compression technique for the extended pencil (5). The data are given by

$$A = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad Q = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R = 0.$$

If interpreted in terms of a linear system as in (7)–(8), Q can be written as

$$Q = C^T \tilde{Q} C, \qquad C = [\,0 \;\; 1\,], \qquad \tilde{Q} = 1.$$

The exact stabilizing solution is $X = I_2$, and the closed-loop spectrum is $\{\,0,\, 0\,\}$. Due to the singularity of R, the condition number K_{DARE} is not defined here (represented by "–" in the table).

This example can be used, e.g., as a first test of any solver to deal with a singular weighting/measurement noise covariance matrix.

Example 1.2 [5, Example 4], [16]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
2	2	2		∞	0.69	1.3×10^2	2.8×10^3	–

This is another example with a singular R–matrix. Furthermore, we have a nonzero S–matrix. The coefficients of (1) are given by

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}, \qquad R = \begin{bmatrix} 9 & 3 \\ 3 & 1 \end{bmatrix}, \\ Q &= \frac{1}{11}\begin{bmatrix} -4 & -4 \\ -4 & 7 \end{bmatrix}, \qquad S = \begin{bmatrix} 3 & 1 \\ -1 & 7 \end{bmatrix}. \end{aligned}$$

Again, the DARE condition number K_{DARE} can not be computed due to the singular R.

Example 1.3 [5, Example 5], [17], [26, Example 2].

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
2	1	2		∞	0.38	5.2	1.1×10^2	1.9

This example shows one of the major differences between the properties of continuous-time algebraic Riccati equations and their discrete counterparts. Consider the DARE defined by

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad Q = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}, \qquad R = 1.$$

The spectrum of the pencil $L - \lambda M$ in (2) is $\{\,0,\ \infty,\ -(3\pm\sqrt{5})/2\,\}$. The DARE has exactly two solutions,

$$X_1 = \begin{bmatrix} 1 & 2 \\ 2 & 2+\sqrt{5} \end{bmatrix}, \qquad X_2 = \begin{bmatrix} 1 & 2 \\ 2 & 2-\sqrt{5} \end{bmatrix}.$$

but neither of them is negative semidefinite. On the other hand, $(A,\,B)$ is controllable. In the case of a continuous-time system, this property would assure the existence of a negative semidefinite solution.

The stabilizing solution in the control-theoretic sense is the positive definite solution X_1.

Example 1.4 [37]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
3	2	3		∞	0.0	1.0×10^5	∞	–

The data matrices are as follows.

$$\begin{aligned} A &= \begin{bmatrix} 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \\ 0 & 0 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \qquad C = I_3, \\ Q &= \begin{bmatrix} 10^5 & 0 & 0 \\ 0 & 10^3 & 0 \\ 0 & 0 & -10 \end{bmatrix}, \qquad R = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \end{aligned}$$

In this example, A is stable. The state-weighting matrix Q is indefinite and the control-weighting matrix R is singular. The corresponding linear-time invariant system is controllable and observable.

The stabilizing solution of the DARE is given by the positive semidefinite matrix

$$X = \begin{bmatrix} 10^5 & 0 & 0 \\ 0 & 10^3 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and the closed-loop spectrum consists of just the eigenvalue zero with algebraic multiplicity 3. The condition number of the DARE is not computed due to the singular R matrix.

Example 1.5 [5, Example 6], [1]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
4	2	4		1.0	0.94	35.4	3.3	30.6

The data of this example represent a simple control problem for a satellite. The system is given by equations describing the small angle altitude variations about the roll and yaw axes of a satellite in circular orbit. These equations originally form a second-order differential equation. A first-order realization of this model and sampling every 0.1 seconds yields the system matrices

$$A = \begin{bmatrix} 0.998 & 0.067 & 0 & 0 \\ -0.067 & 0.998 & 0 & 0 \\ 0 & 0 & 0.998 & 0.153 \\ 0 & 0 & -0.153 & 0.998 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0033 & 0.02 \\ 0.1 & -0.0007 \\ 0.04 & 0.0073 \\ -0.0028 & 0.1 \end{bmatrix}.$$

The weighting matrices used in the performance index $J(x_0, u)$ in (6) are given by

$$Q = \tilde{Q} = \begin{bmatrix} 1.87 & 0 & 0 & -0.244 \\ 0 & 0.744 & 0.205 & 0 \\ 0 & 0.205 & 0.589 & 0 \\ -0.244 & 0 & 0 & 1.048 \end{bmatrix}, \qquad R = I_2.$$

Example 1.6 [5, Example 7], [23]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
4	2	4		19.9	0.99	2.1	1.8×10^2	7.9×10^2

This is a simple example of a control system having slow and fast modes.

$$A = 10^{-3} \times \begin{bmatrix} 984.75 & -79.903 & 0.9054 & -1.0765 \\ 41.588 & 998.99 & -35.855 & 12.684 \\ -546.62 & 44.916 & -329.91 & 193.18 \\ 2662.4 & -100.45 & -924.55 & -263.25 \end{bmatrix},$$

$$B = 10^{-4} \times \begin{bmatrix} 37.112 & 7.361 \\ -870.51 & 0.093411 \\ -11984.0 & -4.1378 \\ -31927.0 & 9.2535 \end{bmatrix}, \qquad R = I_2, \qquad Q = 0.01 I_4.$$

One complex conjugate pair of the computed closed-loop eigenvalues is located on a circle with radius ≈ 0.99 around the origin, i.e., relatively close to the unit circle. Requiring that this distance should not cause any problems for any DARE solver seems to be reasonable.

Example 1.7 [5, Example 8], [24, Example 4.3]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
4	4	4		3.8×10^2	$\approx 1-1.8\times 10^{-5}$	65.8	6.2×10^{11}	5.1×10^4

Here, the coefficient matrices of (1) are constructed as follows. Given

$$A_0 = \begin{bmatrix} 0.4 & 0 & 0 & 0 \\ 1 & 0.6 & 0 & 0 \\ 0 & 1 & 0.8 & 0 \\ 0 & 0 & 0 & -0.999982 \end{bmatrix}, \qquad Q_0 = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$V = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad\Longleftrightarrow\quad V^{-1} = \begin{bmatrix} 1 & 1 & 2 & 4 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

we obtain

$$A = VA_0V^{-1} = \begin{bmatrix} -0.6 & -2.2 & -3.6 & -5.400018 \\ 1 & 0.6 & 0.8 & 3.399982 \\ 0 & 1 & 1.8 & 3.799982 \\ 0 & 0 & 0 & -0.999982 \end{bmatrix}, \qquad B = VI_4 = V,$$

$$Q = V^{-T}Q_0V^{-1} = \begin{bmatrix} 2 & 1 & 3 & 6 \\ 1 & 2 & 2 & 5 \\ 3 & 2 & 6 & 11 \\ 6 & 5 & 11 & 22 \end{bmatrix}, \qquad R = I_4.$$

A factorization in the control-theoretic sense, $Q = C^T\tilde{Q}C$, is therefore given by $C := V^{-1}$ and $\tilde{Q} := Q_0$. All the generalized eigenvalues of $L - \lambda M$ are real. The distance of the largest closed-loop eigenvalue to the unit circle is $\approx 1.8\times 10^{-5}$. The problem is designed so that $\kappa(L+M) \approx 4\times 10^{11}$. Due to this large condition number and the eigenvalues close to the unit circle, problems with the convergence of the iteration process are to be expected when the DARE is solved via a method based on the sign function iteration (e.g., the methods in [4, 12]). Note that K_{DARE} signals only a very mild ill-conditioning of the DARE.

Example 1.8 [5, Example 9], [11, Section 2.7.4]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
5	2	5		23.5	0.98	73.9	73.7	1.0×10^2

The fifth-order linearized state-space model of a chemical plant presented in [13, 34] is discretized by sampling every 0.5 seconds, yielding a discrete-time linear-quadratic control problem of the form (6)–(7) defined by

$$A = 10^{-4}\times \begin{bmatrix} 9540.70 & 196.43 & 35.97 & 6.73 & 1.90 \\ 4084.90 & 4131.70 & 1608.40 & 446.79 & 119.71 \\ 1221.70 & 2632.60 & 3614.90 & 1593.00 & 1238.30 \\ 411.18 & 1285.80 & 2720.90 & 2144.20 & 4097.60 \\ 13.05 & 58.08 & 187.50 & 361.62 & 9428.00 \end{bmatrix}, \quad B = 10^{-4}\times \begin{bmatrix} 4.34 & -1.22 \\ 266.06 & -104.53 \\ 375.30 & -551.00 \\ 360.76 & -660.00 \\ 46.17 & -91.48 \end{bmatrix}.$$

The weighting matrices in the cost functional (6) are chosen as identities, i.e., we have $Q = \tilde{Q} = I_5$ and $R = I_2$.

If we modify the optimal control problem (6)–(8) by allowing the output to depend explicitly upon the control, we obtain the following problem:

Minimize

$$J(x_0,u) = \frac{1}{2}\sum_{k=0}^{\infty}\left(y_k^T\tilde{Q}y_k + u_k^T\tilde{R}u_k\right) \tag{13}$$

subject to the difference equation

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \quad k=0,1,\ldots, \qquad x_0=\xi, && (14)\\ y_k &= Cx_k + Du_k, \quad k=0,1,\ldots. && (15)\end{aligned}$$

Then we can rewrite the cost functional (13) as

$$J(x_0,u) = \frac{1}{2}\sum_{k=0}^{\infty}\left(x^TQx + x^TSu + u^TS^Tx + u^TRu\right) \tag{16}$$

where $Q = C^T\tilde{Q}C$, $R = \tilde{R} + D^T\tilde{Q}D$, and $S = C^T\tilde{Q}D$.

The data of the following example come from such a problem.

Example 1.9 [5, Example 10], [9]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
6	2	2		∞	0.67	2.5	37.4	3.9

The matrices of the linear system (A,B,C,D) are given by

$$A = \begin{bmatrix} 0&1&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&0&1\\ 0&0&0&0&0&0 \end{bmatrix}, \quad B = \begin{bmatrix} 0&0\\0&0\\1&0\\0&0\\0&0\\0&1 \end{bmatrix},$$

$$C = \begin{bmatrix} 1&1&0&0&0&0\\ 0&0&0&1&-1&0 \end{bmatrix}, \quad D = \begin{bmatrix} 1&0\\1&0 \end{bmatrix}.$$

With $\tilde{Q} = \tilde{R} = I_2$, we obtain the following coefficient matrices for the DARE: A, B, C, $\tilde{Q}$ are defined above, and

$$Q = C^T\tilde{Q}C = \begin{bmatrix} 1&1&0&0&0&0\\ 1&1&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&1&-1&0\\ 0&0&0&-1&1&0\\ 0&0&0&0&0&0 \end{bmatrix}, \quad R = \begin{bmatrix} 3&0\\0&1 \end{bmatrix}, \quad S = \begin{bmatrix} 1&0\\1&0\\0&0\\1&0\\-1&0\\0&0 \end{bmatrix}.$$

The system properties are benign. The system can easily be transformed to a standard system as in (2). Therefore, this example is helpful for first verification of any DARE solver based on the extended formulation given in (5) since the results can be compared to those obtained by any other solver based on the formulation by the symplectic pencil (2).

Example 1.10 [5, Example 11], [30]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
9	3	2		1.6×10^6	0.96	6.1×10^2	∞	74.2

This is the data for a 9th-order discrete state-space model of a tubular ammonia reactor. It should be noted that the underlying model includes a disturbance term which is neglected in this context.

The continuous state-space model of this problem was presented as Example 5 in the first part of the benchmark collection [6]. Sampling every 30 seconds yields the following system matrices for the discrete model:

$$A = 10^{-2} \times \begin{bmatrix} 87.01 & 13.50 & 1.159 & 0.05014 & -3.722 & 0.03484 & 0 & 0.4242 & 0.7249 \\ 7.655 & 89.74 & 1.272 & 0.05504 & -4.016 & 0.03743 & 0 & 0.4530 & 0.7499 \\ -12.72 & 35.75 & 81.70 & 0.1455 & -10.28 & 0.0987 & 0 & 1.185 & 1.872 \\ -36.35 & 63.39 & 7.491 & 79.66 & -27.35 & 0.2653 & 0 & 3.172 & 4.882 \\ -96.00 & 164.59 & -12.89 & -0.5597 & 7.142 & 0.7108 & 0 & 8.452 & 12.59 \\ -66.44 & 11.296 & -8.889 & -0.3854 & 8.447 & 1.36 & 0 & 14.43 & 10.16 \\ -41.02 & 69.30 & -5.471 & -0.2371 & 6.649 & 1.249 & 0.01063 & 9.997 & 6.967 \\ -17.99 & 30.17 & -2.393 & -0.1035 & 6.059 & 2.216 & 0 & 21.39 & 3.554 \\ -34.51 & 58.04 & -4.596 & -0.1989 & 10.56 & 1.986 & 0 & 21.91 & 21.52 \end{bmatrix},$$

$$B^T = 10^{-4} \times \begin{bmatrix} 4.76 & 0.879 & 1.482 & 3.892 & 10.34 & 7.203 & 4.454 & 1.971 & 3.773 \\ -0.5701 & -4.773 & -13.12 & -35.13 & -92.75 & -61.59 & -36.83 & -15.54 & -30.28 \\ -83.68 & -2.73 & 8.876 & 24.80 & 66.80 & 38.34 & 20.29 & 6.937 & 14.69 \end{bmatrix}.$$

In the discrete model, only the first and fifth state variables are used as outputs, i.e.,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and the weighting matrices are chosen as $\tilde{Q} = 50 I_2$ and $R = I_3$.

The last three examples in this section come from proportional-plus-integral (PI) control problems. The design includes the original system as in (6)–(8) with coefficient matrices A_1, B_1, C_1, Q_1, and R_1. Additionally, there are r error integrators that are concatenated with the original system. These error integrators are used to achieve zero steady-state regulation error, i.e., to obtain convergence of the outputs to the given reference values. The error integrators are penalized in the cost functional with Q_2. That is, the structure of the matrices involved in the PI control problem that yield the coefficient matrices of the DARE to be solved is

$$A = \begin{bmatrix} A_1 & 0 \\ -C & I_r \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} C_1 & 0 \\ 0 & I_r \end{bmatrix}, \quad Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}, \quad R = R_1. \tag{17}$$

The PI control problem can then be stated exactly as in (6)–(8) with the above matrices. See, e.g., [35, Section 1.2] for details.

Example 1.11 [35, Section 1.2.2], [10]

n	m	p	parameter	$\kappa(A)$	$\vert\lambda^C_{max}\vert$	$\Vert X\Vert$	$\kappa(X)$	K_{DARE}
11	2	4		∞	0.80	4.4×10^4	9.8×10^5	1.9×10^3

This example comes from PI control of a paper machine with two outputs, two input variables, and two error integrators. The actual data are defined by

$$
\begin{aligned}
A_1 &= \begin{bmatrix} 0 & 0 \\ I_4 & 0 \\ 0 & A_{22} \end{bmatrix}, \qquad A_{22} = \begin{bmatrix} 0.222 & 0.778 & 0 & 0 & 0 \\ 0.4 & 0 & 0.6 & 0 & 0 \\ 0 & 0 & 0 & 1.372 & -0.47 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \\
B_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.098 & 0 \end{bmatrix}^T, \\
C_1 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7 & -5.357 & -3.943 \end{bmatrix}, \\
Q_1 &= Q_2 = \begin{bmatrix} 0.5 & 0 \\ 0 & 5 \end{bmatrix}, \qquad R_1 = \begin{bmatrix} 400 & 0 \\ 0 & 700 \end{bmatrix}.
\end{aligned}
$$

Example 1.12 [35, Example 2.6]

n	m	p	parameter	$\kappa(A)$	$\vert\lambda^C_{max}\vert$	$\Vert X\Vert$	$\kappa(X)$	K_{DARE}
13	2	4		∞	0.81	4.0×10^3	9.0×10^4	1.3×10^2

The data in this example describe PI control of a paper machine model and are given by

$$
\begin{aligned}
A_1 &= \begin{bmatrix} 0 & I_4 & 0 \\ 0 & 0 & A_{22} \end{bmatrix}, \qquad A_{22} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0.7788 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -0.4724 & 1.3746 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0.8071 \end{bmatrix}, \\
B_1 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \\
C_1 &= \begin{bmatrix} 3.318 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.5484 & 0 & 0 & 0 & 0 & 0 & -0.3981 & -0.5113 & -5.7865 & 0 & 0 \end{bmatrix}, \\
Q_1 &= Q_2 = \begin{bmatrix} 0.5 & 0 \\ 0 & 5 \end{bmatrix}, \qquad R_1 = \begin{bmatrix} 400 & 0 \\ 0 & 700 \end{bmatrix}.
\end{aligned}
$$

Again, there are two outputs, two input variables, and two error integrators.

Example 1.13 [35, Example 3.16], [18]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
26	6	12		6.0×10^2	0.97	1.8×10^4	1.3×10^7	5.7×10^6

Here, the data describe a discrete-time power plant model. A PI control with $r = 6$ error integrators is applied. The resulting coefficient matrices of the DARE follow from (17) and, using MATLAB notation,

$$A_1(:,1:10) = \begin{bmatrix}
0 & -0.4607 & 0 & 0.0045 & 0 & 0.1304 & 0 & 0.0731 & 0 & 0.0608 \\
1 & 1.4269 & 0 & 0.0034 & 0 & -0.1702 & 0 & -0.0728 & 0 & -0.0527 \\
0 & 0.0179 & 0 & -0.1242 & 0 & -0.1065 & 0 & -0.0351 & 0 & -0.0603 \\
0 & -0.0090 & 1 & 1.0126 & 0 & 0.2113 & 0 & 0.0411 & 0 & 0.0543 \\
0 & 0.0406 & 0 & -0.2096 & 0 & -0.0777 & 0 & 0.0492 & 0 & -0.0334 \\
0 & -0.0431 & 0 & 0.1458 & 1 & 1.1320 & 0 & -0.0392 & 0 & 0.0343 \\
0 & 0.1253 & 0 & 0.1610 & 0 & 0.0953 & 0 & -0.6278 & 0 & -0.0066 \\
0 & -0.1222 & 0 & -0.2340 & 0 & -0.0159 & 1 & 1.5797 & 0 & 0.0551 \\
0 & 0.0086 & 0 & -0.1020 & 0 & -0.1038 & 0 & 0.0057 & 0 & 0.1812 \\
0 & -0.0094 & 0 & 0.1284 & 0 & 0.0851 & 0 & -0.0079 & 1 & 0.7771 \\
0 & 0.0095 & 0 & -0.0669 & 0 & -0.0342 & 0 & 0.0658 & 0 & -0.0341 \\
0 & 0.0237 & 0 & 0.0813 & 0 & 0.0268 & 0 & -0.0848 & 0 & 0.0164 \\
0 & 0.0189 & 0 & -0.3275 & 0 & -0.1496 & 0 & 0.0689 & 0 & -0.1201 \\
0 & -0.0059 & 0 & 0.2417 & 0 & 0.1944 & 0 & -0.0739 & 0 & 0.3157 \\
0 & -0.0373 & 0 & 0.0986 & 0 & -0.1332 & 0 & 0.0657 & 0 & -0.0546 \\
0 & 0.0610 & 0 & -0.0931 & 0 & 0.1297 & 0 & -0.0714 & 0 & 0.0254 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix},$$

$$A_1(:,11:20) = \begin{bmatrix}
0 & 0.0178 & 0 & 0.0067 & 0 & -0.0090 & 0 & 0 & 0 & 0 \\
0 & -0.0595 & 0 & 0.0011 & 0 & 0.0065 & 0 & 0 & 0 & 0 \\
0 & -0.0939 & 0 & 0.0004 & 0 & 0.0693 & 0 & 0 & 0 & 0 \\
0 & 0.0521 & 0 & 0.0013 & 0 & -0.0728 & 0 & 0 & 0 & 0 \\
0 & -0.0421 & 0 & -0.0082 & 0 & -0.1448 & 0 & 0 & 0 & 0 \\
0 & 0.0290 & 0 & 0.0002 & 0 & 0.1535 & 0 & 0 & 0 & 0 \\
0 & 0.0144 & 0 & 0.0047 & 0 & 0.1116 & 0 & 0 & 0 & 0 \\
0 & -0.0192 & 0 & -0.0004 & 0 & -0.1173 & 0 & 0 & 0 & 0 \\
0 & -0.0301 & 0 & 0.0048 & 0 & -0.0517 & 0 & 0 & 0 & 0 \\
0 & 0.0253 & 0 & 0.0081 & 0 & 0.0529 & 0 & 0 & 0 & 0 \\
0 & 0.2095 & 0 & 0.0107 & 0 & 0.0923 & 0 & 0 & 0 & 0 \\
1 & 0.6173 & 0 & -0.0138 & 0 & -0.0945 & 0 & 0 & 0 & 0 \\
0 & 0.1359 & 0 & 0.3119 & 0 & -0.0085 & 0 & 0 & 0 & 0 \\
0 & 0.1776 & 1 & 0.3995 & 0 & 0.0516 & 0 & 0 & 0 & 0 \\
0 & 0.0440 & 0 & -0.0111 & 0 & -0.0391 & 0 & 0 & 0 & 0 \\
0 & -0.0848 & 0 & 0.0055 & 1 & 1.0233 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.1228 & 0 & 2.0351 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.8965 & 0 & 3.0747 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.0041 & 0 & 0.2600 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0042 & 1 & 0.1704
\end{bmatrix},$$

$$
B_1 = \begin{bmatrix}
-0.0026 & 0.1205 & -0.0022 & -0.0603 & 0.0091 & -0.0362 \\
0.0010 & -0.0096 & 0.0038 & -0.0378 & 0.0004 & 0.0019 \\
-0.0025 & 0.0800 & -0.0014 & -0.0647 & 0.0188 & 0.0001 \\
-0.0011 & 0.0399 & 0.0034 & -0.0165 & 0.0057 & 0.0087 \\
-0.0014 & 0.0547 & -0.0205 & -0.0001 & 0.0070 & -0.0005 \\
-0.0004 & 0.0105 & 0.0056 & -0.0131 & -0.0014 & 0.0023 \\
-0.0086 & 0.0656 & 0.0191 & -0.0307 & -0.0617 & 0.0095 \\
-0.0101 & 0.2088 & -0.0095 & -0.0300 & 0.0038 & 0.0086 \\
-0.0033 & -0.0021 & -0.0294 & -0.0086 & 0.0035 & -0.0007 \\
0.0013 & 0.0050 & 0.0187 & -0.0022 & -0.0013 & 0.0002 \\
0.0081 & 0.0446 & -0.0092 & 0.0964 & 0.0258 & -0.0081 \\
0.0244 & 0.1400 & 0.0039 & -0.5574 & -0.0419 & 0.0152 \\
0.0147 & -0.1752 & -0.0013 & -1.2201 & -0.0057 & -0.0014 \\
0.0103 & 0.2547 & 0.0006 & 1.9353 & -0.0402 & 0.0117 \\
0.0008 & 0.1287 & 0.0077 & -0.1102 & 0.0106 & 0.0075 \\
0.0006 & 0.0210 & 0.0002 & 0.0126 & 0.0066 & 0.0004 \\
0 & 0.0901 & -0.0152 & 0 & -0.2026 & 0 \\
0 & 0.0810 & 0.0026 & 0 & -0.0255 & 0 \\
0 & 0.0544 & 0.0011 & 0 & -0.0037 & 0 \\
0 & -0.0668 & -0.0035 & 0 & 0.0039 & 0
\end{bmatrix},
$$

$$
C_1 = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix},
$$

$$
Q_1 = 0, \qquad Q_2 = \begin{bmatrix}
2 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 3 & 0 & 0 & 0 \\
0 & 0 & 0 & 3 & 0 & 0 \\
0 & 0 & 0 & 0 & 4 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}, \qquad R_1 = \begin{bmatrix}
0.1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.1 & 0 & 0 & 0 & 0 \\
0 & 0 & 10 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.1 & 0 & 0 \\
0 & 0 & 0 & 0 & 10 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}.
$$

The closed-loop system is slightly ill-separated from the unit circle and the DARE is fairly ill-conditioned.

2 Parameter-dependent problems of fixed size

Example 2.1 [5, Example 1], [21, Example 2], [37]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
2	1	2	$\delta = 1$	1.1×10^2	0.50	21.0	∞	18.9
			$\underline{\delta = 10^6}$	1.1×10^2	0.999	1.3×10^4	∞	3.9×10^4

This is an example of stabilizable-detectable, but uncontrollable-unobservable data. We have the following system matrices:

$$
A = \begin{bmatrix} 4 & 3 \\ -\frac{9}{2} & -\frac{7}{2} \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \qquad R = \delta, \qquad Q = \begin{bmatrix} 9 & 6 \\ 6 & 4 \end{bmatrix}.
$$

The stabilizing solution is

$$X = \tfrac{1+\sqrt{1+4\delta}}{2} \left[\begin{array}{cc} 9 & 6 \\ 6 & 4 \end{array} \right].$$

Note that in [5, 7, 21], this example was given without parameter, i.e., $R = 1$ was used as weighting matrix. The parameter of R was introduced in [37] to construct ill-conditioned DAREs. Small values for δ will not affect the condition number much while K_{DARE} grows with increasing values δ.

Example 2.2 [5, Example 2], [21, Example 3], [20, Example 6.15], [37]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
2	2	2	$\delta = 1$	1.1	0.69	5.1×10^{-2}	5.0	4.7
			$\underline{\delta = 10^6}$	1.1	0.91	0.1	17.9	3.7×10^7

Here, the coefficient matrices are

$$\begin{aligned} A &= \left[\begin{array}{cc} 0.9512 & 0 \\ 0 & 0.9048 \end{array} \right], \qquad B = \left[\begin{array}{cc} 4.877 & 4.877 \\ -1.1895 & 3.569 \end{array} \right], \\ R &= \left[\begin{array}{cc} \frac{1}{3\delta} & 0 \\ 0 & 3\delta \end{array} \right], \qquad Q = \left[\begin{array}{cc} 0.005 & 0 \\ 0 & 0.02 \end{array} \right]. \end{aligned}$$

Note that in [5, 7, 20, 21], this example was given without parameter, i.e., $\delta = 1$ was used. The parameter of R was introduced in [37] to construct ill-conditioned DAREs. The condition number K_{DARE} grows with increasing values δ and also if $\delta \to 0$.

Example 2.3 [5, Example 12],

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
2	1	2	$\delta = 100$	∞	0.0	1.0×10^4	1.0×10^4	2.7
			$\underline{\delta = 10^6}$	∞	0.0	1.0×10^{12}	1.0×10^{12}	2.7

Here, the matrix A has a parameter and the coefficient matrices of the DARE (1) are

$$A = \left[\begin{array}{cc} 0 & \delta \\ 0 & 0 \end{array} \right], \qquad B = \left[\begin{array}{c} 0 \\ 1 \end{array} \right], \qquad R = 1, \qquad Q = I_2.$$

The stabilizing solution is given by

$$X = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1+\delta^2 \end{array} \right]$$

and the closed-loop spectrum is $\{\,0, 0\,\}$.

For $\delta = 100$, this is Example 2 from [14]. As $\delta \to \infty$, this becomes an example of a DARE which is badly scaled in the sense of [32] due to the fact that $\lVert A\rVert_F \gg \lVert G\rVert_F$, $\lVert Q\rVert_F$. Obviously, the norm (and condition) of the stabilizing solution X grow like δ^2 whereas the DARE condition number K_{DARE} remains constant.

Example 2.4 [5, Example 13], [32]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
3	3	3	$\delta = 1$	∞	0.38	9.1	9.1	2.5
			$\underline{\delta = 10^6}$	∞	0.38	9.1×10^6	9.1	2.5

This example is constructed as follows. Let

$$A_0 = \mathrm{diag}(0,\ 1,\ 3), \qquad V = I - \frac{2}{3}vv^T, \qquad v^T = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.$$

Then

$$A = VA_0V, \qquad G = \frac{1}{\delta}I_3, \qquad Q = \delta I_3.$$

A factorization $Q = C^T\tilde{Q}C$ can be obtained by setting $C := V$ and $\tilde{Q} := Q$; a factorization $G = BR^{-1}B^T$ is given by $B = I_3$ and $R = \delta$. This is used in both the FORTRAN 77 and MATLAB implementations if a factored form is required.

As solution we get

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

where

$$\begin{aligned} x_1 &= \delta, \\ x_2 &= \delta\,\frac{(1+\sqrt{5})}{2}, \\ x_3 &= \delta\,\frac{(9+\sqrt{85})}{2}. \end{aligned}$$

The closed-loop spectrum is given by $\lambda_1 = 0$, $\lambda_2 = \dfrac{(3-\sqrt{5})}{2}$, and $\lambda_3 = \dfrac{(11-\sqrt{85})}{6}$.

For growing δ, the corresponding symplectic pencil (2) becomes more and more badly scaled which leads to a significant loss of accuracy in all DARE solvers based on eigenvalue methods. This demonstrates the need to use an appropriate scaling as proposed in [14].

Example 2.5 [5, Example 14], [8, 29]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
4	1	1	$\tau = 2.0$, $D = 1.0$, $K = 2.0$, $r = 0.25$	∞	9.6×10^{-2}	1.1	1.1	6.2
			$\underline{\tau = 10^8,\ D = 1.0,\ K = 1.0,\ r = 0.25}$	∞	$\approx 1 - \sqrt{5} \times 10^{-8}$	3.1×10^7	3.1×10^7	1.8×10^8
			$\tau = 10^{-6}$, $D = 1.0$, $K = 1.0$, $r = 0.25$	∞	2.0×10^{-7}	1.3	1.3	4.2×10^{12}

The following system describes a very simple process control of a paper machine. The continuous-time model with a time delay is sampled at intervals of length D which yields a singular transition matrix A. The time delay is equal to the length of three sampling intervals. The other parameters defining the system are a first-order time constant τ and the steady-state gain K. The linear system (7)–(8) is then given by

$$A = \begin{bmatrix} 1-D/\tau & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} KD/\tau \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}.$$

The weighting matrices used in this example are $R = r$ and $\tilde{Q} = 1$. Defining

$$\alpha = 1 - \frac{D}{\tau}, \qquad \beta = \frac{KD}{\tau},$$

it can be shown that the solutions of the DARE (1) are given by

$$X = \operatorname{diag}(x_i, 1, 1, 1)$$

where x_i, $i = 1, 2$, solve the scalar quadratic equation

$$(\alpha^2 - 1)x + 1 - \frac{\alpha^2\beta^2}{r + \beta^2 x}x^2 = 0,$$

whence

$$x_i = \frac{1}{2\beta^2}\left(r(\alpha^2 - 1) + \beta^2 \pm \sqrt{\left(r(\alpha^2 - 1) + \beta^2\right)^2 + 4\beta^2 r}\right). \tag{18}$$

The stabilizing positive semidefinite solution of (1) is thus defined by the unique positive solution x_1 of (18) and the closed-loop eigenvalues are

$$\lambda_1 = \frac{\alpha r}{r + \beta^2 x_1} = \frac{(\tau - D)\tau r}{\tau^2 r + (DK)^2 x_1}, \qquad \lambda_2 = \lambda_3 = \lambda_4 = 0.$$

Due to the variety of parameters in this example, it is possible to investigate DAREs with critical properties in many aspects. Since these properties merely rely on α, β, and r, these effects can be produced by keeping K and D constant and varying τ (and r). Since $|\lambda^C_{max}| = \lambda_1$, for $\tau \gg D, K$ the largest closed-loop eigenvalue approaches the unit disk. For $\tau \ll D, K$ the norm and condition of X become large and the DARE becomes ill conditioned with respect to K_{DARE}.

3 Examples of scalable size without parameters

Currently, this section is empty.

4 Parameter-dependent examples of scalable size

Example 4.1 [5, Example 15], [29, Example 3]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
n	1	n	$n = 10$, $r = 1$	∞	0.0	10.0	10.0	11.0
			$\underline{n = 100,\ r = 1}$	∞	0.0	1.0×10^2	1.0×10^2	2.8×10^2

Consider the DARE defined by

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ & & & & 0 \\ 0 & & & 0 & 1 \\ 0 & \dots & & 0 & 0 \end{bmatrix} \in \mathbb{R}^{n\times n}, \qquad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \qquad R = r, \qquad Q = I_n.$$

The stabilizing solution has a very simple form, namely,

$$X = \mathrm{diag}\,(\,1,\,2,\,\dots,\,n\,).$$

The closed-loop eigenvalues are all zero, that is, the spectrum of the symplectic pencil $L - \lambda M$ in (2) is given by the generalized eigenvalues $\lambda_1 = \ldots = \lambda_n = 0$ and $\lambda_{n+1} = \ldots = \lambda_{2n} = \infty$.

This example can be used to test any DARE solver for growing dimension of the problem. The DARE condition number K_{DARE} increases only slowly and for any order of the DARE, $\|X\| = \kappa(X) = n$.

Note further that the choice of r does not influence the stabilizing solution X but for $r < 1$, the condition number K_{DARE} behaves like $1/r$.

Acknowledgments

We would like to express our gratitude to the co-authors of the first version of the benchmark collection DAREX, Volker Mehrmann and Alan J. Laub who have layed the basis for the collection. Moreover, we wish to thank Vasile Sima for providing some of the examples used in this collection and for his efforts in integrating the FORTRAN 77 subroutine DDAREX into SLICOT.

A The FORTRAN 77 subroutine DDAREX

This is the prolog of a FORTRAN 77 subroutine for generating all presented examples. The subroutine was documented according to standards for SLICOT[4] and the SLICOT benchmark collection [28, 27]. The routine will serve as the basis for the SLICOT benchmark routine `BB02AD`. Slight modifications of DDAREX.f due to the integration into the library routine `BB02AD` may be necessary.

For some of the examples, DDAREX reads the data from data files delivered together with DDAREX.f. These are Examples 1.5–1.8, 1.10, 1.11, and 1.13. The corresponding data files are, according to the naming convention proposed in [27],

```
BB02105.DAT  BB02106.DAT  BB02107.DAT  BB02108.DAT
BB02110.DAT  BB02111.DAT  BB02113.DAT
```

Besides calls to LAPACK[5] and BLAS[6] [2], DDAREX calls the subroutines DSP2SY and DSY2SP which are used to convert symmetric matrices from general storage mode to packed storage mode and vice versa. These subroutines are provided in the file DDAREX.f.

The SLICOT subroutine BB02AD together with an example program calling BB02AD, test data and results as well as online documentation can be found in the benchmark chapter of SLICOT at `ftp://wgs.esat.kuleuven.ac.be/pub/WGS/SLICOT/libindex.html#B`.

```
      SUBROUTINE DDAREX(DEF, NR, DPAR, IPAR, BPAR, CHPAR, VEC, N, M, P,
     1                  A, LDA, B, LDB, C, LDC, Q, LDQ, R, LDR, S, LDS,
     2                  X, LDX, DWORK, INFO)
C     .. Scalar Arguments ..
      INTEGER          LDA, LDB, LDC, LDQ, LDR, LDS, LDX, INFO, N, M, P
      LOGICAL          BPAR(7), VEC(10)
      CHARACTER        DEF
C
C     .. Array Arguments ..
      INTEGER          NR(2), IPAR(3)
      DOUBLE PRECISION A(LDA,*), B(LDB,*), C(LDC,*), Q(*), R(*),
     1                 S(LDS,*), X(LDX,*), DPAR(*), DWORK(*)
      CHARACTER        CHPAR*255
C
C
C     PURPOSE
C
C     To generate benchmark examples for the numerical solution of
C     discrete-time algebraic Riccati equations (DAREs) of the form
C
```

[4] **S**ubroutine **LI**brary in **CO**ntrol and Systems **T**heory
[5] Available from `http://www.netlib.org/lapack`
[6] Available from `http://www.netlib.org/blas`

```
C             T                 T                T    -1  T       T
C      0  =  A X A  -  X  -  (A X B + S) (R + B X B)  (B X A + S )  +  Q
C
C      as presented in [1]. Here, A,Q,X are real N-by-N matrices, B,S are
C      N-by-M, and R is M-by-M. The matrices Q and R are symmetric and Q
C      may be given in factored form
C
C                    T
C      (I)    Q  =  C Q0 C .
C
C      Here, C is P-by-N and Q0 is P-by-P. If R is nonsingular and S = 0,
C      the DARE can be rewritten equivalently as
C
C                   T              -1
C      0  =  X  -  A X (I_n + G X)  A  -  Q
C
C      where I_n is the N-by-N identity matrix and
C
C                    -1  T
C      (II)   G = B R   B .
C
C      NOTE: the formulation of the DARE and the related matrix (pencils)
C            used here does not include DAREs as they arise in robust
C            control (H_infinity optimization).
C
C
C      ARGUMENTS
C
C
C      Mode Parameters
C
C      DEF     CHARACTER
C              This parameter specifies if the default parameters are
C              to be used for the example.
C              = 'N' or 'n' : The parameters given in the input vectors
C                             xPAR (x = 'D', 'I', 'B', 'CH') are used.
C              = 'D' or 'd' : The default parameters for the example
C                             are used.
C              This parameter is not referenced if NR(1) = 1.
C
C      Input/Output Parameters
C
C      NR      (input) INTEGER array, DIMENSION 2
C              This array determines the example for which DAREX returns
C              data. NR(1) is the group of examples.
```

```
C             NR(1) = 1 : parameter-free problems of fixed size.
C             NR(1) = 2 : parameter-dependent problems of fixed size.
C             NR(1) = 3 : parameter-free problems of scalable size.
C             NR(1) = 4 : parameter-dependent problems of scalable size.
C             NR(2) is the number of the example in group NR(1).
C             Let NEXi be the number of examples in group i. Currently,
C             NEX1 = 13, NEX2 = 5, NEX3 = 0, NEX4 = 1
C             1 .LE. NR(1), NR(1) .LE. 4
C             1 .LE. NR(2), NR(2) .LE. NEXi, where i = NR(1)
C
C     DPAR    (input/output) DOUBLE PRECISION array, DIMENSION 4
C             Double precision parameter vector. For explanation of the
C             parameters see [1].
C             DPAR(1) defines the parameters 'epsilon' for
C             examples NR = 2.2,2.3,2.4, the parameter 'tau'
C             for NR = 2.5, and the 1-by-1 matrix R for NR = 2.1,4.1.
C             For Example 2.5, DPAR(2) - DPAR(4) define in
C             consecutive order 'D', 'K', and 'r'.
C             NOTE that DPAR is overwritten with default values
C             if DEF = 'D' or 'd'.
C
C     IPAR    (input/output) INTEGER array, DIMENSION 3
C             On input, IPAR(1) determines the actual state dimension,
C             i.e., the order of the matrix A as follows:
C             NR(1) = 1, NR(1) = 2   : IPAR(1) is ignored.
C             NR = NR(1).NR(2) = 4.1 : IPAR(1) determines the order of the
C                                      output matrix A.
C             NOTE that IPAR(1) is overwritten for Examples 1.1-2.3. For
C             the other examples, IPAR(1) is overwritten if the default
C             parameters are to be used.
C             On output, this integer contains the order of the matrix A.
C
C             On input, IPAR(2) is the number of colums in the matrix B
C             and the order of the matrix R (in control problems, the
C             number of inputs of the system). Currently, IPAR(2) is
C             fixed for all examples and thus is not referenced on input.
C             On output, IPAR(2) is the number of columns of the
C             matrix B from (I).
C
C             On input, IPAR(3) is the number of rows in the matrix C
C             (in control problems, the number of outputs of the system).
C             Currently, IPAR(3) is fixed for all examples and thus is
C             not referenced on input.
C
C             NOTE that IPAR(2) and IPAR(3) are overwritten and
```

```
C            IPAR(2) .LE. IPAR(1) and IPAR(3) .LE. IPAR(1)  for all examples.
C
C     BPAR    (input) LOGICAL array, DIMENSION 7
C            This array defines the output of the examples and the
C            storage mode of the arrays Q and R.
C            BPAR(1) = .TRUE.  : Q is returned.
C            BPAR(1) = .FALSE. : Q is returned in factored form, i.e.,
C                                Q0 and C from (I) are returned.
C            BPAR(2) = .TRUE.  : The matrix returned in array Q (i.e.,
C                                Q if BPAR(1) = .TRUE. and Q0 if
C                                BPAR(1) = .FALSE.) is stored as full
C                                matrix.
C            BPAR(2) = .FALSE. : The matrix returned in array Q is
C                                provided in packed storage mode.
C            BPAR(3) = .TRUE.  : If BPAR(2) = .FALSE., the matrix
C                                returned in array Q is stored in upper
C                                packed mode, i.e., the upper triangle
C                                of a symmetric n-by-n matrix is stored
C                                by columns, e.g., the matrix entry
C                                Q(i,j) is stored in the array entry
C                                Q(i+j*(j-1)/2)  for i <= j.
C                                Otherwise, this entry is ignored.
C            BPAR(3) = .FALSE. : If BPAR(2) = .FALSE., the matrix
C                                returned in array Q is stored in lower
C                                packed mode, i.e., the lower triangle
C                                of a symmetric n-by-n matrix is stored
C                                by columns, e.g., the matrix entry
C                                Q(i,j) is stored in the array entry
C                                Q(i+(2*n-j)*(j-1)/2) for j <= i.
C                                Otherwise, this entry is ignored.
C            BPAR(4) = .TRUE.  : The product G in (II) is returned.
C            BPAR(4) = .FALSE. : G is returned in factored form, i.e.,
C                                B and R from (II) are returned.
C            BPAR(5) = .TRUE.  : The matrix returned in array R (i.e.,
C                                G if BPAR(4) = .TRUE. and R if
C                                BPAR(4) = .FALSE.) is stored as full
C                                matrix.
C            BPAR(5) = .FALSE. : The matrix returned in array R is
C                                provided in packed storage mode.
C            BPAR(6) = .TRUE.  : If BPAR(5) = .FALSE., the matrix
C                                returned in array R is stored in upper
C                                packed mode (see above).
C                                Otherwise, this entry is ignored.
C            BPAR(6) = .FALSE. : If BPAR(5) = .FALSE., the matrix
C                                returned in array R is stored in lower
```

```
C                              packed mode (see above).
C                              Otherwise, this entry is ignored.
C          BPAR(7) = .TRUE.  : The coefficient matrix S of the DARE is
C                              returned in array S.
C          BPAR(7) = .FALSE. : The coefficient matrix S of the DARE is
C                              not returned.
C
C  CHPAR   (output) CHARACTER*255.
C          On output, this string contains short information about
C          the chosen  example.
C
C  VEC     (output) LOGICAL array, DIMENSION 10.
C          VEC is a flag vector which indicates the availability of
C          output data for a given example. If on exit
C          Flag vector which displays the availability of the output
C          data:
C          VEC(j), j=1,2,3, refer to N, M, and P, respectively, and
C          are always .TRUE.
C          VEC(4) refers to A and is always .TRUE.
C          VEC(5) is .TRUE. if BPAR(4) = .FALSE., i.e., the factors B
C          and R from (II) are returned.
C          VEC(6) is .TRUE. if BPAR(1) = .FALSE., i.e., the factors C
C          and Q0 from (II) are returned.
C          VEC(7) refers to Q and is always .TRUE.
C          VEC(8) refers to R and is always .TRUE.
C          VEC(9) is .TRUE. if BPAR(7) = .TRUE., i.e., the matrix S
C          is returned.
C          VEC(10) refers to X and is = .TRUE. if the exact solution
C          matrix is available.
C          NOTE that VEC(i) = .FALSE. for i = 1 to 10 if on exit
C          INFO .NE. 0.
C
C  N       (output) INTEGER
C          The order of the matrices A, X, G if BPAR(4) = .TRUE., and
C          Q if BPAR(1) = .TRUE.
C
C  M       (output) INTEGER
C          The number of columns in the matrix B (or the dimension of
C          the control input space of the underlying dynamical
C          system).
C
C  P       (output) INTEGER
C          The number of rows in the matrix C (or the dimension of
C          the output space of the underlying dynamical system).
C
```

```
C     A       DOUBLE PRECISION array, DIMENSION (LDA,N)
C             The leading N-by-N part of this array contains the
C             coefficient matrix A of the DARE.
C
C     LDA     INTEGER
C             The leading dimension of array A as declared in the
C             calling program.
C             LDA .GE. N.
C
C     B       DOUBLE PRECISION array, DIMENSION (LDB,M).
C             If (BPAR(4) = .FALSE.), then the leading N-by-M part
C             of array B contains the coefficient matrix B of the DARE.
C             Otherwise, B is used as workspace.
C
C     LDB     INTEGER
C             The leading dimension of array B as declared in the
C             calling program.
C             LDB .GE. N.
C
C     C       DOUBLE PRECISION array, DIMENSION (LDC,N)
C             If (BPAR(1) = .FALSE.), then the leading P-by-N part
C             of array C contains the matrix C of the factored form (I)
C             of Q.  Otherwise, C is used as workspace.
C
C     LDC     INTEGER
C             The leading dimension of array C as declared in the
C             calling program.
C             LDC .GE. P.
C
C     Q       DOUBLE PRECISION array, DIMENSION at least NQ
C             If (BPAR(1) = .TRUE.) and (BPAR(2) = .TRUE.), then
C             NQ = LDQ*N.
C             IF (BPAR(1) = .TRUE.) and (BPAR(2) = .FALSE.), then
C             NQ = N*(N+1)/2.
C             If (BPAR(1) = .FALSE.) and (BPAR(2) = .TRUE.), then
C             NQ = LDQ*P.
C             IF (BPAR(1) = .FALSE.) and (BPAR(2) = .FALSE.), then
C             NQ = M*(M+1)/2.
C             The symmetric matrix contained in array Q is stored
C             according to BPAR(2) and BPAR(3).
C
C     LDQ     INTEGER
C             If conventional storage mode is used for Q, i.e.,
C             BPAR(2) = .TRUE., then Q is stored like a 2-dimensional
C             array with leading dimension LDQ. If packed symmetric
```

```
C            storage mode is used, then LDQ is not referenced.
C            LDQ .GE. N if BPAR(1) = .TRUE..
C            LDQ .GE. P if BPAR(1) = .FALSE.
C
C     R      DOUBLE PRECISION array, DIMENSION at least MR
C            If (BPAR(4) = .TRUE.) and (BPAR(5) = .TRUE.), then
C            MR = LDR*N.
C            IF (BPAR(4) = .TRUE.) and (BPAR(5) = .FALSE.), then
C            MR = N*(N+1)/2.
C            If (BPAR(4) = .FALSE.) and (BPAR(5) = .TRUE.), then
C            MR = LDR*M.
C            IF (BPAR(4) = .FALSE.) and (BPAR(5) = .FALSE.), then
C            MR = M*(M+1)/2.
C            The symmetric matrix contained in array R is stored
C            according to BPAR(5) and BPAR(6).
C
C     LDR    INTEGER
C            If conventional storage mode is used for R, i.e.,
C            BPAR(5)  = .TRUE., then R is stored like a 2-dimensional
C            array with leading dimension LDR. If packed symmetric
C            storage mode is used, then LDR is not referenced.
C            LDR .GE. N  if BPAR(4) =  .TRUE.
C            LDR .GE. M  if BPAR(4) = .FALSE.
C
C     S      DOUBLE PRECISION array, DIMENSION (LDS,M)
C            If (BPAR(7) = .TRUE.), then the leading N-by-M part of
C            array S contains the coefficient matrix S of the DARE.
C
C     LDS    INTEGER
C            The leading dimension of array S as declared in the
C            calling program.
C            LDS .GE. N if BPAR(7) = .TRUE.
C
C     X      DOUBLE PRECISION array, DIMENSION (LDX,NX).
C            If an exact solution is available (NR = 1.1,1.3,1.4,2.1,
C            2.3,2.4,2.5,4.1), then NX = N and the leading N-by-N part
C            of this array contains the solution matrix X.
C            Otherwise, X is not referenced.
C
C     LDX    INTEGER.
C            The leading dimension of array X as declared in the
C            calling program.
C            LDX .GE. N  if an exact solution is available.
C
C
```

```
C     Work Space
C
C     DWORK   DOUBLE PRECISION array, DIMENSION at least N*N
C
C
C     Error Indicator
C
C     INFO    INTEGER.
C             = 0 : successful exit;
C             < 0 : if INFO = -i, the argument no. i had an illegal value.
C             = 1 : Data file could not be opened or had wrong format.
C             = 2 : Division by zero.
C             = 3 : G can not be computed as in (II) due to a singular R
C                   matrix. This error can only occur if
C                   (BPAR(4) .EQ. .TRUE.).
C
C     REFERENCES
C
C     [1] J. ABELS and P. BENNER
C         DAREX - A Collection of Benchmark Examples for Discrete-Time
C         Algebraic Riccati Equations (Version 2.0)
C         SLICOT Working Note 1999-16, December 1999. Available from
C         http://www.win.tue.nl/niconet/NIC2/reports.html.
C
C     This is an updated and extended version of
C
C     [2] P. BENNER, A.J. LAUB and V. MEHRMANN
C         A Collection of Benchmark Examples for the Numerical Solution
C         of Algebraic Riccati Equations II: Discrete-Time Case.
C         Technical Report SPC 95_23, Fak. f. Mathematik,
C         TU Chemnitz-Zwickau (Germany), December 1995.
C
C     CONTRIBUTOR
C
C     Peter Benner (Universitaet Bremen)
C
C     For questions concerning the collection or for the submission of
C     test examples, please send e-mail to benner@math.uni-bremen.de.
C
C     REVISIONS
C
C     1995, December 14.
C     1996, February 28.
C     1999, November 25.
C
```

```
C     KEYWORDS
C
C     discrete-time, algebraic Riccati equation
C
```

B Reference table

The following table summarizes the properties of all the presented examples. A value "∞" for a condition number means that the corresponding matrix is not invertible with respect to the numerical rank computed by MATLAB. A hyphen "–" in the column for K_{DARE} indicates that the DARE condition number from [14] is not defined for this example. The column X^* indicates whether an analytical stabilizing solution is available ("+") or not ("–").

no.	n	m	p	default	X^*	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
1.1	2	1	1	–	+	5.8	0.0	1.0	1.0	–
1.2	2	2	2	–	–	∞	0.69	1.3×10^2	2.8×10^3	–
1.3	2	1	2	–	+	∞	0.38	5.2	1.1×10^2	1.9
1.4	3	2	3	–	+	∞	0.0	1.0×10^5	∞	–
1.5	4	2	4	–	–	1.0	0.94	35.4	3.3	30.6
1.6	4	2	4	–	–	19.9	0.99	2.1	1.8×10^2	7.9×10^2
1.7	4	4	4	–	–	3.8×10^2	$\approx 1 - \frac{1.8}{10^5}$	65.8	6.8×10^{12}	5.1×10^4
1.8	5	2	5	–	–	23.5	0.98	73.9	73.7	1.0×10^2
1.9	6	2	2	–	–	∞	0.67	2.5	37.4	3.9
1.10	9	3	2	–	–	1.6×10^6	0.96	6.1×10^2	∞	74.2
1.11	11	2	4	–	–	∞	0.80	4.4×10^4	9.8×10^5	1.9×10^3
1.12	13	2	4	–	–	∞	0.81	4.0×10^3	9.0×10^4	1.3×10^2
1.13	26	6	12	–	–	6.0×10^2	0.97	1.8×10^4	1.3×10^7	5.7×10^6
2.1	2	1	2	$\delta = 10^6$	+	1.1×10^2	0.999	1.3×10^4	∞	3.9×10^4
2.2	2	2	2	$\delta = 10^6$	–	1.1	0.91	0.1	17.9	3.7×10^7
2.3	2	1	2	$\delta = 10^6$	+	∞	0.0	1.0×10^{12}	1.0×10^{12}	2.7
2.4	3	3	3	$\delta = 10^6$	+	∞	0.38	9.1×10^6	9.1	2.5
2.5	4	1	1	$\tau = 10^8$, $D = 1.0$ $K = 1.0$, $r = 0.25$	+	∞	$\approx 1 - \frac{\sqrt{5}}{10^8}$	3.1×10^7	3.1×10^7	1.8×10^8
4.1	n	1	n	$n = 100$, $r = 1$	+	∞	0.0	1.0×10^2	1.0×10^2	2.8×10^2

References

[1] G. Ackerson and K. Fu. On the state estimation in switching environments. *IEEE Trans. Automat. Control*, AC-15:10–17, 1970.

[2] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, PA, second edition, 1995.

[3] W.F. Arnold, III and A.J. Laub. Generalized eigenproblem algorithms and software for algebraic Riccati equations. *Proc. IEEE*, 72:1746–1754, 1984.

[4] Z. Bai, J. Demmel, and M. Gu. Inverse free parallel spectral divide and conquer algorithms for nonsymmetric eigenproblems. Technical Report LBL-34969, Lawrence Berkeley Laboratory, University of California, Berkeley, CA 94720, 1994.

[5] P. Benner, A.J. Laub, and V. Mehrmann. A collection of benchmark examples for the numerical solution of algebraic Riccati equations II: Discrete-time case. Technical Report SPC 95_23, Fakultät für Mathematik, TU Chemnitz–Zwickau, 09107 Chemnitz, FRG, 1995. Available from `http://www.tu-chemnitz.de/sfb393/spc95pr.html`.

[6] P. Benner, A.J. Laub, and V. Mehrmann. A collection of benchmark examples for the numerical solution of algebraic Riccati equations I: Continuous-time case. Technical Report SPC 95_22, Fakultät für Mathematik, TU Chemnitz–Zwickau, 09107 Chemnitz, FRG, 1995. Available from `http://www.tu-chemnitz.de/sfb393/spc95pr.html`.

[7] P. Benner, A.J. Laub, and V. Mehrmann. Benchmarks for the numerical solution of algebraic Riccati equations. *IEEE Control Systems Magazine*, 7(5):18–28, 1997.

[8] W.L. Bialkowski. Application of steady-state Kalman filters–Theory with field results. In *Proc. Joint Automat. Cont. Conf.*, Philadelphia, PA, 1978.

[9] E.J. Davison and S.H. Wang. Properties and calculation of transmission zeros of linear multivariable systems. *Automatica*, 10:643–658, 1974.

[10] C. Foulard, S. Gentil, and J.P. Sandraz. *Commande et régulation par calculateur numérique: De la théorie aux applications*. Eyrolles, Paris, 1977.

[11] Z. Gajić and X. Shen. *Parallel Algorithms for Optimal Control of Large Scale Linear Systems*. Springer-Verlag, London, GB, 1994.

[12] J.D. Gardiner and A.J. Laub. A generalization of the matrix-sign-function solution for algebraic Riccati equations. *Internat. J. Control*, 44:823–832, 1986.

[13] K. Gomathi, S. Prabhu, and M. Pai. A suboptimal controller for minimum sensitivity of closed-loop eigenvalues to parameter variations. *IEEE Trans. Automat. Control*, AC-25:587–588, 1980.

[14] T. Gudmundsson, C. Kenney, and A.J. Laub. Scaling of the discrete-time algebraic Riccati equation to enhance stability of the Schur solution method. *IEEE Trans. Automat. Control*, 37:513–518, 1992.

[15] G.A. Hewer. An iterative technique for the computation of steady state gains for the discrete optimal regulator. *IEEE Trans. Automat. Control*, AC-16:382–384, 1971.

[16] V. Ionescu and M. Weiss. On computing the stabilizing solution of the discrete-time Riccati equation. *Linear Algebra Appl.*, 174:229–238, 1992.

[17] E. Jonckheere. On the existence of a negative semidefinite antistabilizing solution to the discrete algebraic Riccati equation. *IEEE Trans. Automat. Control*, AC-26:707–712, 1981.

[18] T. Katayama, T. Ohki, T. Inoue, and T. Kato. Design of an optimal controller for a discrete-time system subject to previewable demand. *Internat. J. Control*, 41(3):677–699, 1985.

[19] M.M. Konstantinov, P.Hr. Petkov, and N.D. Christov. Perturbation analysis of the discrete Riccati equation. *Kybernetica*, 29:18–29, 1993.

[20] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems.* Wiley-Interscience, New York, 1972.

[21] A.J. Laub. A Schur method for solving algebraic Riccati equations. *IEEE Trans. Automat. Control*, AC-24:913–921, 1979.

[22] A.J. Laub. Algebraic aspects of generalized eigenvalue problems for solving Riccati equations. In C.I. Byrnes and A. Lindquist, editors, *Computational and Combinatorial Methods in Systems Theory*, pages 213–227. Elsevier (North-Holland), 1986.

[23] B. Litkouhi. *Sampled-Data Control of Systems with Slow and Fast Modes.* PhD thesis, Michigan State University, 1983.

[24] L.Z. Lu and W.W. Lin. An iterative algorithm for the solution of the discrete time algebraic Riccati equation. *Linear Algebra Appl.*, 188/189:465–488, 1993.

[25] V. Mehrmann. *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution.* Number 163 in Lecture Notes in Control and Information Sciences. Springer-Verlag, Heidelberg, July 1991.

[26] V. Mehrmann. A step towards a unified treatment of continuous and discrete time control problems. *Linear Algebra Appl.*, to appear, 1996. (see also: *Preprint SPC 94_20, Fak. f. Mathematik, TU Chemnitz-Zwickau, Chemnitz, FRG*).

[27] V. Mehrmann and Thilo Penzl. Benchmark collections in SLICOT. SLICOT Working Note 1998-5, The Working Group on Software, June 1998. Available from `http://www.win.tue.nl/niconet/NIC2/reports.html`.

[28] The Working Group on Software. SLICOT implementation and documentation standards 2.1. WGS-report 96-1, Available from `http://www.win.tue.nl/wgs/reports.html`, 1996.

[29] T. Pappas, A.J. Laub, and N.R. Sandell. On the numerical solution of the discrete-time algebraic Riccati equation. *IEEE Trans. Automat. Control*, AC-25:631–641, 1980.

[30] L.M. Patnaik, N. Viswanadham, and I.G. Sarma. Computer control algorithms for a tubular ammonia reactor. *IEEE Trans. Automat. Control*, AC-25(4):642–651, 1980.

[31] P.H. Petkov, N.D. Christov, and M.M. Konstantinov. Numerical properties of the generalized Schur approach for solving the discrete matrix Riccati equation. In *Proc. 18th Spring Conference of the Union of Bulgarian Mathematicians*, pages 452–457, Albena, 1989.

[32] P.H. Petkov, N.D. Christov, and M.M. Konstantinov. A posteriori error analysis of the generalized Schur approach for solving the discrete matrix Riccati equation. Preprint, Department of Automatics, Higher Institute of Mechanical and Electrical Engineering, 1756 Sofia, Bulgaria, 1989.

[33] A.P. Sage. *Optimum Systems Control.* Prentice-Hall, Englewood Cliffs, NJ, 1966.

[34] V.G. Shankar and K. Ramar. Pole assignment with minimum eigenvalue sensitivity to parameter variations. *Internat. J. Control*, 23:493–504, 1976.

[35] V. Sima. *Algorithms for Linear-Quadratic Optimization*, volume 200 of *Pure and Applied Mathematics.* Marcel Dekker, Inc., New York, NY, 1996.

[36] J.-G. Sun. Backward error for the discrete-time algebraic Riccati equation. *Linear Algebra Appl.*, 259:183–208, 1997.

[37] J.-G. Sun. Sensitivity analysis of the discrete-time algebraic Riccati equation. *Linear Algebra Appl.*, 275/276:595–615, 1998.

[38] P. Van Dooren. A generalized eigenvalue approach for solving Riccati equations. *SIAM J. Sci. Statist. Comput.*, 2:121–135, 1981.