Solving Stable Generalized Lyapunov Equations for Hankel Singular Values Computation

Vasile Sima[a]
Technical Sciences Academy of Romania, Bucharest, Romania

Keywords: Balancing, Hankel Singular Values, Lyapunov Equation, Numerical Methods, Stability.

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. This application uses the controllability and observability Gramians of the system, given as the solutions of a pair of generalized Lyapunov equations. The left hand side of each of these equations follows from the other one by applying the (conjugate) transposition operator. If the system is stable, the solutions of both equations are non-negative definite, hence they can be obtained in a factorized form. But these theoretical results may not hold in numerical computations if the symmetry and non-negative definiteness are not preserved by a solver. The paper summarizes new related numerical algorithms for complex continuous- and discrete-time generalized systems. Such solvers are not yet available in the SLICOT Library or MATLAB. The developed solvers address the essential practical issues of reliability, accuracy, and efficiency.

1 INTRODUCTION

Stable generalized complex Lyapunov equations with unknown $X = X^H$ can be written as

$$\operatorname{o}(A)^H X \operatorname{o}(E) + \operatorname{o}(E)^H X \operatorname{o}(A) = -\operatorname{o}(B)^H \operatorname{o}(B), \quad (1)$$
$$\operatorname{o}(A)^H X \operatorname{o}(A) - \operatorname{o}(E)^H X \operatorname{o}(E) = -\operatorname{o}(B)^H \operatorname{o}(B), \quad (2)$$

in the continuous- and discrete-time case, respectively, where $A,\, E \in \mathbf{C}^{n\times n}$, $\operatorname{o}(B) \in \mathbf{C}^{m\times n}$, the operator $\operatorname{o}(M)$ is either M or M^H for any matrix M, and the 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 $\Lambda(E^{-1}A) \in \mathbf{C}_-$, where $\mathbf{C}_-$ is the open left half, or open unit disk of the complex plane for (1), or (2), respectively, and $\Lambda(M)$ is the spectrum of the matrix M. (See, e.g., (Sima, 2019) and the references therein.) Equivalently, the matrix pencil $A - \lambda E$ has only stable eigenvalues in the continuous- or discrete-time sense.

These stable equations have a unique positive-semidefinite solution X, $X \geq 0$, since $\operatorname{o}(B)^H \operatorname{o}(B) \geq 0$. Then, X can be written in a factorized form, $X = \operatorname{o}(U)^H \operatorname{o}(U)$, where U is the Cholesky factor of X, if $X > 0$. Note that any matrix expressed as $\operatorname{o}(B)^H \operatorname{o}(B)$ has real non-negative diagonal elements, given by $b_j^H b_j = \|b_j\|^2$, where b_j is the j-th column of $\operatorname{o}(B)$, and $\|x\|$ is the Euclidean norm of x. For an identity matrix E, $E = I_n$, and $\operatorname{o}(M) = M$, the *standard* stable Lyapunov equations are obtained, dealt with in (Hammarling, 1982). Algorithms for solving real generalized Lyapunov equations have been proposed in (Penzl, 1998), and extensions for singular matrix E are dealt with in (Stykel, 2002; Stykel, 2004). These algorithms belong to the class of transformation methods (Bartels and Stewart, 1972). Solvers implementing some of these algorithms are available, e.g., in the SLICOT Library (Benner et al., 1999; Van Huffel et al., 2004) (https://github.com/SLICOT/SLICOT-Reference) and in MATLAB Control System Toolbox (MathWorks®, 2015).

Standard and generalized Lyapunov equations often appear in systems theory, stability investigation, analysis and design of control systems, signal processing, statistics, and other domains. Few textbooks can be cited in this short paper, e.g., (Bini et al.,

[a] https://orcid.org/0000-0003-1445-345X

Sima, V.
Solving Stable Generalized Lyapunov Equations for Hankel Singular Values Computation.
DOI: 10.5220/0011259900003271
In *Proceedings of the 19th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2022)*, pages 130-137
ISBN: 978-989-758-585-2; ISSN: 2184-2809

2012; Lancaster and Rodman, 1995; Mehrmann, 1991; Varga, 2017), but they include many references. Many algorithmic and computational details for Sylvester and standard Lyapunov equations are given, e.g., in (Sima, 1996). General linear matrix equations are dealt with in (Simoncini, 2016).

This paper extends the results in (Penzl, 1998) to complex equations. New algorithms and associated implementations, based on transformation method, are investigated. The computational details are carefully considered to obtain accurate and reliable solutions. Important issues in this endeavour are presented. Numerical results for a large set of difficult examples illustrate the performance of the new solver.

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 $\mathrm{o}(\cdot)$. Specifically, consider a generalized system,

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

where $x(t) \in \mathbf{C}^n$, $B \in \mathbf{C}^{n\times m}$, $C \in \mathbf{C}^{p\times n}$, and $\lambda(x(t))$ is the differential operator, $\mathrm{d}x(t)/\mathrm{d}t$, or the advance difference operator, $\lambda(x(t)) = x(t+1)$, 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,

$$\left.\begin{aligned} APE^H + EPA^H &= -BB^H, \\ A^HQE + E^HQA &= -C^HC, \end{aligned}\right\} \tag{4}$$

$$\left.\begin{aligned} APA^H - EPE^H &= -BB^H, \\ A^HQA - E^HQE &= -C^HC, \end{aligned}\right\} \tag{5}$$

for continuous- and discrete-time systems, 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, but B and C are directly used. Then, the Hankel singular values of the system are the singular values of the product R_oR_c, numerically guaranteed to be real non-negative.

Solving Lyapunov equations and finding Hankel singular values are essential ingredients for model order reduction, which is a topic of intense research (Benner and Werner, 2020; Cao et al., 2020; Yang et al., 2020).

2 COMPUTATIONAL STEPS

Reduction to Generalized Schur Form. For general, unstructured 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 (Anderson et al., 1999; Golub and Van Loan, 2013). In the complex case, the QZ algorithm returns the reduced matrices, $\widetilde{A}$ and $\widetilde{E}$, and the unitary matrices, $Q, Z \in \mathbf{C}^{n\times n}$, $Q^HQ = QQ^H = I_n$, $Z^HZ = ZZ^H = I_n$, so that

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

where $\widetilde{A}$ and $\widetilde{E}$ are upper triangular, and the diagonal elements of $\widetilde{E}$ are real non-negative. The eigenvalues of the pencil $\widetilde{A} - \lambda\widetilde{E}$ are given by $\lambda_i = \widetilde{a}_{ii}/\widetilde{e}_{ii}$.

Transformation of the Right Hand Side. Since Q and Z are unitary, then premultiplying (1) and (2) by Z^H and postmultiplying them by Z, the following equations are obtained if $\mathrm{o}(M) = M$:

$$\mathrm{o}(\widetilde{A})^H\widetilde{X}\,\mathrm{o}(\widetilde{E}) + \mathrm{o}(\widetilde{E})^H\widetilde{X}\,\mathrm{o}(\widetilde{A}) = -\,\mathrm{o}(\widehat{B})^H\,\mathrm{o}(\widehat{B}), \tag{7}$$

$$\mathrm{o}(\widetilde{A})^H\widetilde{X}\,\mathrm{o}(\widetilde{A}) - \mathrm{o}(\widetilde{E})^H\widetilde{X}\,\mathrm{o}(\widetilde{E}) = -\,\mathrm{o}(\widehat{B})^H\,\mathrm{o}(\widehat{B}), \tag{8}$$

where $\widetilde{X} := Q^HXQ$ and $\widehat{B} := BZ$. Similarly, if $\mathrm{o}(M) = M^H$, premultiplying (1) and (2) by Q^H and postmultiplying by Q, the equations (7) and (8) are obtained again, with $\widetilde{X} := Z^HXZ$ and $\widehat{B} := B^HQ$. The matrix $\widehat{B}$ is not used directly, but after a transformation into a standardized form. Specifically, $\widehat{B}$ is triangularized using QR or RQ factorizations if $\mathrm{o}(M) = M$ or $\mathrm{o}(M) = M^H$, respectively,

$$\left.\begin{aligned} \bar{Q}_B\widetilde{B} &= \begin{bmatrix} \widehat{B} \\ 0 \end{bmatrix}, \quad \text{if } m < n, \\ Q_B\begin{bmatrix} \widetilde{B} \\ 0 \end{bmatrix} &= \widehat{B}, \quad \text{if } m \geq n, \end{aligned}\right\} \tag{9}$$

$$\left.\begin{aligned} \begin{bmatrix} \widetilde{B} & 0 \end{bmatrix}\bar{Q}_B &= \begin{bmatrix} \widehat{B} & 0 \end{bmatrix}, \quad \text{if } m < n, \\ \begin{bmatrix} \widetilde{B} & 0 \end{bmatrix}Q_B &= \widehat{B}, \qquad \text{if } m \geq n, \end{aligned}\right\} \tag{10}$$

$$\bar{Q}_B := \begin{bmatrix} Q_B & 0 \\ 0 & I_{n-m} \end{bmatrix},$$

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

$$\mathrm{o}(\widetilde{A})^H\widetilde{X}\,\mathrm{o}(\widetilde{E}) + \mathrm{o}(\widetilde{E})^H\widetilde{X}\,\mathrm{o}(\widetilde{A}) = -\,\mathrm{o}(\widetilde{B})^H\,\mathrm{o}(\widetilde{B}), \quad (11)$$
$$\mathrm{o}(\widetilde{A})^H\widetilde{X}\,\mathrm{o}(\widetilde{A}) - \mathrm{o}(\widetilde{E})^H\widetilde{X}\,\mathrm{o}(\widetilde{E}) = -\,\mathrm{o}(\widetilde{B})^H\,\mathrm{o}(\widetilde{B}). \quad (12)$$

Solution of the Reduced Equation. The solution of the reduced equations (11) and (12) is discussed in the next two sections. The result is obtained in a factorized form, $\widetilde{X} = \mathrm{o}(\widetilde{U})^H\,\mathrm{o}(\widetilde{U})$, 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 $\mathrm{o}(M) = M$ or $\mathrm{o}(M) = M^H$ is obtained using the QR or RQ factorization, respectively, as follows,

$$\begin{aligned} Q_U U &= \widetilde{U}Q^H, &\quad &\text{if } \mathrm{o}(M) = M,\\ UQ_U &= Z\widetilde{U}, &\quad &\text{if } \mathrm{o}(M) = M^H, \end{aligned} \quad (13)$$

where Q_U is unitary and U is upper triangular with real diagonal elements. If $u_{ii} < 0$, the i-th row or column, respectively, is scaled by -1.

3 SOLVING REDUCED CONTINUOUS-TIME EQUATIONS

The solution of the reduced equations (11) 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 $\mathbf{o}(M) = M$. Consider first the case $\mathrm{o}(M) = M$ and the following matrix partition

$$A = \begin{bmatrix} a_{11} & a_{12} \\ 0 & A_{22} \end{bmatrix}, \quad E = \begin{bmatrix} e_{11} & e_{12} \\ 0 & E_{22} \end{bmatrix},$$
$$B = \begin{bmatrix} b_{11} & b_{12} \\ 0 & B_{22} \end{bmatrix}, \quad U = \begin{bmatrix} u_{11} & u_{12} \\ 0 & U_{22} \end{bmatrix}, \quad (14)$$

where $a_{11} \in \mathbf{C}$, $e_{11} > 0$, $u_{11}, b_{11} \geq 0$, a_{12}, e_{12}, u_{12}, $b_{12} \in \mathbf{C}^{1\times(n-1)}$, and A_{22}, E_{22}, U_{22}, $B_{22} \in \mathbf{C}^{(n-1)\times(n-1)}$. If the matrices in (14) are used in (11), its solution can be found recursively. Specifically, evaluating the (1,1), (2,1), and (2,2) block elements of the resulting left and right hand side expressions, (11) can be decomposed as

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

$$\begin{aligned} &u_{11}(e_{11}A_{22}^H + a_{11}E_{22}^H)u_{12}^H = \\ &\quad - b_{11}b_{12}^H - u_{11}^2(e_{11}a_{12}^H + a_{11}e_{12}^H), \end{aligned} \quad (16)$$

$$\begin{aligned} &A_{22}^H U_{22}^H U_{22}E_{22} + E_{22}^H U_{22}^H U_{22}A_{22} = \\ &\quad - b_{12}^H b_{12} - ae^H - ea^H - B_{22}^H B_{22}, \end{aligned} \quad (17)$$

where $\bar{x}$ denotes the conjugate of x, and

$$a := u_{11}a_{12}^H + A_{22}^H u_{12}^H, \quad e := u_{11}e_{12}^H + E_{22}^H u_{12}^H. \quad (18)$$

These equations can be solved successively for u_{11}, u_{12}^H, and U_{22}, as shown below. Indeed, (15), (16), and (17) can be rewritten as

$$2\Re(a_{11})e_{11}u_{11}^2 = -b_{11}^2, \quad (19)$$
$$(A_{22}^H + m_1E_{22}^H)u_{12}^H = -m_2b_{12}^H - u_{11}(a_{12}^H + m_1e_{12}^H), \quad (20)$$
$$A_{22}^H U_{22}^H U_{22}E_{22} + E_{22}^H U_{22}^H U_{22}A_{22} = -B_{22}^H B_{22} - yy^H, \quad (21)$$

respectively, where $\Re(\alpha)$ denotes the real part of a complex number α, and

$$m_1 := a_{11}/e_{11}, \quad m_2 := b_{11}/e_{11}/u_{11}, \quad (22)$$
$$y := b_{12}^H - m_2e. \quad (23)$$

From (19), it follows that

$$u_{11} = b_{11}/\sqrt{-2\Re(a_{11})e_{11}}. \quad (24)$$

Note that $u_{11} \in \mathbf{R}$, since $b_{11} \geq 0$, $e_{11} > 0$, and, by the stability assumption, $a_{11} \in \mathbf{C}_-$. Moreover, $u_{11} > 0$, if $b_{11} > 0$, and $u_{11} = 0$, if $b_{11} = 0$. Equation (20) follows by dividing (16) by $u_{11}e_{11}$, if $u_{11} \neq 0$, and using (22). By a continuity argument, (20) holds also for $u_{11} = 0$; moreover, note that m_2 can be rewritten as $m_2 = \sqrt{-2\Re(a_{11})e_{11}}/e_{11}$, which is defined also for $u_{11} = 0$. Hence,

$$m_2^2 = -2\Re(a_{11})/e_{11} = -(m_1 + \bar{m}_1). \quad (25)$$

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., (Golub and Van Loan, 2013), with care to avoid overflow.

Equation (21) is obtained from (17), noting that, with (23),

$$\begin{aligned} yy^H &= \left[b_{12}^H - m_2e\right]\left[b_{12} - m_2e^H\right] \\ &= b_{12}^H b_{12} - m_2b_{12}^H e^H - m_2eb_{12} + m_2^2 ee^H. \end{aligned} \quad (26)$$

But from (20), it follows that

$$a = -m_2b_{12}^H - m_1e, \quad (27)$$

so that,

$$ae^H = -m_2 b_{12}^H e^H - m_1 ee^H. \tag{28}$$

Adding the conjugate transpose of (28), and using (17) and (26), it follows that (21) holds. If $\widetilde{B}_{22}$ is the square triangular factor of the QR factorization

$$\widetilde{Q}\begin{bmatrix} \widetilde{B}_{22} \\ 0 \end{bmatrix} = \begin{bmatrix} B_{22} \\ y^H \end{bmatrix}, \tag{29}$$

then the right hand side from (21) becomes $-\widetilde{B}_{22}^H\widetilde{B}_{22}$. Consequently, the corresponding equation has the same structure as the original equation (11), but its order is initially $n-1$. Therefore, the same solution technique can be used recursively. The factorization in (29) is computed using $n-1$ Givens rotations (Golub and Van Loan, 2013).

The Case o$(M) = M^H$. Solving (11) with o$(M) = M^H$ uses a different partition, namely,

$$A = \begin{bmatrix} A_{22} & a_{12} \\ 0 & a_{11} \end{bmatrix}, \quad E = \begin{bmatrix} E_{22} & e_{12} \\ 0 & e_{11} \end{bmatrix}, \tag{30}$$

and similarly for B and U, where $a_{11} \in \mathbf{C}$, $e_{11} > 0$, $u_{11}, b_{11} \geq 0$, a_{12}, e_{12}, u_{12}, $b_{12} \in \mathbf{C}^{(n-1)\times 1}$, and A_{22}, E_{22}, U_{22}, $B_{22} \in \mathbf{C}^{(n-1)\times(n-1)}$.

With (30), the formulas for the o$(M) = M^H$ case can essentially be obtained by taking the conjugate transpose of the matrices and vectors used for the o$(M) = M$ case. The equation for u_{11} is the same, and the other two equations are

$$(A_{22} + \bar{m}_1 E_{22})u_{12} = -m_2 b_{12} - u_{11}(a_{12} + \bar{m}_1 e_{12}), \tag{31}$$

$$A_{22}U_{22}U_{22}^H E_{22}^H + E_{22}U_{22}U_{22}^H A_{22}^H = -B_{22}B_{22}^H - yy^H, \tag{32}$$

with $y := b_{12} - m_2(u_{11}e_{12} + E_{22}u_{12})$. Using an RQ factorization of the matrix

$$\begin{bmatrix} \widetilde{B}_{22} & 0 \end{bmatrix}\widetilde{Q} = \begin{bmatrix} B_{22} & y \end{bmatrix}, \tag{33}$$

the right hand side of (32) becomes $-\widetilde{B}_{22}\widetilde{B}_{22}^H$.

4 SOLVING REDUCED DISCRETE-TIME EQUATIONS

The Case o$(M) = M$. In a similar manner to the continuous-time case, the basic equations for the discrete-time case (12) with o$(M) = M$ are as follows

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

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

$$A_{22}^H U_{22}^H U_{22} A_{22} - E_{22}^H U_{22}^H U_{22} E_{22} = - b_{12}^H b_{12} - aa^H + ee^H - B_{22}^H B_{22}, \tag{36}$$

where m_1 and m_2 are defined in (22) and a and e are defined in (18). The solution of (34) is

$$u_{11} = b_{11}/\sqrt{e_{11}^2 - |a_{11}|^2}, \tag{37}$$

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

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

Define also a 2×2 matrix

$$M := I_2 - \begin{bmatrix} m_2 \\ m_1 \end{bmatrix}\begin{bmatrix} m_2 \\ m_1 \end{bmatrix}^H = \begin{bmatrix} |m_1|^2 & -m_2\bar{m}_1 \\ -m_1 m_2 & m_2^2 \end{bmatrix} =: FF^H, \tag{39}$$

where (38) was used and FF^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),

$$\lambda_1 + \lambda_2 = |m_1|^2 + m_2^2 = 1,$$
$$\lambda_1\lambda_2 = |m_1|^2 m_2^2 - |m_1|^2 m_2^2 = 0. \tag{40}$$

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 F of the rank-1 matrix M in (39) can be taken as the eigenvector of M corresponding to the unit eigenvalue. Defining now

$$y := \begin{bmatrix} b_{12}^H & u_{11}a_{12}^H + A_{22}^H u_{12}^H \end{bmatrix} F, \tag{41}$$

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

$$A_{22}^H U_{22}^H U_{22} A_{22} - E_{22}^H U_{22}^H U_{22} E_{22} = -B_{22}^H B_{22} - yy^H. \tag{42}$$

Indeed, from (35) and (18), it follows that

$$e = m_2 b_{12}^H + m_1 a, \tag{43}$$

and therefore, with (41) and (39)

$$\begin{aligned} yy^H &= \begin{bmatrix} b_{12}^H & a \end{bmatrix}\begin{bmatrix} |m_1|^2 & -m_2\bar{m}_1 \\ -m_1 m_2 & m_2^2 \end{bmatrix}\begin{bmatrix} b_{12} \\ a^H \end{bmatrix} \\ &= |m_1|^2 b_{12}^H b_{12} - \bar{m}_1 m_2 b_{12}^H a^H - m_1 m_2 a b_{12} \\ &\quad + m_2^2 aa^H. \end{aligned} \tag{44}$$

The equivalence of (36) to (42) follows, since (43), (44), and (38) imply

$$\begin{aligned} ee^H - b_{12}^H b_{12} - aa^H &= m_2^2 b_{12}^H b_{12} + \bar{m}_1 m_2 b_{12}^H a^H \\ &\quad + m_1 m_2 a b_{12} + |m_1|^2 aa^H - b_{12}^H b_{12} - aa^H \\ &= \bar{m}_1 m_2 b_{12}^H a^H + m_1 m_2 a b_{12} - m_2^2 aa^H \\ &\quad - |m_1|^2 b_{12}^H b_{12} = -yy^H. \end{aligned} \tag{45}$$

Replacing the triangular factor $\widetilde{B}_{22}$ of the QR factorization (29) in (42), a reduced Lyapunov equation of order $n-1$ in U_{22} is obtained, and the procedure continues recursively in the same way.

The Case o$(M) = M^H$. Solving (12) with o$(M) = M^H$ is similar. With a partition as in (30), the equation for u_{11} is the same, and the other two equations are

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

$$A_{22}U_{22}U_{22}^H A_{22}^H - E_{22}U_{22}U_{22}^H E_{22}^H = -B_{22}B_{22}^H - yy^H, \tag{47}$$

with

$$y := \begin{bmatrix} b_{12} & u_{11}a_{12} + A_{22}u_{12} \end{bmatrix} F. \tag{48}$$

5 NUMERICAL ISSUES

If the Lyapunov equation has general matrices A and E, the solver starts by using the QZ algorithm. If $A = \widetilde{A}$, $E = \widetilde{E}$, this step is optionally skipped, but the solver can accept the matrices Q and Z on input and apply them to B and to the solution of the reduced equation, $\widetilde{X}$, for obtaining X. Other solver options specify the o(M) 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 $\widetilde{A}$, $\widetilde{E}$, Q, Z, as well as the solution of one of the equations in (4) (or (5)); the second call can use $\widetilde{A}$, $\widetilde{E}$, Q, and Z, and compute the solution of the second equation. In this way the most time consuming computational step, the reduction to GSF, is skipped for the second equation. The stability condition is easily checked out using the diagonal elements (or 2×2 real diagonal blocks) of the GSF, and an error indicator is returned if that condition fails.

For maximum efficiency, the computation of $\widehat{B}$ in (7) and (8) is performed with 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 value will compute the solution. The computation of the right hand sides in (13) involves a product of an upper triangular matrix and another (unitary) matrix (or in the reverse order, for o$(M) = M^H$). While BLAS Library (Dongarra et al., 1990) includes a subroutine for such products (ZTRMM), the result is overwritten on the general matrix. This is unsuitable for solving Lyapunov equations, since Q or Z in (13) should be returned by the solver. For this reason, a new routine has been developed which overwrites $\widetilde{U}$, possibly without additional workspace, using block-row or block-column operations.

The computation of u_{12}^H or u_{12} in (20), (35), or in (31), (46), requires the solution of a triangular linear system of equations with coefficient matrix $A_{22}^H + m_1 E_{22}^H$, $m_1 A_{22}^H - E_{22}^H$, $A_{22} + \bar{m}_1 E_{22}$, or $\bar{m}_1 A_{22} - E_{22}$, respectively. 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 similarly, for the other expressions), and restore A_{22} after finding the solution u_{12}^H, using $A_{22}^H := A_{22}^H - m_1 E_{22}^H$. But the chosen technique is more efficient. 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.

Scaling is used to avoid overflows when solving the linear algebraic systems in (20), (35), (31), or (46). Specifically, a system $Mx = \sigma b$ is solved instead of $Mx = b$, where $\sigma \in [0, 1]$ is chosen so that the elements of the computed x are representable in a computer. Such a solver is available in the LAPACK package (Anderson et al., 1999); it scales x and σ to avoid overflow or divide-by-zero. Usually, $\sigma = 1$. If M is singular, then $\sigma = 0$, and a non-trivial solution of $Mx = 0$ is obtained. The current value of σ is used by the Lyapunov solver to update the current results, and the final σ value is returned. Actually, the solver computes the solution of an equation (1) or (2) with the right hand side replaced by $-\sigma\, \mathrm{o}(B)^H \mathrm{o}(B)$, where $\sigma \neq 1$ only if it is necessary.

To obtain the factor F 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. The eigenvector corresponding to the unit eigenvalue of M is needed for a discrete-time equation (2). The Hermitian matrix M is transformed to a similar real tridiagonal symmetric matrix, by setting the off-diagonal elements to the modulus of m_{21} (or m_{12}) and preserving the diagonal elements. In this way, the sum and product of the eigenvalues for the two matrices are the same.

Some Lyapunov equations may have one or all matrices A, E, and B with elements of very large or very small magnitude. This may cause numerical difficulties, including inaccurate results or even overflows. Besides the scaling mentioned above, possibly activated when computing u_{12}^H or u_{12}, other two special scaling strategies have been included in the main solver. One strategy scales A, E, and B if the maximum absolute values of their elements are outside a range $[s, b]$, where $s = (\mathrm{sm})^{1/2}/\varepsilon$, $b = 1/s$, sm is the safe minimum, and ε is the machine accuracy. The second strategy scales A, E, and B if the maximum absolute values of the elements of A, E, and B differ too much, or their global minimum (maximum) is too large (small, respectively); specifically, this scaling is performed if $m < Ms$, or $M < s$, or $m > b$, where m and M are the minimum and maximum, respectively, of the maximum absolute values in A, E, and B. Both strategies are effective and ensure the same accuracy of the results, but the second strategy reduces the number of instances when the output scaling factor, σ, is strictly smaller than 1. The implementation first checks out the conditions for the second scaling strategy. Moreover, the scaling factors of E may be set equal to those for A, to preserve stability in the discrete-time case. If used, the special scaling is undone at the end of the computations and the scaling factor σ is updated, if necessary. Often, the value σ returned by the solver is 1, even if the matrices A, E, and/or B have values with magnitude close to the largest representable number. This does not happen for other solvers.

6 NUMERICAL RESULTS

An extensive testing has been performed to evaluate the new solver. The computations have been done in double precision on an Intel Core i7-3820QM portable computer (2.7 GHz, 16 GB RAM). An executable MEX-file has been built using the new solver, SLICOT routines and MATLAB-provided optimized LAPACK and BLAS routines. Tests with randomly generated matrices (from a uniform distribution) have been run, and the normalized residuals have been analyzed.

For a performance investigation, examples from the COMPl$_e$ib collection (Leibfritz and Lipinski, 2003) have been used. The collection contains 124 standard continuous-time systems, but with variations, a total of 168 systems can be defined. Note that 33 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. 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

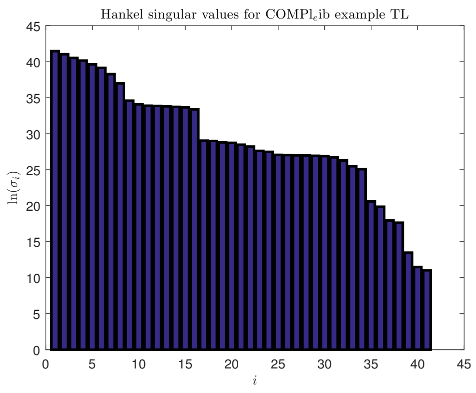

Figure 1: Significant Hankel singular values of the example `TL` from the COMPl$_e$ib collection.

transmission line, has a large condition number for E, namely, $7.7579 \cdot 10^6$. The Hankel singular values for this difficult example, computed using the singular values of the matrix R_oR_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_oR_c are retained, that is, the largest $\mathrm{rank}(R_oR_c)$ ones are displayed.

The other 32 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 had been obtained using a discretization algorithm which often produced large scale finite dimensional approximations of the original infinite dimensional problems (examples `HF2D1` – `HF2D8`, with variations). Other examples (`HF2D10` – `HF2D17`) are their corresponding highly reduced order approximations computed using the *proper orthogonal decomposition* approach.

Five of the investigated examples, `TL`, `HF2D3`, `HF2D4`, `HF2D12`, and `HF2D13`, are stable. Each of the other 28 examples has a single unstable eigenvalue, which should be made stable. For instance, consider the example `HF2D1_M541`, that has $n = 541$, $m = 2$, and $p = 3$. There is an unstable eigenvalue, $\lambda_{47} = 0.45992$ (to five significant digits). To obtain a stable system, it is possible to use the generalized real Schur form (RSF) of the matrix pencil $A - \lambda E$,

$$\widetilde{A} = Q^TAZ, \quad \widetilde{E} = Q^TEZ, \tag{49}$$

with $\widetilde{E}$ upper triangular with positive diagonal elements. Hence, the system can be made stable by changing the sign of $\widetilde{a}_{47,47}$. With this modification the system will remain real. The transformation in (49) is obtained, e.g., using the MATLAB QZ algorithm, `qz`, with the option `'real'` specified as an input argument of the command. (The same idea can be used to generate stable complex systems by calling `qz` without that option.) Figure 2 displays the significant Han-

Table 1: The largest two Hankel singular values for modified COMPl$_e$ib examples of generalized systems. The number j is the index for which λ_j is originally unstable; r is the number of significant Hankel singular values, that is, $\mathrm{rank}(R_oR_c)$. A hyphen is used when there are no unstable eigenvalues.

Example	n	m	p	$\sigma_{1,2}$		j	r
TL	256	2	2	1.018e+18	6.630e+17	-	41
HF2D1	3796	2	3	6.4334e-1	2.7337e-1	328	46
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	2.3534e-1	8.5433e-2	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	6.3154e-1	2.4018e-1	-	5
HF2D13	5	2	4	1.7742e+1	3.3692e+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

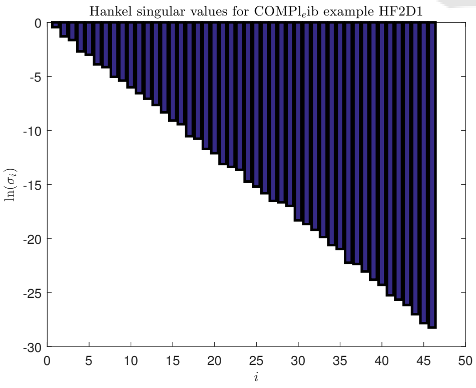

Figure 2: Significant Hankel singular values of the modified example HF2D1 from the COMPl$_e$ib collection.

kel singular values for the larger system HF2D1, of order 3796, modified similarly.

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

The maximum total CPU execution time was of 46.46 minutes, for example HF2D5, with $n = 4489$, $m = 2$, and $p = 4$. The most time consuming step is the reduction to the RSF using the QZ algorithm. The ratios between the timing values needed by this algorithm and the solution of the two reduced equations vary in the range (1.05, 3.35) and the mean value of these ratios is 2.34. The ratios are larger for higher order examples. Similarly, the ratios between the timing values needed by QZ and the computation of the singular values vary in the range (14, 72) and the mean value is 34.19. The total CPU time for each system of order at most 541 was less than 3.16 seconds. Similarly, each system of order 2025 needed less than 224 seconds. Other solvers would need by about 1.7 times more CPU time for each example, since the computation of GSF should be done twice.

All 33 examples have also been modified to obtain and solve complex Lyapunov equations. However, 18 COMPl$_e$ib examples have real eigenvalues only. For these systems, the `qz` function returns real matrices $\widetilde{A}$ and $\widetilde{E}$. To get a complex dynamic system, the matrix $\widetilde{A}$ has been modified by adding $\imath|\lambda_j|/\varepsilon^{1/2}$ to $\widetilde{a}_{jj}$, where j is the index of the real eigenvalue with the minimum modulus, and $\varepsilon \approx 2.22 \cdot 10^{-16}$. The first two largest singular values were the same as in Table 1, except for `HF2D1`, `HF2D2`, `HF2D12`, and `HF2D13`, for which they were somewhat larger.

7 CONCLUSIONS

New numerically attractive algorithms for solving stable generalized complex Lyapunov equations for both continuous- and discrete-time systems is presented. Two equations with the matrices A, E, and A^H, 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. New computational formulas are derived, and related numerical issues are highlighted. The numerical results for a big set of examples, some of large order, based on the COMPl$_e$ib collection, illustrate the performance of the software implementation. The CPU computing time for finding the Hankel singular values can be with 70% smaller than for other approaches. Moreover, new scaling strategies allow to solve badly scaled problems for which other implementations would fail. The proposed solver is currently incorporated in SLICOT Library and MATLAB.

REFERENCES

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. (1999). *LAPACK Users' Guide: Third Edition*. Software · Environments · Tools. SIAM, Philadelphia, MA.

Bartels, R. H. and Stewart, G. W. (1972). Algorithm 432: Solution of the matrix equation $AX + XB = C$. *Comm. ACM*, 15(9):820–826.

Benner, P., Mehrmann, V., Sima, V., Van Huffel, S., and Varga, A. (1999). SLICOT — A subroutine library in systems and control theory. In Datta, B. N. (ed), *Applied and Computational Control, Signals, and Circuits*, vol. 1, pp. 499–539. Birkhäuser, Boston, MA.

Benner, P. and Werner, S. W. R. (2020). Frequency- and time-limited balanced truncation for large-scale second-order systems. pp. 1–30. [Online]. Available: https://arxiv.org/abs/2001.06185v1.

Bini, D. A., Iannazzo, B., and Meini, B. (2012). *Numerical Solution of Algebraic Riccati Equations*. SIAM, Philadelphia, MA.

Cao, X., Benner, P., Duff, I. P., and Schilders, W. (2020). Model order reduction for bilinear control systems with inhomogeneous initial conditions. *Int. J. Control*, 1–10.

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

Golub, G. H. and Van Loan, C. F. (2013). *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, fourth edition.

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

Lancaster, P. and Rodman, L. (1995). *The Algebraic Riccati Equation*. Oxford University Press, Oxford.

Leibfritz, F. and Lipinski, W. (2003). Description of the benchmark examples in *COMPl$_e$ib*. Tech. report, Dep. of Mathematics, University of Trier, Germany.

MathWorks® (2015). *Control System Toolbox*™. The MathWorks, Inc., Natick, MA.

Mehrmann, V. (1991). *The Autonomous Linear Quadratic Control Problem. Theory and Numerical Solution*. Springer-Verlag, Berlin.

Penzl, T. (1998). Numerical solution of generalized Lyapunov equations. *Advances in Comp. Math.*, 8:33–48.

Sima, V. (1996). *Algorithms for Linear-Quadratic Optimization*. Marcel Dekker, Inc., New York.

Sima, V. (2019). Comparative performance evaluation of an accuracy-enhancing Lyapunov solver. *Information, Special Issue ICSTCC 2018: Advances in Control and Computers*, 10(6 215). 22 pages.

Simoncini, V. (2016). Computational methods for linear matrix equations. *SIAM Rev.*, 58(3):377–441.

Stykel, T. (2002). Numerical solution and perturbation theory for generalized Lyapunov equations. *Lin. Alg. Appl.*, 349(5):155–185.

Stykel, T. (2004). Gramian based model reduction for descriptor systems. *Math. Control Signals Syst.*, 16:297–319.

Van Huffel, S., Sima, V., Varga, A., Hammarling, S., and Delebecque, F. (2004). High-performance numerical software for control. *IEEE Control Syst. Mag.*, 24(1):60–76.

Varga, A. (2017). *Solving Fault Diagnosis Problems: Linear Synthesis Techniques*. Springer, Berlin.

Yang, S., Birk, W., and Cao, Y. (2020). A new controllability index based on Hankel singular value. In *Preprints of the 21st IFAC World Congress* (Virtual), pp. 4736–4741, Berlin, Germany.