Development of the Systems and Control Library SLICOT

P. Benner[1,2] and M. Voigt[1]

[1]Max Planck Institute for Dynamics of Complex Technical Systems Magdeburg, Research Group Computational Methods in Systems and Control Theory
[2]Chemnitz University of Technology, Department of Mathematics, Research Group Mathematics in Industry and Technology

Introduction

The software library SLICOT provides Fortran 77 implementations of numerical algorithms for computations in systems and control theory. Based on routines from the BLAS and LAPACK libraries, SLICOT provides methods for the design and analysis of control systems. The current version of SLICOT consists of about 500 user-callable and computational routines in various domains of systems and control.

Basic Ideas Behind the Library

1. Usefulness of algorithms.
2. Robustness: Routines must return reliable results or an error/warning indicator.
3. Numerical stability and accuracy: The results are as good as can be expected when working at a given precision. If possible an estimate of the achieved accuracy should be given.
4. Performance with respect to speed and memory requirements: Although important because of ever increasing complexity of control problems, this objective may never be met at cost of the two previous ones.
5. Portability and reusability: The library should be independent of platforms.
6. Standardization: The library is based on rigorous programming and documentation standards.
7. Benchmarking: A standardized set of examples that allows an evaluation of the performance of a method as well as the implementation with respect to correctness, accuracy, and speed. Benchmarking gives also insight in the behavior of the method and its implementation in extreme situations.

Structure of the Library

SLICOT is structured into different chapters, each of them covers a particular field in systems and control theory:

- A : Analysis Routines
- B : Benchmark and Test Problems
- C : Adaptive Control (currently empty)
- D : Data Analysis
- F : Filtering
- I : Identification
- M : Mathematical routines
- N : Nonlinear Systems
- S : Synthesis Routines
- T : Transformation Routines
- U : Utility Routines

Cooperations & Coworkers

- The Numerics in Control Network NICONET e.V.
- SynOptio GmbH
- The MathWorks, Inc.
- Dr. Vasile Sima (ICI Bucharest, Romania)
- Prof. Dr. Volker Mehrmann (TU Berlin, Germany)

References

[1] P. Benner, R. Byers, V. Mehrmann, and H. Xu, *Numerical computation of deflating subspaces of embedded Hamiltonian pencils*, tech. rep., Chemnitz University of Technology, Department of Mathematics, Germany, June 1999. SFB393-Preprint 99-15.

[2] P. Benner, V. Sima, and M. Voigt, *Algorithm xxx: Fortran 77 subroutines for the solution of skew-Hamiltonian/Hamiltonian eigenproblems, part I: algorithms and applications*. In preparation for ACM Trans. Math. Softw.

[3] —, *Algorithm xxx: Fortran 77 subroutines for the solution of skew-Hamiltonian/Hamiltonian eigenproblems, part II: implementation and numerical results*. In preparation for ACM Trans. Math. Softw.

[4] —, *$\mathcal{L}_\infty$-norm computation for continuous-time descriptor systems using structured matrix pencils*, IEEE Trans. Automat. Control, 57 (2012), pp. 233–238.

For more information on SLICOT visit `www.slicot.org`.

Structure-Preserving Algorithms for Skew-Hamiltonian/Hamiltonian Eigenvalue Problems

Let $\mathcal{J} := \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$. We call

- the matrix $\mathcal{S} \in \mathbb{R}^{2n\times 2n}$ *skew-Hamiltonian* if $(\mathcal{S}\mathcal{J})^T = -\mathcal{S}\mathcal{J}$,
- the matrix $\mathcal{H} \in \mathbb{R}^{2n\times 2n}$ is *Hamiltonian* if $(\mathcal{H}\mathcal{J})^T = \mathcal{H}\mathcal{J}$,
- the matrix pencil $\lambda\mathcal{S} - \mathcal{H} \in \mathbb{R}^{2n\times 2n}$ *skew-Hamiltonian/Hamiltonian* if $\mathcal{S}$ is skew-Hamiltonian and $\mathcal{H}$ is Hamiltonian.

Some Structural Properties:

- block structure:

$$\lambda\mathcal{S} - \mathcal{H} = \lambda \begin{bmatrix} F & G \\ H & F^T \end{bmatrix} - \begin{bmatrix} R & S \\ T & -R^T \end{bmatrix},$$
$$G = -G^T,\ H = -H^T,\ S = S^T,\ T = T^T,$$

- Hamiltonian spectral symmetry: $\lambda \in \Lambda(\mathcal{S},\mathcal{H}) \longrightarrow -\lambda, \overline{\lambda}, -\overline{\lambda} \in \Lambda(\mathcal{S},\mathcal{H})$,

For many applications this eigenvalue structure is essential. Thus also numerical methods must respect and preserve this structure to obtain meaningfull results. SLICOT provides a new structure-preserving solver which is also often faster than the ordinary QZ algorithm for smaller problems.

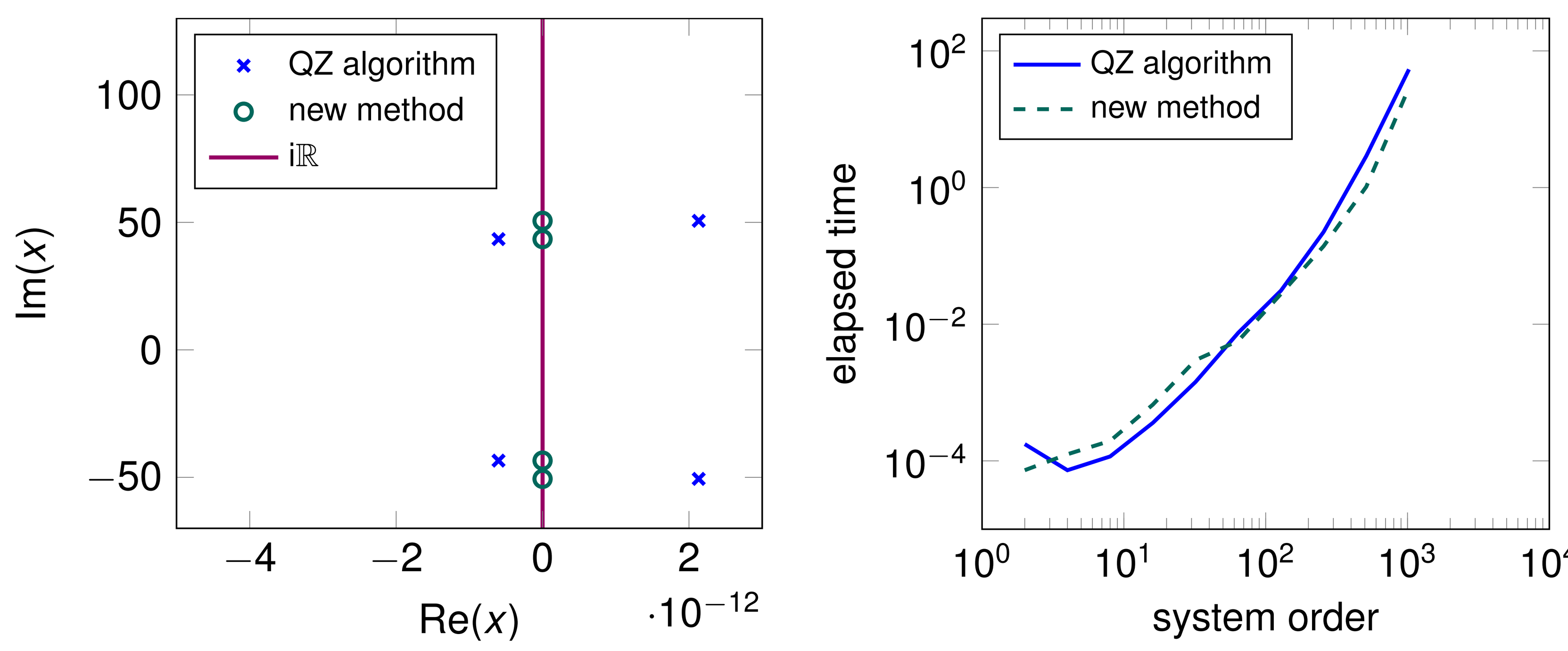

Figure 1: Comparison of the SLICOT routine `MB04BD` with the QZ algorithm: computed eigenvalues (left) and CPU times (right)

$\mathcal{L}_\infty$-Norm Computation for Descriptor Systems

Consider continuous linear time-invariant descriptor system

$$E\dot{x}(t) = Ax(t) + Bu(t),$$
$$y(t) = Cx(t) + Du(t),$$

with transfer function

$$G(s) := C(sE - A)^{-1}B + D.$$

For G which are bounded on the imaginary axis, the $\mathcal{L}_\infty$-norm is defined by

$$\|G\|_{\mathcal{L}_\infty} := \sup_{\omega\in\mathbb{R}} \sigma_{\max}(G(\mathrm{i}\omega)).$$

Computation of the norm is connected to the spectrum of the skew-Hamiltonian/Hamiltonian pencils

$$\lambda N - M_\gamma = \lambda \begin{bmatrix} E & 0 \\ 0 & E^T \end{bmatrix} - \left(\begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & -C^T \end{bmatrix} \begin{bmatrix} -D & \gamma I \\ \gamma I & -D^T \end{bmatrix}^{-1} \begin{bmatrix} C & 0 \\ 0 & B^T \end{bmatrix} \right).$$

Theorem: Under some conditions it holds: $\|G\|_{\mathcal{L}_\infty} < \gamma$ if and only if $\lambda N - M_\gamma$ has no purely imaginary eigenvalues.

$\longrightarrow$ An iteration over γ is used to determine the norm value.

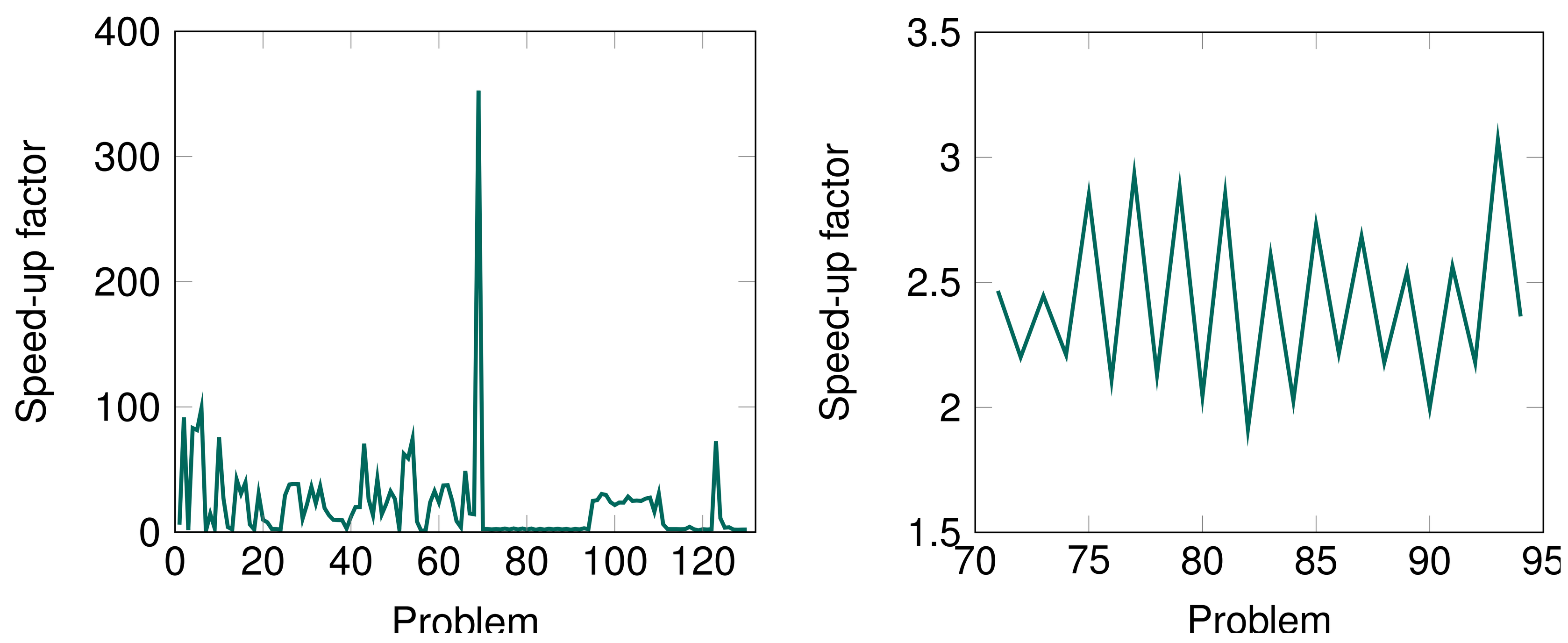

Figure 2: Speed-ups of the SLICOT routine versus the Control System Toolbox of MATLAB