Technical Sciences
Academy of Romania
www.jesi.astr.ro

Journal of Engineering Sciences and Innovation
Volume 7, Issue 1 / 2022, pp. 71-86
http://doi.org/10.56958/jesi.2022.7.1.71

C. Chemical Engineering, Materials Science and Engineering

Received **12 September 2021** *Accepted* **7 March 2022**
Received in revised form **26 January 2022**

Numerical solution of stable generalized complex Lyapunov equations

VASILE SIMA[*]

National Institute for Research & Development in Informatics, Bucharest, Romania
Technical Sciences Academy of Romania

Abstract. Generalized Lyapunov equations are often encountered in systems theory, analysis and design of control systems, and in many applications, including balanced realization algorithms, procedures for reduced order models, or Newton methods for generalized algebraic Riccati equations. An important application is the computation of the Hankel singular values of a generalized dynamical system, whose behavior is defined by a regular matrix pencil (E, A), with E nonsingular. This application uses the controllability and observability Gramians of the system, given as the solutions of a pair of related generalized Lyapunov equations. For a stable system, the solutions of both equations are non-negative definite. The paper summarizes the numerical algorithms for complex continuous- and discrete-time generalized systems. Such solvers are not yet available in the SLICOT Library or MATLAB toolboxes, but could be an important addition. The developed solvers address the essential practical issues of reliability, accuracy, and efficiency.

Keywords: linear multivariable systems, lyapunov equation, numerical methods, software, stability.

1. Introduction

Stable generalized complex Lyapunov equations can be written as

$$\mathrm{op}(A)^H X \mathrm{op}(E) + \mathrm{op}(E)^H X \mathrm{op}(A) = -\mathrm{op}(B)^H \mathrm{op}(B), \qquad (1)$$

$$\mathrm{op}(A)^H X \mathrm{op}(A) - \mathrm{op}(E)^H X \mathrm{op}(E) = -\mathrm{op}(B)^H \mathrm{op}(B), \qquad (2)$$

in the continuous- and discrete-time case, respectively, where A, $E \in \boldsymbol{C}^{n\times n}$, $\mathrm{op}(B) \in \boldsymbol{C}^{m\times n}$, the operator $\mathrm{op}(M)$ is either M or $\mathrm{op}(M)^H$ for any matrix M, and the

[*]Correspondence address: vasilesima@ymail.com

superscript H denotes the conjugate transpose. (In the real case, H is replaced by T, denoting transposition.) A necessary condition for the nonsingularity of the associated linear algebraic systems is that both A and E, for (1), or either A or E, for (2), are nonsingular. Since A and E have a symmetric role in (2), it may be assumed, without loss of generality, that E is nonsingular. The stability assumption means that $\Lambda(AE^{-1}) \in \boldsymbol{C}_-$, for (1), and $\varrho(AE^{-1}) < 1$, for (2), where $\boldsymbol{C}_-$ is the open left half of the complex plane, and $\Lambda(M)$ and $\varrho(M)$ are the spectrum and the *spectral radius* (i.e., the maximum moduli of the eigenvalues) of the matrix M, respectively. (See, e.g., [1] and the references therein.) Equivalently, the matrix pencil $A - \lambda E$ has only stable eigenvalues in the continuous- or discrete-time sense. Note that the nonsingularity of E implies the regularity of $A - \lambda E$. These stable equations have a unique positive-semidefinite solution X, denoted $X \geq 0$, since $\mathrm{op}(B)^H \mathrm{op}(B) \geq 0$. Then, X can be written in a factorized form, $X = \mathrm{op}(U)^H \mathrm{op}(U)$, where U is the Cholesky factor of X, if $X > 0$. It should be noted that any matrix expressed in the form $\mathrm{op}(B)^H \mathrm{op}(B)$ has real non-negative diagonal elements, since these elements are given by $b_j^H b_j = \left\|b_j\right\|^2 \geq 0$, where b_j is the j-th column of $\mathrm{op}(B)$, and $\|x\|$ is the Euclidean norm of the vector x. For an identity matrix E, $E = I_n$, and $\mathrm{op}(M) = M$, the *standard* stable Lyapunov equations are obtained, dealt with in [2]. Algorithms for solving real generalized Lyapunov equations have been proposed in [3]. These algorithms belong to the class of transformation methods, described in the seminal paper [4]. Solvers implementing these algorithms are available, e.g., in the SLICOT Library [5], [6] and in MATLAB Control System Toolbox [7] (based on SLICOT). Many algorithmic and computational details for Sylvester and standard Lyapunov equations are given, e.g., in [8]. General linear matrix equations are dealt with in [9].

This paper extends the results in [3] to complex equations. The complex case is theoretically simpler than the real case, since in the latter case the generalized Schur form (used to reduce the computational effort from the order of n^6 operations to an order of n^3 operations) can have 2×2 blocks on the diagonal, corresponding to complex conjugate eigenvalues, while in the former case, all diagonal blocks are 1×1. However, the computational details should be carefully considered to obtain accurate and reliable solutions. Important issues in this endeavour are presented in this paper, accompanying the theoretical derivations of the needed formulas.

This section is ended by presenting an important application: computation of the Hankel singular values of a dynamical system, which are essential input-output invariants. This application needs both forms of op(·). Specifically, consider a generalized system,

$$E\lambda\big(x(t)\big) = Ax(t) + Bu(t), \;\; y(t) = Cx(t) \tag{3}$$

where $x(t) \in \boldsymbol{C}^n$, $B \in \boldsymbol{C}^{n\times m}$, $C \in \boldsymbol{C}^{p\times n}$, and $\lambda\big(x(t)\big)$ is the differential operator, $\mathrm{d}x(t)/\mathrm{d}t$, or the advance difference operator, $\lambda\big(x(t)\big) = x(t+1)$, for continuous- and discrete-time case, respectively. The *Hankel singular values* of (3) are the non-negative square roots of the eigenvalues of the matrix product QP, where P and Q are the *controllability* and *observability Gramians*, respectively, of (3), i.e., the solutions of the two closely related generalized Lyapunov equations,

$$APE^H + EPA^H = -BB^H, \quad A^HQE + E^HQA = -C^HC, \tag{4}$$

$$APA^H - EPE^H = -BB^H, \quad A^HQA - E^HQE = -C^HC, \tag{5}$$

in the two cases, respectively. For a stable system, the product QP has theoretically only non-negative eigenvalues. But numerical computations performed without taking into account the symmetry and semidefiniteness of the solutions, might result in nonsymmetric or indefinite Gramians, due to accumulated rounding errors. Consequently, some computed Hankel singular values might appear as negative or even complex numbers. This proves how important is to ensure the reliability and accuracy of the computations. For this application, it is preferable to use the algorithms described below, which deliver the Choleky factors R_c and R_o of the Gramians, $P = R_cR_c^H$, $Q = R_o^HR_o$, with R_c and R_o upper triangular. Moreover, the matrix products BB^H and C^HC are not evaluated, and B and C are directly used. Then, the Hankel singular values of the system are the singular values of the product R_oR_c. Consequently, they are numerically guaranteed to be real non-negative.

2. Basic computational steps

Reduction to generalized Schur form. For general matrices A and E, the first step in solving (1) or (2) is the computation of the (complex) generalized Schur form (GSF) of the matrix pencil $A - \lambda E$, using the QZ algorithm, see, e.g., [10], [11] and the references therein. In the complex case, the QZ algorithm returns the reduced matrices, $\tilde{A}$ and $\tilde{E}$, as well as the unitary transformation matrices, $Q, Z \in \boldsymbol{C}^{n\times n}$, $Q^HQ = QQ^H = I_n$, $Z^HZ = ZZ^H = I_n$, so that

$$\tilde{A} = Q^HAZ, \quad \tilde{E} = Q^HEZ, \tag{6}$$

with both $\tilde{A}$ and $\tilde{E}$ upper triangular. Moreover, without loss of generality, the matrices are transformed so that the diagonal elements of $\tilde{E}$ be real non-negative. The diagonal elements of $\tilde{A}$ and $\tilde{E}$ define the eigenvalues λ_i of the pencil as rational complex numbers with numerators $\tilde{a}_{ii}$ and denominators $\tilde{e}_{ii}$.

Transformation of the right hand side. If $\mathrm{op}(\cdot) = \cdot$, then premultiplying (1) and (2) by Z^H and postmultiplying them by Z, and using the fact that Q and Z are unitary matrices, the following equations are obtained:

$$\mathrm{op}(\tilde{A})^H \tilde{X} \mathrm{op}(\tilde{E}) + \mathrm{op}(\tilde{E})^H \tilde{X} \mathrm{op}(\tilde{A}) = -\mathrm{op}(\hat{B})^H \mathrm{op}(\hat{B}), \tag{7}$$

$$\mathrm{op}(\tilde{A})^H \tilde{X} \mathrm{op}(\tilde{A}) - \mathrm{op}(\tilde{E})^H \tilde{X} \mathrm{op}(\tilde{E}) = -\mathrm{op}(\hat{B})^H \mathrm{op}(\hat{B}), \tag{8}$$

where $\tilde{X} := Q^H X Q$ and $\hat{B} := BZ$. Similarly, if $\mathrm{op}(\cdot) = \cdot^H$, then premultiplying (1) and (2) by Q^H and postmultiplying them by Q, the equations (7) and (8) are obtained, with $\tilde{X} := Z^H X Z$ and $\hat{B} := B^H Q$. The matrix $\hat{B}$ is not used directly, but after a transformation into a standardized form. Specifically, $\hat{B}$ is triangularized using QR or RQ factorizations if $\mathrm{op}(\cdot) = \cdot$ or $\mathrm{op}(\cdot) = \cdot^H$, respectively,

$$\begin{aligned} &\begin{bmatrix} Q_B & 0 \\ 0 & I_{n-m} \end{bmatrix} \tilde{B} = \begin{bmatrix} \hat{B} \\ 0 \end{bmatrix}, \quad \text{if } m < n; \quad Q_B \begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} = \hat{B}, \quad \text{if } m \geq n; \\ &[\tilde{B} \quad 0] \begin{bmatrix} Q_B & 0 \\ 0 & I_{n-m} \end{bmatrix} = [\hat{B} \quad 0], \quad \text{if } m < n; \quad [\tilde{B} \quad 0] Q_B = \hat{B}, \quad \text{if } m \geq n, \end{aligned} \tag{9}$$

where Q_B is a unitary matrix given as a product of Householder transformations, but the product should not be computed. These computations make the diagonal elements of $\tilde{B}$ real numbers. Further scaling by -1 of the rows (if $\mathrm{op}(\cdot) = \cdot$) or columns (if $\mathrm{op}(\cdot) = \cdot^H$) of $\tilde{B}$ having negative elements in their diagonal positions, delivers the standardized form of $\tilde{B}$. The final *reduced* equations are then the following

$$\mathrm{op}(\tilde{A})^H \tilde{X} \mathrm{op}(\tilde{E}) + \mathrm{op}(\tilde{E})^H \tilde{X} \mathrm{op}(\tilde{A}) = -\mathrm{op}(\tilde{B})^H \mathrm{op}(\tilde{B}), \tag{10}$$

$$\mathrm{op}(\tilde{A})^H \tilde{X} \mathrm{op}(\tilde{A}) - \mathrm{op}(\tilde{E})^H \tilde{X} \mathrm{op}(\tilde{E}) = -\mathrm{op}(\tilde{B})^H \mathrm{op}(\tilde{B}). \tag{11}$$

Solution of the reduced equation. The solution of the reduced equations (10) and (11) is discussed in the next two sections. The result is obtained in a factorized form, $\tilde{X} = \mathrm{op}(\widetilde{U})^H \mathrm{op}(\widetilde{U})$, where $\widetilde{U}$ is upper triangular with real non-negative diagonal elements.

Solution of the original equation. Having the „Cholesky" factor, $\widetilde{U}$, the corresponding factor, U, of the solution X of the original equation with $\mathrm{op}(\cdot) = \cdot$ or $\mathrm{op}(\cdot) = \cdot^H$ is obtained using the QR or RQ factorization, respectively, as follows,

$$Q_U U = \widetilde{U} Q^H, \quad \text{if } \mathrm{op}(\cdot) = \cdot, \quad U Q_U = Z \widetilde{U}, \quad \text{if } \mathrm{op}(\cdot) = \cdot^H, \tag{12}$$

where Q_U is unitary and U is upper triangular with real non-negative diagonal elements. (The QR and RQ algorithms and their usual implementations return real diagonal elements. If $u_{ii} < 0$, the i-th row or column, respectively, is scaled by -1 to ensure non-negativity.)

3. Solving reduced stable generalized continuous-time complex Lyapunov equations

The solution of the reduced equations (10) is presented in this section. For convenience, the tilde signs are omitted. Note that all involved matrices, A, E, B, and the solution factor, U, are upper triangular and E, B, and U have real non-negative diagonal elements.

The case $\mathbf{op}(\boldsymbol{M}) = \boldsymbol{M}$. Consider first the case $\mathrm{op}(\cdot) = \cdot$ and the following matrix partition

$$A = \begin{bmatrix} a_{11} & a_{12} \\ 0 & A_{22} \end{bmatrix}, \quad E = \begin{bmatrix} e_{11} & e_{12} \\ 0 & E_{22} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} \\ 0 & B_{22} \end{bmatrix}, \quad U = \begin{bmatrix} u_{11} & u_{12} \\ 0 & U_{22} \end{bmatrix}, \tag{13}$$

where $a_{11} \in \boldsymbol{C}$, e_{11}, b_{11}, $u_{11} \in \boldsymbol{R}$, $a_{12}, e_{12}, b_{12}, u_{12} \in \boldsymbol{C}^{1\times(n-1)}$, and A_{22} , E_{22} , B_{22} , $U_{22} \in \boldsymbol{C}^{(n-1)\times(n-1)}$. Equation (1) becomes

$$\begin{bmatrix} \bar{a}_{11} & 0 \\ a_{12}^H & A_{22}^H \end{bmatrix} \begin{bmatrix} u_{11} & 0 \\ u_{12}^H & U_{22}^H \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & U_{22} \end{bmatrix} \begin{bmatrix} e_{11} & e_{12} \\ 0 & E_{22} \end{bmatrix} \tag{14}$$

$$+ \begin{bmatrix} e_{11} & 0 \\ e_{12}^H & E_{22}^H \end{bmatrix} \begin{bmatrix} u_{11} & 0 \\ u_{12}^H & U_{22}^H \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & U_{22} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ 0 & A_{22} \end{bmatrix} = - \begin{bmatrix} b_{11} & 0 \\ b_{12}^H & B_{22}^H \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ 0 & B_{22} \end{bmatrix}$$

($\bar{x}$ denotes the conjugate of x) and its solution can then be found recursively. Specifically, evaluating the (1,1), (2,1), and (2,2) elements of the left and right hand side expressions, (14) can be decomposed as

$$(\bar{a}_{11} + a_{11}) e_{11} u_{11}^2 = -b_{11}^2 \,, \tag{15}$$

$$u_{11}(e_{11} A_{22}^H + a_{11} E_{22}^H) u_{12}^H = -b_{11} b_{12}^H - u_{11}^2 (e_{11} a_{12}^H + a_{11} e_{12}^H) \,, \tag{16}$$

$$A_{22}^H U_{22}^H U_{22} E_{22} + E_{22}^H U_{22}^H U_{22} A_{22} + (u_{11} a_{12}^H + A_{22}^H u_{12}^H)(u_{11} e_{12} + u_{12} E_{22}) +$$
$$(u_{11} e_{12}^H + E_{22}^H u_{12}^H)(u_{11} a_{12} + u_{12} A_{22}) = -b_{12} b_{12}^H - B_{22}^H B_{22} \,. \tag{17}$$

These equations can be solved successively for u_{11} , u_{12}^H , and U_{22} , as shown below. Indeed, the equations above can be rewritten as

$$2\mathcal{R}(a_{11}) e_{11} u_{11}^2 = -b_{11}^2 \,, \tag{18}$$

$$(A_{22}^H + m_1 E_{22}^H) u_{12}^H = \; -m_2 b_{12}^H - u_{11}(a_{12}^H + m_1 e_{12}^H) \,, \tag{19}$$

$$A_{22}^H U_{22}^H U_{22} E_{22} + E_{22}^H U_{22}^H U_{22} A_{22} = -B_{22}^H B_{22} - y y^H \,, \tag{20}$$

where $\mathcal{R}(\alpha)$ denotes the real part of a complex number α, and

$$m_1 := \frac{a_{11}}{e_{11}}, \;\; m_2 =:= \frac{b_{11}}{e_{11} u_{11}} \,, \tag{21}$$

$$y := b_{12}^H - m_2 (u_{11} e_{12}^H + E_{22}^H u_{12}^H) \,. \tag{22}$$

From (18), it follows that

$$u_{11} = \frac{b_{11}}{\sqrt{-2\mathcal{R}(a_{11})e_{11}}} . \tag{23}$$

Note that $u_{11} \in \boldsymbol{R}$, since $b_{11} \geq 0$, $e_{11} > 0$, and, by the stability assumption, $a_{11} \in \boldsymbol{C}_-$. Moreover, $u_{11} > 0$, if $b_{11} > 0$, and $u_{11} = 0$, if $b_{11} = 0$. Equation (19) follows by dividing (16) by $u_{11}e_{11}$, if $u_{11} \neq 0$, and using (21). By a continuity argument, (19) holds also for $u_{11} = 0$; moreover, note that m_2 can be rewritten as $m_2 = \frac{\sqrt{-2\mathcal{R}(a_{11})e_{11}}}{e_{11}}$, which is defined also for $u_{11} = 0$. Hence,

$$m_2^2 = -\frac{2\mathcal{R}(a_{11})}{e_{11}} = -(m_1 + \bar{m}_1) . \tag{24}$$

The solution u_{12}^H is then obtained by solving a linear triangular system of equations (initially, of order $n - 1$) using forward substitution, see, e.g., [11]. Scaling is used to avoid overflows. Specifically, a system $Mx = sb$ is solved instead of $Mx = b$, where $s \in [\, 0, 1]$ is chosen so that the elements of the computed x are representable in a computer. Such a solver is available in the LAPACK package [10]; it checks for possible overflow or divide-by-zero at every operation, and scales x and s, if necessary. Usually, $s = 1$. If M is singular, then $s = 0$, and a non-trivial solution of $Mx = 0$ is obtained. If the unscaled problem will not cause overflows, a Level 2 BLAS algorithm (trsv) is used. Clearly, the current value of s should be used by the Lyapunov solver to update the current results, and the final s value should be returned.

Equation (20) is obtained from (17), noting that, with (22),

$$\begin{aligned} yy^H &= [\, b_{12}^H - m_2(u_{11}e_{12}^H + E_{22}^H u_{12}^H)\,]\,[\, b_{12} - m_2(u_{11}e_{12} + u_{12}E_{22})] \\ &= b_{12}^H b_{12} - m_2 b_{12}^H(u_{11}e_{12} + u_{12}E_{22}) - m_2(u_{11}e_{12}^H + E_{22}^H u_{12}^H)b_{12} \\ &\quad + m_2^2(u_{11}e_{12}^H + E_{22}^H u_{12}^H)(u_{11}e_{12} + u_{12}E_{22}) . \end{aligned} \tag{25}$$

But from (19), it follows that

$$u_{11}a_{12}^H + A_{22}^H u_{12}^H = -m_2 b_{12}^H - m_1(u_{11}e_{12}^H + E_{22}^H u_{12}^H) , \tag{26}$$

so that,

$$\begin{aligned} (u_{11}a_{12}^H + A_{22}^H u_{12}^H)(u_{11}e_{12} + u_{12}E_{22}) &= -m_2 b_{12}^H(u_{11}e_{12} + u_{12}E_{22}) \\ &\quad - m_1(u_{11}e_{12}^H + E_{22}^H u_{12}^H)(u_{11}e_{12} + u_{12}E_{22}) . \end{aligned} \tag{27}$$

Adding the conjugate transpose of (27), and using (17), (24), and (25), it follows that (20) holds. If $\tilde{B}_{22}$ is the square triangular factor of the QR factorization

$$\tilde{Q}\begin{bmatrix} \tilde{B}_{22} \\ 0 \end{bmatrix} = \begin{bmatrix} B_{22} \\ y^H \end{bmatrix} , \tag{28}$$

then the right hand side from (20) becomes $-\tilde{B}_{22}^H\tilde{B}_{22}$. Consequently, the corresponding equation has the same structure as the original equation (14), but its order is initially $n - 1$. Therefore, the same solution technique can be used recursively. The factorization in (28) is computed using $n-1$ Givens rotations [11].

The case op($\boldsymbol{M}$)= $\boldsymbol{M^H}$. Solving the equation (1) with $\mathrm{op}(M) = M^H$ is similar. However, the partition used for matrices is different, namely,

$$A = \begin{bmatrix} A_{22} & a_{12} \\ 0 & a_{11} \end{bmatrix}, \quad E = \begin{bmatrix} E_{22} & e_{12} \\ 0 & e_{11} \end{bmatrix}, \quad B = \begin{bmatrix} B_{22} & b_{12} \\ 0 & b_{11} \end{bmatrix}, \quad U = \begin{bmatrix} U_{22} & u_{12} \\ 0 & u_{11} \end{bmatrix}, \quad (29)$$

where $a_{11} \in \boldsymbol{C}$, e_{11}, b_{11}, $u_{11} \in \boldsymbol{R}$, $a_{12}, e_{12}, b_{12}, u_{12} \in \boldsymbol{C}^{(n-1)\times 1}$, and A_{22} , E_{22} , B_{22} , $U_{22} \in \boldsymbol{C}^{(n-1)\times(n-1)}$. With (29), the formulas for solving (1) can be obtained by taking the conjugate transpose of the relations above. The equation for u_{11} is the same, and the other two equations are

$$(A_{22} + \bar{m}_1 E_{22}) u_{12} = -m_2 b_{12} - u_{11}(a_{12} + \bar{m}_1 e_{12}) \,, \quad (30)$$

$$A_{22} U_{22} U_{22}^H E_{22}^H + E_{22} U_{22} U_{22}^H A_{22}^H = -B_{22} B_{22}^H - y y^H \,, \quad (31)$$

with $y := b_{12} - m_2(u_{11} e_{12} + E_{22} u_{12})$. Since the right hand side in (31) is $-B_{22}B_{22}^H - yy^H$, an RQ factorization of the matrix

$$[\tilde{B}_{22} \quad 0]\tilde{Q} = [B_{22} \quad y] \,, \quad (32)$$

is used, so that the right hand side becomes $-\tilde{B}_{22}\tilde{B}_{22}^H$.

4. Solving reduced stable generalized discrete-time complex Lyapunov equations

The case op($\boldsymbol{M}$) = $\boldsymbol{M}$. In a similar manner to the continuous-time case, the basic equations for the discrete-time case with $\mathrm{op}(\cdot) = \cdot$ are

$$\begin{bmatrix} \bar{a}_{11} & 0 \\ a_{12}^H & A_{22}^H \end{bmatrix} \begin{bmatrix} u_{11} & 0 \\ u_{12}^H & U_{22}^H \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & U_{22} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ 0 & A_{22} \end{bmatrix} \quad (33)$$

$$- \begin{bmatrix} e_{11} & 0 \\ e_{12}^H & E_{22}^H \end{bmatrix} \begin{bmatrix} u_{11} & 0 \\ u_{12}^H & U_{22}^H \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & U_{22} \end{bmatrix} \begin{bmatrix} e_{11} & e_{12} \\ 0 & E_{22} \end{bmatrix} = - \begin{bmatrix} b_{11} & 0 \\ b_{12}^H & B_{22}^H \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ 0 & B_{22} \end{bmatrix} .$$

It follows that u_{11} , u_{12}^H , and U_{22} can be obtained by solving the following equations

$$(|a_{11}|^2 - e_{11}^2) u_{11}^2 = -b_{11}^2 \,, \quad (34)$$

$$(m_1 A_{22}^H - E_{22}^H) u_{12}^H = -m_2 b_{12}^H + u_{11}(e_{12}^H - m_1 a_{12}^H) \,, \quad (35)$$

$$A_{22}^H U_{22}^H U_{22} A_{22} - E_{22}^H U_{22}^H U_{22} E_{22} + (u_{11} a_{12}^H + A_{22}^H u_{12}^H)(u_{11} a_{12} + u_{12} A_{22})$$
$$- (u_{11} e_{12}^H + E_{22}^H u_{12}^H)(u_{11} e_{12} + u_{12} E_{22}) = -b_{12}^H b_{12} - B_{22}^H B_{22} \,, \quad (36)$$

where m_1 and m_2 are defined in (21). The solution of (34) is

$$u_{11} = \frac{b_{11}}{\sqrt{e_{11}^2 - |a_{11}|^2}}\,, \qquad (37)$$

which is a real non-negative number, since the stability assumption implies $|a_{11}|/e_{11} < 1$. In this case, m_1 and m_2 satisfy the following relation

$$|m_1|^2 + m_2^2 = \frac{|a_{11}|^2}{e_{11}^2} + \frac{e_{11}^2 - |a_{11}|^2}{e_{11}^2} = 1\,. \qquad (38)$$

Define also a 2 × 2 matrix

$$M := I_2 - \begin{bmatrix} m_2 \\ m_1 \end{bmatrix} \begin{bmatrix} m_2 \\ m_1 \end{bmatrix}^H = \begin{bmatrix} 1 - m_2^2 & -\bar{m}_1 m_2 \\ -m_1 m_2 & 1 - |m_1|^2 \end{bmatrix} = \begin{bmatrix} |m_1|^2 & -\bar{m}_1 m_2 \\ -m_1 m_2 & m_2^2 \end{bmatrix} =: CC^H,$$

(39)

where CC^H is a factorization of M. Note that $M = M^H$, hence M is a Hermitian matrix and, therefore, its eigenvalues, λ_j , $j = 1, 2$, are real. But using (38) and (39),

$$\lambda_1 + \lambda_2 = |m_1|^2 + m_2^2 = 1, \quad \lambda_1 \lambda_2 = |m_1|^2 m_2^2 - |m_1|^2 m_2^2 = 0\,. \qquad (40)$$

Consequently, $\Lambda(M) = \{\,1, 0\,\}$, and considering the spectral decomposition of M, $M = V\Lambda(M)V^H$, it follows that $M = V_1 V_1^H$, where V_1 is the first column of V. Hence, the factor C of the rank-1 matrix M in (39) can be taken as the eigenvector of M corresponding to the unit eigenvalue. Defining now

$$y := [\, b_{12}^H \quad u_{11} a_{12}^H + A_{22}^H u_{12}^H \,] C\,, \qquad (41)$$

it is easy to show that (36) is equivalent to

$$A_{22}^H U_{22}^H U_{22} A_{22} - E_{22}^H U_{22}^H U_{22} E_{22} = -B_{22}^H B_{22} - yy^H\,. \qquad (42)$$

Indeed, defining, for convenience,

$$a := u_{11} a_{12}^H + A_{22}^H u_{12}^H\,, \quad e := u_{11} e_{12}^H + E_{22}^H u_{12}^H = m_2 b_{12}^H + m_1 a\,, \qquad (43)$$

where the last equality derives from (35), it follows that

$$\begin{aligned} yy^H &= [b_{12}^H \quad a] M \begin{bmatrix} b_{12} \\ a^H \end{bmatrix} = [b_{12}^H \quad a] \begin{bmatrix} |m_1|^2 & -\bar{m}_1 m_2 \\ -m_1 m_2 & m_2^2 \end{bmatrix} \begin{bmatrix} b_{12} \\ a^H \end{bmatrix} \\ &= |m_1|^2 b_{12}^H b_{12} - \bar{m}_1 m_2 b_{12}^H a^H - m_1 m_2 a b_{12} + m_2^2 a a^H\,. \qquad (44) \end{aligned}$$

But with (43), (36) becomes

$$A_{22}^H U_{22}^H U_{22} A_{22} - E_{22}^H U_{22}^H U_{22} E_{22} = -B_{22}^H B_{22} - b_{12}^H b_{12} - aa^H + ee^H$$

(45)

$$= -B_{22}^H B_{22} - b_{12}^H b_{12} - aa^H + (m_2 b_{12}^H + m_1 a)(m_2 b_{12} + \bar{m}_1 a^H)$$

$$\begin{aligned} = -B_{22}^H B_{22} - b_{12}^H b_{12} - aa^H + m_2^2 b_{12}^H b_{12} + \bar{m}_1 m_2 b_{12}^H a^H + m_1 m_2 a b_{12} \\ + |m_1|^2 a a^H \end{aligned}$$

$$= -B_{22}^H B_{22} - |m_1|^2 b_{12}^H b_{12} + \bar{m}_1 m_2 b_{12}^H a^H + m_1 m_2 a b_{12} - m_2^2 a a^H = -B_{22}^H B_{22} - y y^H .$$

Replacing the triangular factor $\tilde{B}_{22}$ of the QR factorization (28) in the last equality in (45), a reduced Lyapunov equation of order $n-1$ in U_{22} is obtained, and the procedure continues recursively in the same way.

The case op(M)= M^H. Solving the equation (2) with $\mathrm{op}(M) = M^H$ is similar to the case $\mathrm{op}(\cdot) = \cdot$. The formulas can be obtained by taking the conjugate transpose of the relations above. The equation for u_{11} is the same, and the other two equations are

$$(\bar{m}_1 A_{22} - E_{22}) u_{12} = -m_2 b_{12} + u_{11}(e_{12} - \bar{m}_1 a_{12}) , \qquad (46)$$

$$A_{22} U_{22} U_{22}^H A_{22}^H - E_{22} U_{22} U_{22}^H E_{22}^H = -B_{22} B_{22}^H - y y^H , \qquad (47)$$

with $y := [\, b_{12} \;\; u_{11} a_{12} + A_{22} u_{12} \,] C$. An RQ factorization of the matrix $[\, B_{22} \;\; y \,]$ is used, as in (32).

5. Numerical issues

If the Lyapunov equation is unreduced, the QZ algorithm is first used, and the equation is transformed to the reduced form (10) or (11). Otherwise, the reduction step is optionally skipped, but the solver can accept the matrices Q and Z on input and apply them to B and to back transform the solution of the reduced equation, $\tilde{X}$. Other options specify the $\mathrm{op}(\cdot)$ operator or the type of equation as continuous- or discrete-time. Using these options is useful, for instance, to compute the controllability and observability Gramians of a linear dynamical system (3). Indeed, the first call of the solver could reduce the matrix pencil $A - \lambda E$ to GSF and return the matrices $\tilde{A}$, $\tilde{E}$, Q, Z, as well as the solution of one of the equations in (4) (or (5)); the second call can use $\tilde{A}$, $\tilde{E}$, Q, and Z, and compute the solution of the second equation. In this way the most time consuming step in the solution process, the reduction to GSF, is skipped for the second equation. A quick test is made to detect identity matrices Q and/or Z and avoid their use in multiplications. The stability condition is easily checked out using the diagonal elements of the GSF, and an error indicator is returned if that condition fails.

For maximum efficiency, the computation of $\hat{B}$ in (7) and (8) is performed using BLAS 3 gemm operations, using as large blocks of columns or rows as possible, depending on the available workspace size. An optimal workspace size can be returned using a special call of the solver with the size set to -1. Then, another call with the obtained size will compute the solution. But the solver can be used with a minimum workspace size of max(1, $3n-3$, $2n$), for $n \geq 0$. The computation of the right hand sides in (12) involves a product of an upper triangular matrix and another (unitary) matrix. While BLAS Library [12] includes a subroutine for such products (trmm), the result is overwritten on the general matrix. For solving Lyapunov equations this is unsuitable, since Q and Z in (12) should be returned by the solver. For this reason, a new routine has been developed which overwrites $\tilde{U}$, possibly without additional workspace. Block-row or block-column operations are performed with block sizes as large as the available workspace allows.

The computation of u_{12}^H in (19) or (35) requires the solution of a triangular linear system of equations with coefficient matrix $A_{22}^H + m_1 E_{22}^H$, or $m_1 A_{22}^H - E_{22}^H$. These matrices must be evaluated, but submatrices of A_{22} and E_{22} are needed in the subsequent computations. It is possible to overwrite, e.g., A_{22}^H by $A_{22}^H + m_1 E_{22}^H$ (or $m_1 A_{22}^H - E_{22}^H$), and restore A_{22} after finding the solution u_{12}^H, by reversing the operations (such as $A_{22}^H := A_{22}^H - m_1 E_{22}^H$). But the chosen technique is more efficient, and takes into account that the upper triangular matrices A and E are either given or returned by the QZ algorithm. Specifically, the strictly lower triangular part of E is overwritten by the conjugate transpose of the strictly upper part, before starting the recursion for solving a reduced equation. Moreover, the diagonal elements of A are saved in the workspace. Then, at each iteration of the recursion, the lower triangular part of the current A_{22} is similarly overwritten by the conjugate transpose of its upper triangular part, and then it is updated to account for the contribution of E_{22}^H and m_1. This updated lower triangular part is used for finding the current u_{12}^H. Then, the diagonal elements of the current A_{22} are restored. Note that this technique preserves the upper triangles of A_{22} and E_{22} and needs no additional computations. Moreover, taking conjugate transposes is anyway needed.

To obtain the factor C of the matrix M in (39), a LAPACK routine, ZSTEIN, is called. This routine can use selected eigenvalues of a tridiagonal symmetric matrix to compute the associated eigenvectors by inverse iteration. For the case of discrete-time generalized Lyapunov equation, the eigenvector corresponding to the unit eigenvalue is needed, but the matrix M is Hermitian. It can be transformed to a similar real tridiagonal symmetric matrix. Indeed, the diagonal elements are preserved, and the off-diagonal elements are set to the modulus of m_{21} (or m_{12}). In this way, both the sum and product of the eigenvalues for the two matrices are the same.

6. Numerical results

An extensive testing has been performed to evaluate the new solver. Some real case examples from [2], [3] have been used to verify the correctness of the delivered results. Tests with randomly generated matrices (from a uniform distribution) have also been performed, and the normalized residuals have been analyzed. For a performance investigation, examples from the COMPl$_e$ib collection [13] have been used. The collection contains 124 standard continuous-time systems, but with variations, a total of 168 systems can be defined. Note that some COMPl$_e$ib examples were derived from systems with general, but nonsingular matrix E, by multiplying the matrices in the state equation in (3) by E^{-1} from the left. These examples have been modified for this paper in order to be obtain stable generalized complex systems and Lyapunov equations.

The computations have been performed in double precision on an Intel Core i7-3820QM portable computer (2.7 GHz, 16 GB RAM), under Windows 7 Professional (Service Pack 1) operating system (64 bit), with Intel Visual Fortran Composer XE 2015, and MATLAB 8.6.0.267246 (R2015b). An executable MEX-file has been built using the new solver, SLICOT routines and MATLAB-provided optimized LAPACK and BLAS routines.

A collection of 33 generalized systems has been derived from the COMPl$_e$ib examples for which the matrix E was available, usually in binary mat files (for 31 examples). For all these examples, E is nonsingular and its inverse has been used to generate the systems in the COMPl$_e$ib collection. Among these, 8 examples (HF2Di, with i = 1, 2, ..., 8) have orders greater than or equal to 2025. Only one example, TL, which describes a transmission line, has a condition number for E larger than 4, namely, $7.7579 \cdot 10^6$. The Hankel singular values for the original TL example, computed using the singular values of the matrix $R_o R_c$, as described in Section 1, are represented in the bar graph of Fig.1, using a logarithmic scale for the ordinate. Only the *significant* singular values of $R_o R_c$ are retained, that is, the largest rank($R_o R_c$) ones are displayed.

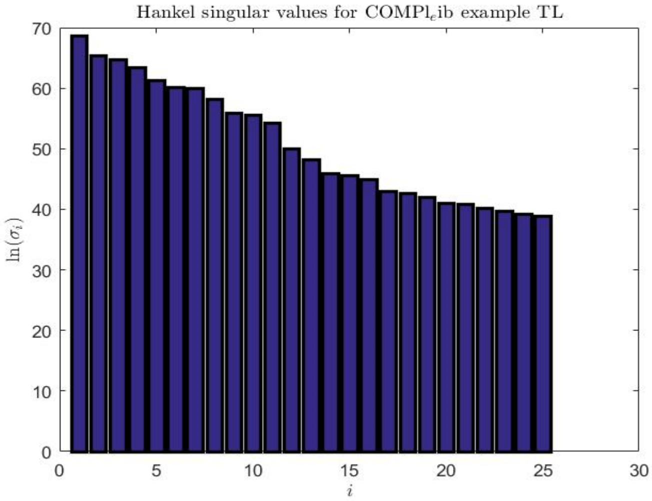

Fig. 1. Significant Hankel singular values of the example TL from the COMPl$_e$ib collection.

A modified TL system with complex matrices has been then obtained using the procedure described below. The significant Hankel singular values for this system are shown in Fig. 2.

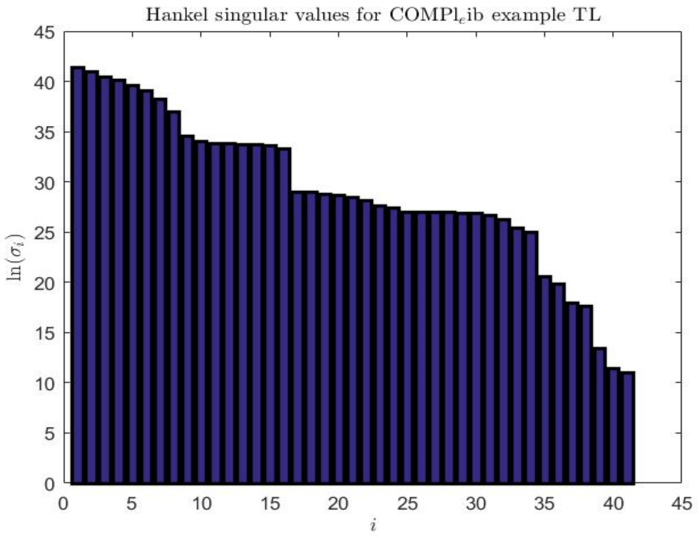

Fig. 2. Significant Hankel singular values of the modified example TL from the COMPl$_e$ib collection.

The other COMPl$_e$ib examples with given matrix E belong to the group of two-dimensional (2D) heat flow models, which arise in the design of static output feedback control laws. The models have been obtained using a discretization algorithm which often produced large scale finite dimensional approximations of the original infinite dimensional problems (examples HF2D1−HF2D8, and their variations). Other examples (HF2D10−HF2D17) are actually their corresponding highly reduced order approximations computed using the *proper orthogonal decomposition* approach.

Procedure to generate stable complex systems. Five of the investigated examples, TL, HF2D3, HF2D4, HF2D12, and HF2D13, are stable. Therefore, these examples have been modified to obtain complex counterparts. The other examples should also be made stable. The procedure to generate such examples is described for the COMPl$_e$ib example HF2D1_M316. This example has $n = 316$, $m = 2$, and $p = 3$. The state matrix is pentadiagonal, specifically tridiagonal plus two other off-diagonals starting at row and column 20, and reducing the distance to the diagonal to 14 at the row/column 198. The matrix E is diagonal with positive elements. There is an unstable eigenvalue, λ_{29} =0.48523 (retaining five significant digits). To obtain a stable system, it is possible to use the real generalized Schur form of the matrix pencil $A - \lambda E$,

$$\tilde{A} = Q^T AZ, \quad \tilde{E} = Q^T EZ \tag{48}$$

with $\tilde{E}$ upper triangular with positive diagonal elements. The positive eigenvalue of $\tilde{A} - \boldsymbol{\lambda}\tilde{E}$ is again λ_{29}, and therefore, the system can be modified, e.g., by changing the sign of $\tilde{a}_{29,29}$. (In control engineering, stability can be enforced, for stabilizing systems, using for instance stabilization or pole assignment algorithms.) But with this modification the system will remain real. In order to obtain a complex system, a feature of the QZ algorithm, qz, included in MATLAB, is used. Specifically, by default, this algorithm returns a complex GSF (6) for real matrices having complex conjugate eigenvalues. (The real GSF is obtained if the option 'real' is specified as an input argument of the qz command.) Since the matrices $\tilde{A}$ and $\tilde{E}$, returned by qz, are upper triangular and $\tilde{E}$ has positive diagonal elements, a stable complex system is obtained by changing the signs of the negative diagonal elements in $\tilde{A}$, e.g., of $\tilde{a}_{29,29}$, for the considered example. The significant Hankel singular values of the modified system are shown in the bar graph of Fig. 3. Fig. 4 shows the results for the larger system HF2D1, of order 3796, modified similarly.

However, 18 COMPl$_e$ib examples, HF2D5_M289, HF2D5_M529, HF2D6_M289, HF2D6_M529, HF2D7, HF2D8, HF2D10, HF2D11, HF2D12, HF2D13, HF2D14, HF2D15, HF2D16, HF2D17, HF2D_IS1, HF2D_IS2_M529, HF2D_IS5, and HF2D_IS6 have real eigenvalues only. For these systems, the qz function returns real matrices $\tilde{A}$ and $\tilde{E}$. To get a complex dynamic system, the matrix $\tilde{A}$ has been modified by replacing $\tilde{a}_{j,j}$ by $\iota\left|\boldsymbol{\lambda}_j\right| \varepsilon^{1/2} - \tilde{a}_{j,j}$, where ι denotes the purely imaginary unit, j is the index of the real eigenvalue with the minimum modulus, and ε is the machine accuracy, $\varepsilon \approx 2.22 \cdot 10^{-16}$.

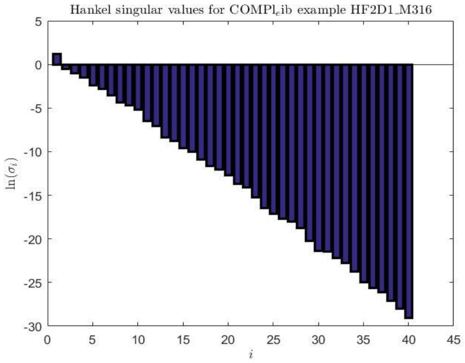

Fig. 3. Significant Hankel singular values of the modified example HF2D1_M316 from the COMPl$_e$ib collection.

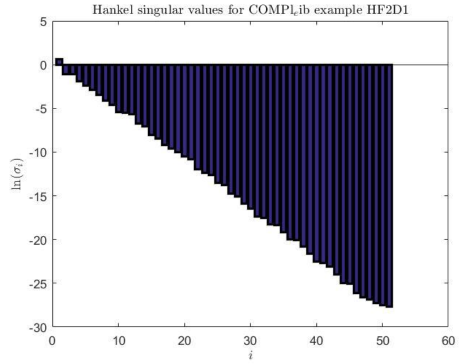

Fig. 4 Significant Hankel singular values of the modified example HF2D1 from the COMPl$_e$ib collection.

Table 1 shows the first two Hankel singular values $\sigma_{1,2}$ for all 33 COMPl$_e$ib examples, modified as described above. The number j is the index for which λ_j is unstable; r is the number of significant Hankel singular values, that is, rank$(R_o R_c)$. A hyphen is used when there are no unstable eigenvalues; this happened for five examples mentioned above. There is a single originally unstable eigenvalue for the remaining examples.

Table 1. The largest two Hankel singular values for modified COMPl$_e$ib examples of generalized systems.

Example	n	m	p	$\sigma_{1,2}$		j	r
TL	256	2	2	1.018e+18	6.630e+17	-	41
HF2D1	3796	2	3	1.8941e+0	3.5536e-1	328	51
HF2D1_M316	316	2	3	3.5506e+0	6.2466e-1	29	40
HF2D1_M541	541	2	3	1.7852e+0	4.6071e-1	47	44
HF2D2	3796	2	3	1.0307e+0	1.3844e-1	328	47
HF2D2_M316	316	2	3	5.5555e-1	1.6910e-1	30	40
HF2D2_M541	541	2	3	5.6463e-1	1.5720e-1	47	42
HF2D3	4489	2	4	6.2163e-1	2.8049e-1	-	50
HF2D4	2025	2	4	1.7699e+1	3.1309e+0	-	44
HF2D5	4489	2	4	2.9820e+1	1.8711e+1	1	53
HF2D5_M289	289	2	4	3.0656e+0	2.1297e+0	4	41
HF2D5_M529	529	2	4	6.3719e+0	4.2424e+0	4	43
HF2D6	2025	2	4	8.0719e+0	5.3872e+0	9	46
HF2D6_M289	289	2	4	1.3787e+0	7.6935e-1	1	37
HF2D6_M529	529	2	4	4.1366e+0	2.6703e+0	1	41
HF2D7	4489	2	4	4.2784e+1	1.7350e+1	2	51
HF2D8	2025	2	4	8.9382e+0	3.6434e+0	9	46
HF2D10	5	2	3	1.9855e+0	3.8523e-1	3	5
HF2D11	5	2	3	1.1360e+0	1.4687e-1	3	5
HF2D12	5	2	4	5.9841e+0	8.2993e-1	-	5
HF2D13	5	2	4	8.1003e+1	4.1171e+0	-	5
HF2D14	5	2	4	2.8355e+1	1.0403e+1	2	5
HF2D15	5	2	4	7.9188e+0	3.4392e+0	2	5
HF2D16	5	2	4	4.3374e+1	1.5587e+1	2	5
HF2D17	5	2	4	8.9934e+0	3.9228e+0	2	5
HF2D_IS1	4489	2	4	3.8087e+0	1.2708e+0	4	51
HF2D_IS1_M361	361	2	4	2.9901e+0	1.3285e+0	1	41
HF2D_IS1_M529	529	2	4	1.0183e+0	8.3036e-1	4	43
HF2D_IS2	4489	2	4	3.4372e-1	2.2463e-1	9	49
HF2D_IS2_M361	361	2	4	5.3589e-1	3.4371e-1	1	39
HF2D_IS2_M529	529	2	4	4.6915e-1	3.0184e-1	1	42
HF2D_IS5	5	2	4	1.6869e+0	3.3007e-1	2	5
HF2D_IS6	5	2	4	3.2556e-1	6.2128e-2	2	5

Table 2 shows the CPU execution times in seconds needed to compute the results. Specifically, the CPU times spent by the QZ algorithm, computation of the eigenvalues, transformation of the right hand sides and solution of the two related reduced Lyapunov equations (10), and computation of the singular values (including evaluation of the product $R_c R_o$) are shown in separate columns. No record is included when all these four timing values are zero. The maximum total time is of 57.01 minutes, for example HF2D5, with $n = 4489$, $m = 2$, and $p = 4$. The most time consuming step is the reduction to complex GSF using qz command. The ratios between the timing values needed by qz and the solution of the two reduced equations vary in the range (1.29, 2.58) and the mean value of these ratios is 1.71. Similarly, the ratios between the timing values needed by qz and the computation of the singular values vary in the range (5.22, 20) and the mean value is 11.81. The computation of the eigenvalues needs a negligible CPU time. The total CPU time for each system of order at most 541 is less than 3 seconds. Similarly, the computational problem for each system of order 2025 has been solved in less than 283 seconds.

Table 2. CPU time in seconds for modified COMPl$_e$ib examples of generalized systems.

Example	Timing (seconds)			
	QZ	λ	Solution	SVD
TL	0.18	0.00	0.10	0.01
HF2D1	916.91	0.00	708.08	165.67
HF2D1_M316	0.34	0.00	0.16	0.02
HF2D1_M541	1.53	0.00	0.98	0.10
HF2D2	915.35	0.00	699.00	175.17
HF2D2_M316	0.36	0.00	0.14	0.02
HF2D2_M541	1.51	0.00	0.97	0.09
HF2D3	1876.68	0.00	1191.23	294.67
HF2D4	158.70	0.00	94.61	28.55
HF2D5	1906.48	0.00	1226.94	287.45
HF2D5_M289	0.26	0.00	0.12	0.02
HF2D5_M529	1.59	0.00	1.01	0.09
HF2D6	161.25	0.00	94.48	27.26
HF2D6_ M289	0.25	0.00	0.15	0.02
HF2D6_ M529	1.52	0.00	1.00	0.09
HF2D7	1913.63	0.41	1131.39	299.06
HF2D8	161.26	0.08	91.25	27.56
HF2D_IS1	1695.43	0.41	1112.78	269.86
HF2D_IS1_ M361	0.48	0.00	0.25	0.03
HF2D_IS1_ M529	1.60	0.00	0.92	0.08
HF2D_IS2	1832.53	0.00	1229.07	295.56
HF2D_IS2_ M361	0.45	0.00	0.23	0.03
HF2D_IS2_ M529	1.54	0.00	0.95	0.10

7. Conclusions

A numerically attractive solution for stable generalized complex Lyapunov equations for both continuous- and discrete-time case is presented. Two equations with the matrices A and E, and A^H and E^H, respectively, can be solved using a single computation of the generalized Schur form. This is useful, for instance, when finding the Hankel singular values of linear generalized dynamical systems. The basic computational formulas are derived in a systematic manner, and the related numerical issues are highlighted. The numerical results for a significant set of large examples based on the COMPl$_e$ib collection illustrate the solver performance. The proposed solver could be a useful addition to the state-of-the-art numerical software packages and environments.

References

[1] Sima V., *Comparative performance evaluation of an accuracy-enhancing Lyapunov solver,* Information, Special Issue ICSTCC 2018: Advances in Control and Computers, **10**, (6 215), 2019.

[2] Hammarling S. J., *Numerical solution of the stable, non-negative definite Lyapunov equation,* IMA J. Numer. Anal., **2**, 3, 1982, p. 303–323.

[3] Penzl T., *Numerical solution of generalized Lyapunov equations*, Advances in Comp. Math., **8**, 1998, p. 33–48.

[4] Bartels R. H., Stewart G. W., *Algorithm 432: Solution of the matrix equation AX + XB = C*, Comm. ACM, **15**, 9, 1972, p. 820–826.

[5] Benner P., Mehrmann V., Sima V., Van Huffel S., Varga A., *SLICOT — A subroutine library in systems and control theory*, Applied and Computational Control, Signals, and Circuits, Datta B. N., Ed., Birkhäuser, Boston, MA, **1**, 10, 1999, p. 499–539.

[6] Van Huffel S., Sima V., Varga A., Hammarling S., Delebecque F., *High-performance numerical software for control*, IEEE Control Syst. Mag., **24**, 1, 2004, p. 60–76.

[7] MathWorks®, *Control System Toolbox*™, The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA, 01760–2098, 2015.

[8] Sima V., *Algorithms for Linear-Quadratic Optimization*, Pure and Applied Mathematics: A Series of Monographs and Textbooks, Taft E. J., Nashed Z. (Series Eds.), vol. 200, Marcel Dekker, Inc., New York, 1996.

[9] Simoncini V., *Computational methods for linear matrix equations*, SIAM Rev., **58**, 3, 2016, p. 377–441.

[10] Anderson E., Bai Z., Bischof C., Blackford S., Demmel J., Dongarra J., Du Croz J., Greenbaum A., Hammarling S., McKenney A., Sorensen D., *LAPACK Users' Guide: Third Edition*, ser. Software・Environments・Tools, SIAM, Philadelphia, PA, 1999.

[11] Golub G. H., Van Loan C. F., *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MA, 1996.

[12] Dongarra J. J., Du Croz J., Duff I. S., Hammarling S., *Algorithm 679: A set of Level 3 Basic Linear Algebra Subprograms*, ACM Trans. Math. Softw., **16**, 1, 1990, p. 1–17, 18–28.

[13] Leibfritz F., Lipinski W., *Description of the benchmark examples in COMPl$_e$ib,* Technical report, Department of Mathematics, University of Trier, D–54286 Trier, Germany, 2003.