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

Journal of Engineering Sciences and Innovation
Volume 8, Issue 3 / 2023, pp. 311-322

F. Electrical, Electronics Engineering, Computer Sciences and Engineering

Received ***4 July 2023*** *Accepted* ***29 September 2023***
Received in revised form ***25 August 2023***

Performance improvement in computing the L-infinity norm of descriptor systems

VASILE SIMA*

Technical Sciences Academy of Romania, Bucharest, Romania

Abstract. Efficient and realible algorithms for finding the L-infinity norm for both continuous- and discrete-time descriptor systems have been recently developed. These algorithms exploit the underlying Hamiltonian or symplectic structure of the computational problem. The solver incorporating these advances has been extensively tested on large sets of control applications. Numerical results and comparisons illustrated the good performance and effectiveness of this solver. However, further investigations have shown that the performance can still be improved. The refinements performed include a better selection of the test frequencies used to find a lower bound for the L-infinity norm and of the tolerances for detecting the poles lying on the boundary of the stability domain, a better use of the computer memory hierarchy, an optimal workspace for computations, etc. Such refinements allowed not only to reduce the computing time, but also to improve the accuracy and reliability of the results.

Keywords: descriptor system, linear multivariable systems, numerical methods, robust control, software.

1. Introduction

One of the most important norm for the analysis and design of linear dynamical systems is the L_∞-norm, tphat is often encountered in robust control, model order reduction, and other applications. For instance, the L_∞-norm is used as a robustness measure in the robust control field [1], [2], or as an error measure for model and controller order reduction applications, see [3] and the references therein.
A linear time-invariant (LTI) system can be defined by

$$E\lambda\big(x(t)\big) = Ax(t) + Bu(t), \;\; y(t) = Cx(t) + Du(t), \qquad (1)$$

where $A, E \in \boldsymbol{R}^{n\times n}$, $B \in \boldsymbol{R}^{n\times m}$, $C \in \boldsymbol{R}^{p\times n}$, $D \in \boldsymbol{R}^{p\times m}$, $x(t) \in \boldsymbol{R}^n$ is the state vector, $y(t) \in \boldsymbol{R}^p$ is the output vector, $u(t) \in \boldsymbol{R}^m$ is the input vector, and $\lambda\big(x(t)\big)$

*Correspondence address: vasilesima@ymail.com

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 input vector can include disturbance and control components, while the output vector can contain measured and regulated components. The matrix E can be singular. This happens when model (1) includes algebraic constraints, besides differential or difference equations. Such systems are referred to as *descriptor* (or *singular*) *systems*. It is assumed in the sequel that the matrix pencil $\lambda E - A$ is regular, that is, $\det(\lambda E - A) \not\equiv 0$. The *transfer function* matrix of the system (1) is

$$G(\lambda) = C(\lambda E - A)^{-1}B + D. \quad (2)$$

The L_∞-norm is defined by

$$\|G\|_{L_\infty} := sup_{\omega \in \boldsymbol{R}} \sigma_M(G(i\omega)), \quad (3)$$

for $G \in RL_\infty^{p\times m}(i\omega)$, and by

$$\|G\|_{L_\infty} := sup_{\omega \in [-\pi,\pi)} \sigma_M(G(e^{i\omega})), \quad (4)$$

for $G \in RL_\infty^{p\times m}(e^{i\omega})$, where $\sigma_M(.)$ denotes the maximum singular value, and $RL_\infty^{p\times m}(i\omega)$ and $RL_\infty^{p\times m}(e^{i\omega})$ are

rational subspaces of Banach spaces of all $p \times m$ matrix-valued functions that are bounded on the imaginary axis, or the unit circle, for continuous- and discrete-time systems, respectively. Each $G \in RL_\infty^{p\times m}(i\omega)$ or $G \in RL_\infty^{p\times m}(e^{i\omega})$ has a realization of the form (1). As a convention, $\|G\|_{L_\infty} = \infty$ if G is not in $RL_\infty^{p\times m}(\cdot)$, where $\cdot$ stands for $i\omega$ or $e^{i\omega}$. For continuous-time systems, this happens when G has purely imaginary poles, or when it is improper, that is, $\lim_{\omega\to\infty} G(i\omega) = \infty$. For discrete-time systems, G is not in $RL_\infty^{p\times m}(e^{i\omega})$ when G has *unitary* poles, that is, poles on the unit circle. The poles of G are the controllable and observable eigenvalues of the matrix pencil $\lambda E - A$.

There is a connection between the singular values of $G(i\omega)$ or $G(e^{i\omega})$ and the finite, purely imaginary or unitary eigenvalues, respectively, of some structured matrix pencils [4]. For continuous-time systems, such a pencil is

$$H_c(\gamma) = \begin{bmatrix} \lambda E - A & 0 \\ 0 & \lambda E^T + A^T \end{bmatrix} - \begin{bmatrix} B & 0 \\ 0 & -C^T \end{bmatrix} \begin{bmatrix} -D & \gamma I_p \\ \gamma I_m & -D^T \end{bmatrix}^{-1} \begin{bmatrix} C & 0 \\ 0 & B^T \end{bmatrix}, \quad (5)$$

where γ is a parameter and I_q denotes the identity matrix of order q. The pencil corresponding to discrete-time systems, $H_d(\gamma)$, has a similar formula, but the (2,2) elements of the first two matrices in the right-hand side are $\lambda A^T - E^T$ and $-\lambda C^T$, respectively. (See, for instance, [5] and the references therein.) The following *theorem* is proven in [6]: Assume that $G \in RL_\infty^{p\times m}(i\omega)$, $\gamma > 0$ is not a singular value of D and $\omega_0 \in \boldsymbol{R}$. Then, γ is a singular value of $G(i\omega_0)$ if and only if $H_c(\gamma)$ has the eigenvalue $i\omega_0$. The result for discrete-time systems is obtained by replacing $i\omega_0$ by $e^{i\omega_0}$ (with $\omega_0 \in [-\pi, \pi)$).

The results above have the following consequences [5]: Assume that $G \in RL_\infty^{p\times m}(i\omega)$ and let $\gamma > \min_{\omega\in\boldsymbol{R}} \sigma_M(G(i\omega))$ be not a singular value of D. Then, $\|G\|_{L_\infty} \geq \gamma$ if and only if $H_c(\gamma)$ has finite, purely imaginary eigenvalues.

Similarly, assume that $G \in RL_\infty^{p\times m}(e^{i\omega})$ and let $\gamma > \min\limits_{\omega\in[-\pi,\pi)} \sigma_M(G(e^{i\omega}))$ be not a singular value of D. Then, $\|G\|_{L_\infty} \geq \gamma$ if and only if $H_d(\gamma)$ has unitary eigenvalues. The first part has been proven in [6]. The second part follows from the equivalence of pencils for discrete- and continuous-time systems [7]. These results allow to extend to descriptor systems the quadratically convergent method in [8], [9] for the computation of the L_∞-norm.

Conceptual algorithms are presented in [5], [6] and summarized in [10]. They start with an initial lower bound, γ_l, for the L_∞-norm, found by evaluating $\sigma_M(G(\cdot))$ on the boundaries of the frequency interval $[0,\infty)$ or $[0,\pi)$, respectively, and on further well-chosen inner test frequencies. For continuous-time, *standard systems*, that is, with $E = I_n$, $\sigma_M(D)$ is also included in the set of test frequencies. At each iteration of the algorithm, a value γ is set as $\gamma = (1+\varepsilon)\gamma_l$, where ε is a given tolerance. Then, the finite, purely imaginary or unitary eigenvalues, $i\omega_j$ or $e^{i\omega_j}$, $j = 1,\dots,k$, of $H_c(\gamma)$ or $H_d(\gamma)$, respectively, are used, via a bisection technique, to improve the approximation of the lower bound. Specifically, the *midpoints*, $m_j = \sqrt{\omega_j\omega_{j+1}}$ or $m_j = (\omega_j+\omega_{j+1})/2$, respectively, $j = 1,\dots,k-1$, are obtained and the maximum over j of $\sigma_M(G(im_j))$ or $\sigma_M(G(e^{im_j}))$ is computed. This maximum is then used as the new value of γ_l. When no purely imaginary or unitary eigenvalues are found, the L_∞-norm is set to as $\|G\|_{L_\infty} = \gamma_l$. For continuous-time, standard systems, all γ values are larger than or, sometimes, equal to $\sigma_M(D)$.

The pencils $H_c(\gamma)$ and $H_d(\gamma)$ have a special structure: $H_c(\gamma)$ is a *skew-Hamiltonian/Hamiltonian* pencil, and $H_d(\gamma)$ is a generalization of a *symplectic* pencil. Consequently, their spectra have symmetry with respect to both real and imaginary axes, or with respect to the unit circle in the complex plane, respectively. Finding reliable and accurate eigenvalues of these pencils is key for avoiding failures and for increasing the rate of convergence of the L_∞-norm computational algorithms. In order to make this possible, the pencils $H_c(\gamma)$ and $H_d(\gamma)$ are transformed to some equivalent *even* matrix pencils, and then to *skew-Hamiltonian/Hamiltonian* pencils, whose spectra can be computed with structure-preserving algorithms [11], [12].

2. Computation of eigenvalues

Even numerically stable algorithms for eigenvalue computation may deliver very inaccurate results when applied to the matrix pencils $H_c(\gamma)$ and $H_d(\gamma)$. The matrix to be inverted in (5) is very ill-conditioned if γ is close to a singular value of D. This loss of accuracy can be avoided by replacing $H_c(\gamma)$ and $H_d(\gamma)$ by some extended *skew-Hamiltonian/Hamiltonian* matrix pencils [5], $\overline{H}_c(\gamma)$ and $\overline{H}_d(\gamma)$, of order $2\bar{n} := 2n+p+m+r$, which have the same finite eigenvalues as the original pencils, but only use the given data. The *generalized Cayley transform* and an additional *drop/add transformation* [7] are used to get $\overline{H}_d(\gamma)$. This way, the unit

circle in the complex plane is mapped to the imaginary axis. The number r is 0 if $2n + p + m$ is even, and it is 1, otherwise. More details are given in [5], [10].

There are special algorithms which exploit the structure of the pencils $\overline{H}_c(\gamma)$ or $\overline{H}_d(\gamma)$ and ensure the needed symmetry of the spectra [11], [12]. These algorithms reduce a regular real skew-Hamiltonian/Hamiltonian matrix pencil of order $2\bar{n}$, $\lambda S - H$, where S is *skew-Hamiltonian* ($SJ = (SJ)^T$) and H is *Hamiltonian* ($HJ = -(HJ)^T$), to the following form

$$Q_1^T S J Q_1 J^T = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{11}^T \end{bmatrix}, \quad J Q_2^T J^T S Q_2 = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{11}^T \end{bmatrix}, \quad Q_1^T H Q_2 = \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix},$$
$$J \coloneqq \begin{bmatrix} 0 & I_{\bar{n}} \\ -I_{\bar{n}} & 0 \end{bmatrix}, \tag{6}$$

where Q_1 and Q_2 are orthogonal matrices, S_{11}, T_{11}, H_{11} are upper triangular, H_{22}^T is upper quasi-triangular (that is, block upper-triangular with 1×1 and 2×2 diagonal blocks), and the *formal* matrix product $S_{11}^{-1} H_{11} T_{11}^{-1} H_{22}^T$ is in a real *periodic Schur form* [13]. The first two matrices in (6) are skew-Hamiltonian and the third one is Hamiltonian. The spectrum of $\lambda S - H$ is given by

$$\lambda(S, H) = \pm i \sqrt{\lambda(S_{11}^{-1} H_{11} T_{11}^{-1} H_{22}^T)}, \tag{7}$$

and it can be obtained by using the diagonal blocks. It follows that the finite, purely imaginary eigenvalues correspond to the 1×1 diagonal blocks of the formal matrix product. Consequently, there will be no error in the real parts, hence, a robust and reliable detection of the desired eigenvalues is achieved. Details are given in [12]. Note that he submatrices S_{11} and T_{11} in (7) can be singular. The eigenvalues of the formal matrix product are found using the iterative *periodic QZ algorithm* (pQZ) [13]. To increase the convergence rate, implicitly defined shifts are used and applied via an embedding of the Wilkinson polynomial. But the implicit approach may not converge for some periodic eigenvalue problems, since the shifts involved may be indefinitely unsuitable. Several improvements have been proposed in [14]-[16] to avoid failures and reduce the number of iterations. For instance, in a semi-implicit approach [15] the shifts are chosen based on eigenvalues computed explicitly using a special pQZ algorithm for subproblems of order two. Moreover, it was found that alternating implicit and semi-implicit iterations offers the advantages of both approaches, improving the behavior of the pQZ algorithm [17].

If $E = I_n$, the pencil $H_c(\gamma)$ in (5) represents a standard eigenvalue problem for a Hamiltonian matrix. Let $D(\gamma)$ be the block matrix whose inverse appears in (5). If $D(\gamma)$ is well-conditioned, then it is more efficient to compute the eigenvalues of this Hamiltonian matrix than the eigenvalues of the skew-Hamiltonian/Hamiltonian matrix pencil, $\overline{H}_c(\gamma)$. This approach can be used if $D = 0$ or if the condition number of $D(\gamma)$, $c(D(\gamma))$, is sufficiently small; $c(D(\gamma))$ can be exactly computed, using the extreme singular values of the matrix D. Indeed, $1/c\big(D(\gamma)\big) = (\gamma^2 - \sigma_M^2)/(\gamma^2 - \sigma_m^2)$, where σ_M and σ_m are the maximum and minimum singular values of $D(\gamma)$. Note that γ, as well as the above ratio, increase during iterations, and therefore the condition number decreases. Structured algorithms for solving eigenvalue problems for Hamiltonian matrices are described in [18], [19]. The data

matrices in the Hamiltonian $H_c(\gamma)$ can be used directly via the symplectic URV decomposition,

$$U^T H_c(\gamma) V = \begin{bmatrix} T & G \\ 0 & S^T \end{bmatrix},$$

and the periodic Schur decomposition, where U and V are orthogonal symplectic matrices, S is in a (upper) real Schur form, and T is upper triangular. Such an algorithm is backward stable and preserves the eigenvalue pairings in finite precision arithmetic.

The computation of spectra for Hamiltonian matrices and skew-Hamiltonian/Hamiltonian matrix pencils, as well as for formal matrix products involved in periodic Schur and qz decompositions, can be performed using subroutines from the SLICOT Library [20], available on GitHub, https://github.com/SLICOT/SLICOT-Reference.

3. Implementation issues

A preliminary version of the L_∞-norm solver has been developed several years ago [5], [6]. Recently, a new version, linorms, has been prepared and tested. The solver works on both continuous- and discrete-time, standard or descriptor systems, with or without a feedthrough matrix D. The given matrices A, E, B, and C can optionally be balanced, to make the rows and columns of the system pencil matrix as close in norm as possible. Additional scaling can be performed for matrices with too large or too small elements, to avoid overflows during computations. There are options to check the properness of the transfer function matrix of a continuous-time descriptor system, and to reduce the system order (before computing the L_∞-norm), by removing all uncontrollable and unobservable poles. It is possible to specify an estimate of the frequency where the gain of the frequency response would achieve its peak value. The tolerance ε used to set the accuracy in determining the L_∞-norm should be specified, but other tolerances have default values. The optimal sizes of the real and complex working arrays can be computed by the solver using a special call with those sizes set to -1; the two returned values can then be given as input arguments in a second solver call. The use of this feature could significantly reduce the computing time for systems with large order. The selection of the test frequencies for finding a lower bound of the L_∞-norm has been improved. For discrete-time case, only the eigenvalues with magnitude at most π are considered (since this selection is performed before the transformation which maps the unit circle to the real axis). During the iterative process, the original data matrices (but possibly modified by the optional scaling and/or by reducing the order when testing the system properness) are used. This could ensure more accurate results. Modified tolerances have been employed for detecting eigenvalues on the boundary of the stability domain.

The latest implemented versions for solving skew-Hamiltonian/Hamiltonian eigenvalue problems, as well as for the pQZ algorithm, have been used. A version of the skew-Hamiltonian/Hamiltonian solver that works on panels of columns of the pencil matrices is called for large order problems, aiming to better exploit the memory hierachy of modern computers, and hence increase the computational

efficiency. The standard continuous-time case with $D = 0$ is dealt with directly, in the most efficient way, by solving eigenvalue problems for Hamiltonian matrices during iterations. A similar approach is used for standard continuous-time systems with $D \neq 0$ if the condition number of $D(\gamma_0)$ satisfies the inequality $c\big(D(\gamma_0)\big) < 0.01\ {\varepsilon_M}^{-0.5} \approx 6.71 \cdot 10^5$, where γ_0 is the γ value at the beginning of the iterative process ($\gamma_0 > \sigma_M(D)$ for this case). Otherwise, the skew-Hamiltonian/Hamiltonian solver is called.

4. Numerical results

Extensive testing has been performed to evaluate the linorms 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 linorms source code, SLICOT routines and MATLAB-provided optimized LAPACK and BLAS routines. Tests with with LTI systems from the COMPl$_e$ib collection [21] have been run. The results have been compared to those returned by the MATLAB function norm from Release 2021b.
The COMPl$_e$ib collection includes 124 standard continuous-time systems ($E = I_n$), with several variations. All but 16 problems (for systems of order larger than 2000, with matrices in sparse format) have been tried. The number of solved examples is 152. The tolerance $\sqrt{\varepsilon_M}$ has been used for all runs. The relative error between the results returned by norm and linorms has usually been of the order of $\sqrt{\varepsilon_M}$, sometimes much smaller. Singular E matrices have been obtained by setting $E_{nn} = 0$.

Figures $1-3$ present the performance results for all 152 COMPl$_e$ib examples, with $E = I_n$ and $D = D_1 := [\ D_{21}\ 0\ ]$, where D_{21} is defined in [21] and the zero submatrix has size (p, m). The sums of the total time needed by norm and linorms for all systems are 651.26s and 176.87s, respectively. The time for linorms has always been smaller than that for norm. The minimum, maximum, mean and median values of the CPU time ratios are 1.359, 360.29, 19.31, and 12.97, respectively. The ratios have been limited to 100 in Fig.2. There are eight examples, AC11, CSE1, CSE2, PAS, CM3−CM6, with relative error larger than $\sqrt{\varepsilon_M}$. For AC11 and PAS, linorms found an infinite L_∞-norm, while norm found values of order 10^{19}. For CSE1 and CSE2 linorms returned values larger than 10^{15} and 10^{14}, while norm computed values of order 10^{16} and 10^{17}, respectively. The remaining four examples have errors of orders between 10^{-8} and 10^{-6}. In Fig. 3, the relative errors are taken as zero, if a solver found an infinite value, and the other found a value larger than $1/\varepsilon_M$. Moreover, in order to use a logarithmic scale, all zero error values have been replaced by the closest power of 10 that is 10 times smaller than the smallest nonzero error value.

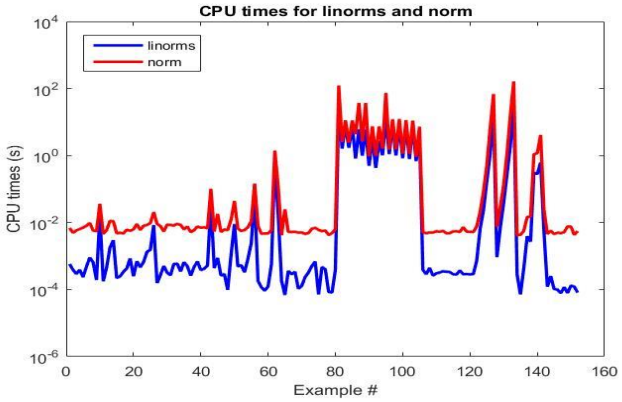

Fig.1. CPU times in seconds for 152 COMPl*e*ib examples ($D = D_1$), using linorms and norm.

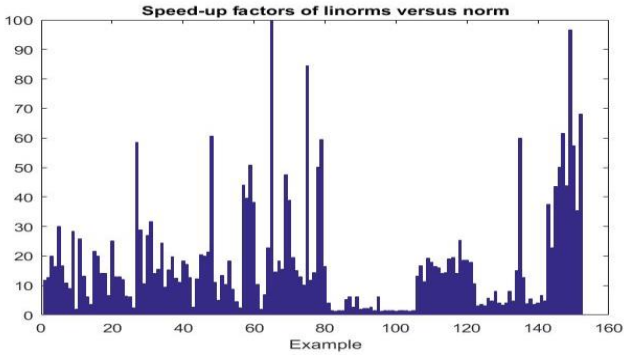

Fig.2. Ratios of the CPU times for norm and linorms for 152 COMPleib examples ($D = D_1$).

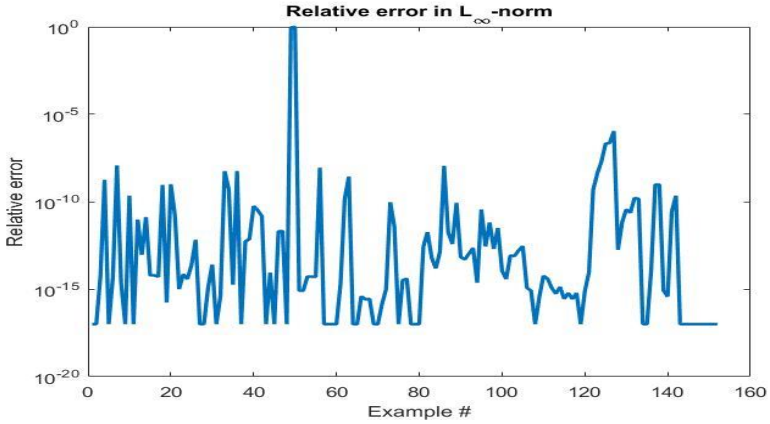

Fig.3. Relative error of the L∞-norm computed by linorms for 152 COMPleib examples ($D = D_1$).

Figures 4 and 5 display the ratios of CPU times for norm and linorms, and the relative errors, respectively, for all large HF2D examples, with $E = I_n$ and $D = D_1$. These include the examples with the smallest ratios in Fig.2. Still, the corresponding time ratios are between 1.35 and 6.21. All their relative errors are smaller than $\sqrt{\varepsilon_M}$.

Figure 6 displays the CPU times for all COMPl$_e$ib examples, taken as discrete-time, when the matrix E is singular ($E_{1:n-1,1:n-1} = I_{n-1}, E_{nn} = 0$) and $D = 0$. A similar behavior appears for $E = I_n$.

Figures 7 and 8 show the CPU times and the relative error, respectively, for all examples, considered as discrete-time systems, with E singular and $D = D_1$. The sums of the total time needed by norm and linorms are 604.55s and 418.18s, respectively. The minimum, maximum, mean and median values of the CPU time ratios are 0.42, 1882.9, 26.92 and 7.87, respectively. There are 27 examples for which linorms needed more CPU time than norm: JE1, CSE2, TL, and all large HF2D examples (numbered 82 : 105 in the figures), but the time differences

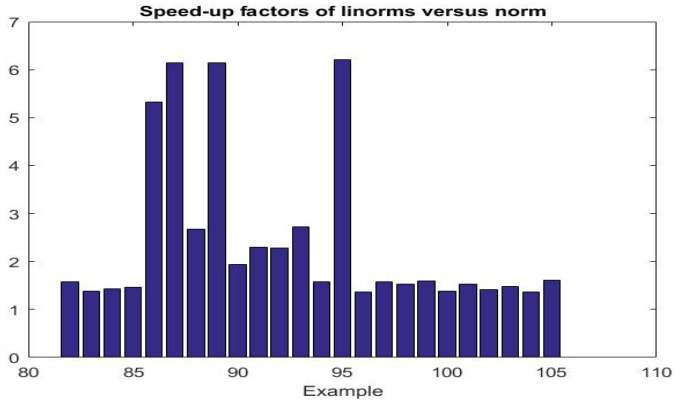

Fig.4. Ratios of the CPU times for norm and linorms for the large HF2D COMPleib examples ($D = D_1$).

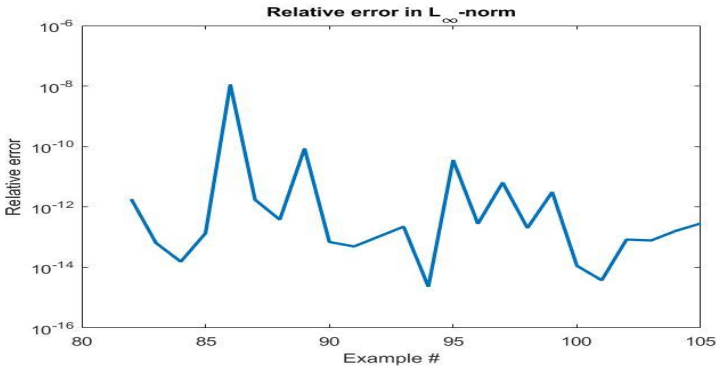

Fig.5. Relative error of the L∞-norm computed by linorms for the large HF2D COMPleib examples ($D = D_1$).

for the first three examples are small. For $D = D_1$, the large HF2D examples have $n \geq 256$, and $m = n + 2$, and consequently, the order $2\bar{n}$ of the skew-Hamiltonian/Hamiltonian problem is big. Other examples with even larger order n, but with small values of m, have been solved faster by linorms, so that the total CPU time was smaller than for norm. There are ten examples, DIS3, IH, CSE2, EB1, EB2, TF1 − TF3, NN11, and NN18, for which the relative error is larger than $\sqrt{\varepsilon_M}$. For CSE2, the error is of order 10^{-5}. For EB1 and EB2, norm found values of order 10^{15}, while linorms found larger values, but with less than one order of magnitude. For the remaining seven examples, norm found values of order 10^{16} or much larger, while linorms found infinite values.

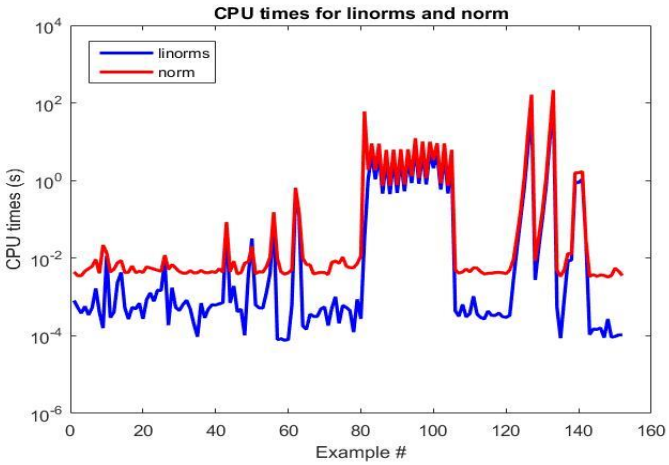

Fig.6. CPU times in seconds for 152 COMPleib examples, taken as discrete-time ($D = 0$), using linorms and norm.

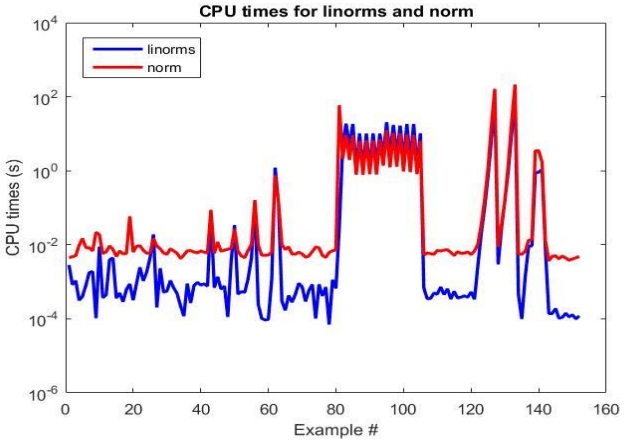

Fig.7. CPU times in seconds for 152 COMPleib examples, taken as discrete-time ($D = D_1$), using linorms and norm.

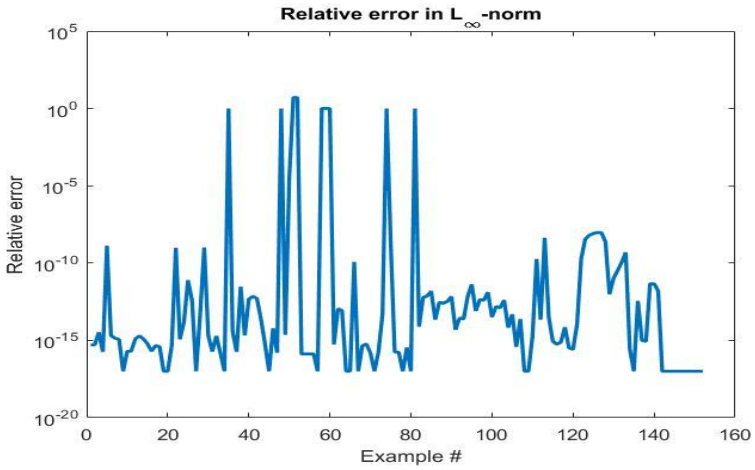

Fig. 8. Relative error of the L∞-norm computed by linorms for 152 COMPleib examples, taken as discrete-time ($D = D_1$).

This section is ended by discussing the results for the classic example in [22],

$$A = \begin{bmatrix} -0.08 & 0.83 & 0 & 0 \\ -0.83 & -0.08 & 0 & 0 \\ 0 & 0 & -0.7 & 9 \\ 0 & 0 & -9 & -0.7 \end{bmatrix},\ B = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1 & -1 \\ 0 & 0 \end{bmatrix},\ C \begin{bmatrix} 0.4 & 0 & 0.4 & 0 \\ 0.6 & 0 & 1 & 0 \end{bmatrix},$$

$$D = \begin{bmatrix} 0.3 & 0 \\ 0 & -0.15 \end{bmatrix}.$$

For getting a result with five digits of accuracy, 17 bisection iterations are reported in [22], while linorms performed just two iterations to achieve at least eight digits of accuracy. The $\mathcal{L}_\infty$-norm computed by linorms is 6.4405165308; its relative error (to the value returned by MATLAB function norm) is about $7.11 \cdot 10^{-11}$. Similarly, the relative error in the correponding frequency is about $1.15 \cdot 10^{-6}$. This illustrates a significantly improved performance.

5. Conclusions

A very important characteristic value for a descriptor system is the $\mathcal{L}_\infty$-norm of its corresponding transfer function matrix. The computation of this norm is essential in robust control, model order reduction, and other applications. Efficient and reliable algorithms for finding the $\mathcal{L}_\infty$-norm for continuous- and discrete-time descriptor systems have been briefly described. The underlying Hamiltonian or symplectic structure of the associated matrix pencils is exploited. The original pencils are transformed into skew-Hamiltonian/Hamiltonian pencils, conserving the finite eigenvalues for continuous-time systems; for discrete-time systems the eigenvalues are mapped by a Cayley transform. This endeavor allowed the use of structure-exploiting algorithms for eigenvalue computation during the iterative process. An improved solver has been developed and will be made available in the

SLICOT Library. Numerical results and comparisons with the state-of-the-art MATLAB function norm illustrate the good performance and effectiveness of this new solver.

References

[1] Zhou K., Doyle J.D., *Essentials of Robust Control*, Prentice Hall, 1st edition, 1998.
[2] Losse P., Mehrmann V., Poppe L., Reis T., *The modified optimal* H_∞ *control problem for descriptor systems*, SIAM J. Control Optim., **47**, 6, 2008, p. 2795–2811.
[3] Mehrmann V., Stykel T., *Balanced truncation model reduction for large-scale systems in descriptor form*, Benner, P., Mehrmann, V., D. Sorensen (eds.), Dimension Reduction of Large-Scale Systems, vol. 45 of Lecture Notes in Computational Science and Engineering, ch. 3, p. 89–116, Springer-Verlag, Berlin, Heidelberg, New York, 2005.
[4] Genin Y., Van Dooren P., Vermaut V., *Convergence of the calculation of* H_∞*-norms and related questions*, Proceedings MTNS-98, 1998, p. 429–432.
[5] Benner P., Sima V., Voigt M., *Robust and efficient algorithms for* L_∞*-norm computation for descriptor systems*, Preprints of the 7th IFAC Symposium on Robust Control Design (ROCOND'12), Aalborg, Denmark, June 20–22, 2012, p. 189–194.
[6] Benner P., Sima V., Voigt M., *L_∞-norm computation for continuous-time descriptor systems using structured matrix pencils*, IEEE Trans. Automat. Control, **57**, 1, 2012, p. 233–238.
[7] Xu H., *On equivalence of pencils from discrete-time and continuous-time control*, Lin. Alg. Appl., **414**, 1, 2006, p. 97–124.
[8] Bruinsma N.A., Steinbuch M., *A fast algorithm to compute the H_∞-norm of a transfer function matrix*, Syst. Control Lett., **14**, 4, 1990, p. 287–293.
[9] Boyd S., Balakrishnan V., *A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its L_∞-norm*, Syst. Control Lett., **15**, 1, 1990, p. 1–7.
[10] Sima V., *Efficient computation of the L-infinity norm of descriptor systems*, Journal of Engineering Sciences and Innovation, **7**, 4, 2022, p. 463–472.
[11] Benner P., Byers R., Losse P., Mehrmann V., Xu H., *Numerical solution of real skew-Hamiltonian/Hamiltonian eigenproblems*, Techn. Rep., Nov. 2007.
[12] Benner P., Sima V., Voigt M., *Algorithm 961: FORTRAN 77 subroutines for the solution of skew-Hamiltonian/Hamiltonian eigenproblems*, ACM Trans. Math. Softw., **42**, 3, Article No. 24 2016, p. 1–26.
[13] Bojanczyk, A.W., Golub, G., Van Dooren, P., *The periodic Schur decomposition: Algorithms and applications*, Luk, F.T. (ed.) SPIE Conference Advanced Signal Processing Algorithms, Architectures, and Implementations III, vol. 1770, 1992, p. 31–42.
[14] Sima, V., *Computation of initial transformation for implicit double step in the periodic QZ algorithm*, Precup, R.E. (ed.) 2019 23th International Conference on System Theory, Control and Computing, 2019, p. 7–12.
[15] Sima V., *A new semi-implicit approach for the periodic QZ algorithm*, 2020 24th International Conference on System Theory, Control and Computing, 2020, p. 190–195.
[16] Sima V., Gahinet P., *Improving the convergence of the periodic QZ algorithm*, Gusikhin O., Madani K., Zaytoon J. (eds.), 16th International Conference on Informatics in Control, Automation and Robotics, vol. 1, 2019, p. 261-268.
[17] Sima V., Gahinet P., *Alternating implicit and semi-implicit iterations in the periodic QZ algorithm*, Gusikhin O., Madani K., Zaytoon (Eds.), Informatics in Control, Automation and Robotics, Lecture Notes in Electrical Engineering, vol. 793, Springer International Publishing, Cham, pp. 47–71, 2022.
[18] Benner P., Mehrmann V., Xu H., *A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils*, Numer. Math., **78**, 3, 1998, p. 329–358.

[19] Kressner D., *Numerical Methods for General and Structured Eigenvalue Problems*, Lecture Notes in Computational Science and Engineering, vol. 46. Springer, Berlin, 2005.
[20] 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.
[21] 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.
[22] Boyd S., Balakrishnan V., Kabamba P., *A bisection method for computing the H_∞ norm of a transfer matrix and related problems*, Mathematics of Control, Signals, and Systems, **2**, 3, 1989, p. 207-219.