1

ISSN 1373-6787

July 1998

NICONET Newsletter

Distributed by: **Working Group on Software WGS**

Contents: **Page**

Contact address of the WGS: Mrs. Ida Tassens, Secretary of WGS
Katholieke Universiteit Leuven
Dept. of Electrical Engineering (ESAT-SISTA)
Kardinaal Mercierlaan 94
3001 Leuven-Heverlee, Belgium

email: ida.tassens@esat.kuleuven.ac.be
phone: + 32 16 32 17 09
fax: + 32 16 32 19 70

1 Editorial

Welcome to the first issue of the NICONET newsletter which informs you about all future developments of the SLICOT library. This WGS activity is now integrated in the EC thematic network, entitled *"Network for development and evaluation of numerically reliable software in control engineering and its implementation in production technologies"* with acronym NICONET. More information on that project, as well as a list of all partners involved, appeared in the latest WGS newsletter 12[1].

NICONET started on January 1, 1998, and during our kick-off meeting in Leuven on January 22-23 we discussed the main activities of NICONET, in particular the future developments of the SLICOT library. More detailed information on the present status and the new SLICOT tools to be added is given in the next Sections 4 to 8. These new tools require new standards, as described in Section 2, for high-performance SLICOT routines running on parallel machines and for dealing with nonlinear systems in robotics. In addition, much attention is paid to the development of a SLICOT benchmark library for assessing the performance, reliability and versatility of SLICOT in industry. More details are provided in Section 3. Four researchers are involved in our project to coordinate and execute all NICONET tasks. In particular, we welcome Vasile Sima from Romania, Petko Petkov and Mihail Constantinov from Bulgaria, and Cristian Oară from Romania.

As announced in WGS newsletter 12, the NICONET newsletter will appear twice a year (in July and January) and will inform you on the evolution of the SLICOT library and other CACSD software, as well as about our upcoming workshops which are open for everybody. Also contributions from our readers related to control software are welcome. We invite all of you to submit your contributions to the WGS secretariat. Each submission will be acknowledged and reviewed by the NICONET newsletter editor. If this information is not sufficient for you, we invite you to subscribe to our NICONET E-letter[2] which informs you quarterly about the newest SLICOT updates and any feedback given by SLICOT users (on performance, improvements, new suggestions).

And last but not least, if you are developing control software which might be a valuable addition to the SLICOT library, you can submit your software to the NICONET secretariat or directly to the appropriate topic coordinator. Also, carefully chosen benchmark examples for control methods are welcome. More details on how to contribute are given in Section 2.

Sabine Van Huffel
Chairperson of WGS and Coordinator of NICONET.

[1] accessible through World Wide Web (URL http://www.win.tue.nl/wgs/) or via anonymous ftp from `wgs.esat.kuleuven.ac.be/pub/WGS/NEWSLETTER/issue12.ps.Z`.

[2] To subscribe, send e-mail to majordomo@win.tue.nl with the message "subscribe niconet" in the body

2 New SLICOT standards for parallel machines and nonlinear systems

2.1 Standards for parallel machines

One of the aims of the NICONET project is the development of a parallel version of the SLICOT library, which will be called PSLICOT, that will contain the same functionality as the sequential SLICOT but offering better performance by using parallel systems.

As a numerical software library like SLICOT, PSLICOT must offer an homogeneous interface to the user, and it must offer good performance. A SLICOT working note 1998-1[3] has been prepared to determine the common standard for the design and implementation of the parallel subroutines. Good performance will follow from the use of standard parallel linear algebra and communication libraries, such as ScaLAPACK, PBLAS and BLACS.

The parallel model used is distributed memory architecture and the programming model is message passing. The process for initialising the processors and distributing the data will be the same as in ScaLAPACK, that is, throughout the use of BLACS. This will ensure portability and efficiency, since many architectures have their own native versions of this library, at least throughout optimised versions of MPI.

For facilitating the process of migrating from sequential SLICOT to parallel PSLICOT, the parallel subroutine interface is quite similar to the SLICOT standard, with the minimum changes required for the parallel programming. Also, standards for documentation and implementation have been described.

The implementation standards rely mainly on the use of FORTRAN77, modular and structured programming, meaningful naming of variables and functions and a set of rules which are common to the SLICOT standard format. Parallel data distribution is recommended to be a variant of Cyclic 2D Block distribution when possible. Default values for the parameters that define the parallel distribution (topology, block size, number of processors,..) should be provided.

Regarding documentation, the same standards as SLICOT are required in PSLICOT, with the inclusion of new values related to parallelism, such as parallel complexity or data distribution. Examples will be provided in the documentation reports. Moreover, in order to provide the user with larger examples, ftp adresses will be given.

The rules that have been defined for the implementation of PSLICOT will lead to an efficient numerical library, which will require minimal changes to the routines that migrate from SLICOT to PSLICOT. Portability is ensured also by the use of standard parallel libraries.

2.2 Standards for nonlinear systems

The nonlinear dynamic systems considered are described either by *ordinary differential equations* (ODEs)

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), p, t) \\ y(t) &= g(x(t), u(t), p, t), \end{aligned} \tag{1}$$

[3]Report in preparation. This report will be available in October 1998 by anonymous ftp from `wgs.esat.kuleuven.ac.be/WGS/REPORTS/SLWN1998-1.ps.Z`

or by *differential-algebraic equations* (DAEs)

$$\begin{aligned} 0 &= f(\dot{x}(t), x(t), u(t), p, t) \\ y(t) &= g(\dot{x}(t), x(t), u(t), p, t), \end{aligned} \tag{2}$$

where $x(t)$ is the state vector, $u(t)$ is the input vector, $y(t)$ is the output vector, p is the parameter vector. The ODE and DAE classes of nonlinear dynamics models include multibody mechanical systems, many chemical process models, dynamic aircraft models, electric circuits, and so on. An important characteristic of all the above models is the explicit parametric dependence, that is, parameters like masses, moments of inertia, heights, etc. are explicitly contained in the model definition. This allows for instance to perform parameter studies or parameter tuning on the above nonlinear models.

The SLICOT standard for nonlinear systems[4] provides standardized Fortran interfaces for the description of nonlinear systems to allow an easy interfacing with software for integration of differential equations, nonlinear programming and solving nonlinear equations. The proposed standard defines a set of low-level interface subroutines to describe parametrized, nonlinear dynamic systems of forms (1) or (2). Additionally to the interface routines for the functions f and g, definitions are provided for interface routines for various Jacobians $\partial f/\partial x$, $\partial f/\partial u$, $\partial f/\partial p$, etc. A nonlinear system model which is described according to this interface definition is called **DSblock** (Dynamic System block). The DSblock interface may serve as a neutral *model bus* to exchange nonlinear models between different modelling and simulation environments. The proposed DSblock interface is compatible with the automatically generated DSblock codes in modern object oriented modelling environments (like the Dymola[5] software based on the **Modelica** modelling language[6]).

The proposed first version of standard interfaces is intended mainly for an easy interfacing with popular ODE and DAE solvers. However, these interfaces are suitable to be used with routines for nonlinear systems solvers or gradient based optimization as well. For the efficient evaluation of the gradients, special higher level interface routines are necessary, since the evaluation of each gradient component requires the integration of the ODE (1) over a time interval. Alternatively reverse-time integration is sometimes used. The standardization of gradient computations raises several delicate technical aspects and will constitute an issue of further extension of the present standard.

Andras Varga

[4] Report available by anonymous ftp from `wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/SLWN1998-4.ps.Z`

[5] http://www.dynasim.se/products.html

[6] http://www.Dynasim.se/Modelica/

3 The SLICOT benchmark library

After detailed discussions at the NICONET meetings in Leuven (January 22-23, 1998) and Chemnitz (June 18–19, 1998), a standard for the implementation of benchmark collections was developed and described in SLICOT Working Note 1998-5[7].

The following list shows the topics which will be covered by the SLICOT benchmark library. Furthermore, it contains the NICONET partners who are responsible for collecting the benchmark examples and implementing the benchmark routines.

- Continuous-time Lyapunov equations (V. Mehrmann, T. Penzl – *TU Chemnitz*)
- Continuous-time Riccati equations (P. Benner – *Universität Bremen*)
- Continuous-time state-space systems (V. Mehrmann, T. Penzl – *TU Chemnitz*)
- Discrete-time Lyapunov equations (V. Mehrmann, T. Penzl – *TU Chemnitz*)
- Discrete-time Riccati equations (P. Benner – *Universität Bremen*)
- Discrete-time state-space systems (V. Mehrmann, T. Penzl – *TU Chemnitz*)
- H_∞ control (D.W. Gu – *Leicester University*)
- Matrix exponentials (P. Van Dooren – *Université Catholique de Louvain*)
- Periodic systems (P. Van Dooren – *Université Catholique de Louvain*)
- Subspace identification (M. Verhaegen – *TU Delft*)

The SLICOT benchmark library still lacks examples which represent "real world" problems or challenges for numerical algorithms. Persons outside NICONET are strongly encouraged to provide examples to the library. Benchmark examples can be submitted either to the NICONET partner who is responsible for the respective topic or to the NICONET partner *TU Chemnitz*:

Prof. Volker Mehrmann
Fakultät für Mathematik
TU Chemnitz
D–09107 Chemnitz
Germany
e-mail: mehrmann@mathematik.tu-chemnitz.de

There are only moderate requirements for the submission of benchmark examples by external contributors. These requirements are described in the SLICOT Working Note 1998-5[7].

Thilo Penzl and Volker Mehrmann

[7]Report available by anonymous ftp from `wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/SLWN1998-5.ps.Z`

4 SLICOT basic software tools

The current status of the SLICOT library covers already a large number of basic mathematical and system theoretic computations. To guarantee a proper distribution of the SLICOT library, the product has been made freely available. Most of the 90 SLICOT routines of Release 2.0 have been upgraded to the new standards and have been made available on ftp. However, to match better the recent system theoretic advances of the last decade, new basic mathematical tools have to be added to the library. In the first six months our main emphasis went to completing the key chapters of mathematical routines, transformation routines and synthesis routines related to standard state-space models. Later on, our emphasis will go towards basic linear algebra techniques for (i) generalized state space models and factorizations of transfer functions[8], and (ii) fast numerical procedures for structured matrices and perturbations.

4.1 Algorithms and Software for Basic Control Problems

Consider the linear state-space system G

$$\begin{array}{rcl} \lambda x(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) + Du(t), \end{array} \tag{3}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times m}$, and where λ is either the differential operator d/dt for a continuous-time system or the advance operator z for a discrete-time system. The *transfer function matrix* is the $p \times m$ proper rational matrix

$$G(\lambda) = C(\lambda I - A)^{-1}B + D. \tag{4}$$

For an invertible matrix T, two state-space systems (A, B, C, D) and $(\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D})$ related by

$$\widetilde{A} = T^{-1}AT, \quad \widetilde{B} = T^{-1}B, \quad \widetilde{C} = CT, \quad \widetilde{D} = D, \tag{5}$$

are called *similar* and the transformation (5) is called a *similarity transformation*. Note that similar state-space systems have the same transfer matrix.

The similarity transformations are the basic preprocessing tools for most analysis, model conversion and synthesis problems. From a numerical point of view, it is important that the transformation matrix T is a well-conditioned (ideally an orthogonal) matrix.

In our library we added routines to perform

- a diagonal scaling T to reduce the 1-norm of the transformed system matrix (such a transformation is frequently necessary to improve the accuracy of subsequent numerical computations involving the system matrices)

- a similarity transformation resulting in a state matrix $\widetilde{A}$ in *block-diagonal form*. Such an additive decomposition of G is useful for many control computations, as for instance [8]: discretization of continuous-time systems, computation of frequency responses, modal approach to model reduction, evaluation of the transfer-function matrix of large scale systems.

- a similarity transformation resulting in a state matrix $\widetilde{A}$ in stable/unstable additive spectral decomposition where A_1 and A_2 contain the systems poles lying in the stable and unstable regions, respectively.

[8]Report available by anonymous ftp from `wgs.esat.kuleuven.ac.be/WGS/REPORTS/SLWN1998-3.ps.Z`

- an orthogonal state-space coordinate transformation determined to reduce A to a real Schur form $\widetilde{A}$. Such a reduction is frequently necessary as a preprocessing step in the routines for balancing related model reduction.
- an orthogonal similarity transformation to reduce the pair (A, B) to the controllability staircase form. Such a transformation, with optional accumulation of the transformation matrix Q, is useful in an efficient implementation of the minimal realization routine.

For the solution of Lyapunov equations a new function was added that implements Hammerling's method for positive definite solutions $X = U^T U$ of the continuous-time equations of the form

$$AX + XA^T + BB^T = 0, \quad A^T X + XA + C^T C = 0$$

and of the discrete-time equations of the form

$$AXA^T - X + BB^T = 0, \quad A^T XA - X + C^T C = 0$$

where $A \in \mathbb{R}^{n \times n}$ is stable, (A, B) is controllable and (A, C) is observable. The factor U of the solution X can be either upper or lower triangular. This flexibility is useful for balancing transformations.

References

[1] C. Bavely and G. W. Stewart. An algorithm for computing reducing subspaces by block diagonalization. *SIAM J. Numer. Anal.*, 16:359–367, 1979.

[2] G. Grübel, A. Varga, A. van den Boom, and A. J. Geurts. Towards a coordinated development of numerical CACSD software: the RASP/SLICOT compatibility concept. In *Prepr. of IEEE/IFAC Symp. CACSD'94, Tucson, Arizona*, pages 499–504, 1994.

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

[4] B. Kågström and P. Van Dooren. Additive decomposition of a transfer function with respect to a specified region. In *Proc. MTNS Symp., Brussels*, 1989.

[5] C. B. Moler and C. F. Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review*, 20:801–836, 1978.

[6] V. Sima. On the real Schur form in linear control system design. *Rev. Roum. Scie. Techn. – Electrotech. et Energ.*, 25, 1980.

[7] P. Van Dooren. The generalized eigenstructure problem in linear systems theory. *IEEE Trans. Autom. Control*, 26:111–129, 1981.

[8] A. Varga. Computational techniques based on the block-diagonal form for solving large systems modeling problems. In *Proc. 1993 IEEE Conference on Aerospace Control Systems, Westlake Village, CA*, pages 693–697, 1993.

[9] A. Varga and V. Sima. BIMAS - A basic mathematical package for computer aided systems analysis and design. In J. Gerter and L. Keviczky, editors, *Proc. of 9-th IFAC World Congress, Budapest, Hungary*, 1985.

4.2 List of Additional State-Space Routines

4.2.1 Mathematical Routines

Name	Function
MB03RD	computes the block-diagonal form of a square matrix
MB03QD	reorders the eigenvalues of a real Schur matrix according to several reordering criteria
MB03UD	computes all, or part, of the singular value decomposition of an upper triangular matrix

4.2.2 Transformation Routines

Name	Function
TB01ID	performs the scaling of a state-space model
TB01KD	computes the terms G_1 and G_2 of an additive spectral decomposition of a transfer-function matrix G with respect to a specified region of the complex plane
TB01LD	performs an orthogonal similarity trasformation to reduce the system state matrix to an ordered real Schur form
TB01OD	computes the orthogonal controllability staircase form of a state-space model
TB01WD	performs an orthogonal similarity trasformation to reduce the system state matrix to the real Schur form

4.2.3 Synthesis Routines

Name	Function
SB03OD	solves for $X = \mathrm{op}(U)'\mathrm{op}(U)$ either the stable non-negative definite continuous-time Lyapunov equation $\mathrm{op}(A)'X + X\mathrm{op}(A) = -\sigma^2\mathrm{op}(B)'\mathrm{op}(B)$ or the convergent non-negative definite discrete-time Lyapunov equation $\mathrm{op}(A)'X\mathrm{op}(A) - X = -\sigma^2\mathrm{op}(B)'\mathrm{op}(B)$, where $\mathrm{op}(K) = K$ or K'.

Paul Van Dooren and Andras Varga

5 SLICOT tools for model reduction

The basic model reduction algorithms implemented and standardized in SLICOT[9] belong to the class of methods based on or related to balancing techniques [11, 10, 6] and are primarily intended for the reduction of linear, stable, continuous- or discrete-time systems. All methods rely on guaranteed error bounds and have particular features which recommend them for use in specific applications. The basic methods combined with coprime factorization or spectral decomposition techniques can be used to reduce unstable systems [9] or to perform frequency-weighted model reduction [8, 5].

The basis of the standardization of the model reduction routines for SLICOT is the collection of routines available in the RASP-MODRED library [16] for the reduction of stable systems. The RASP-MODRED routines are based on the linear algebra standard package LAPACK [2]. The underlying algorithms represent the latest developments of various procedures for solving computational problems appearing in the context of model reduction. It is worth mentioning that the new standardized model reduction routines of SLICOT are generally superior to those available in the model reduction tools of commercial packages [4, 3, 1].

5.1 The algorithms for model reduction

Consider the n-th order original state-space model $G := (A, B, C, D)$ with the *transfer-function matrix* (TFM) $G(\lambda) = C(\lambda I - A)^{-1}B + D$, and let $G_r := (A_r, B_r, C_r, D_r)$ be an r-th order approximation of the original model ($r < n$), with the TFM $G_r = C_r(\lambda I - A_r)^{-1}B_r + D_r$. The balancing based model reduction methods starts by performing a similarity transformation Z yielding

$$\left[\begin{array}{c|c} Z^{-1}AZ & Z^{-1}B \\ \hline CZ & D \end{array}\right] := \left[\begin{array}{cc|c} A_{11} & A_{12} & B_1 \\ A_{21} & A_{22} & B_2 \\ \hline C_1 & C_2 & D \end{array}\right],$$

and then defining the reduced model (A_r, B_r, C_r, D_r) simply as $(A_{11},\ B_1,\ C_1,\ D)$. Note that when writing $Z := [\, T\ U\,]$ and $Z^{-1} := [\, L^T\ V^T\,]^T$, then $\Pi = TL$ is a projector on T along L and $LT = I_r$. Thus the reduced system can be defined as $(A_r, B_r, C_r, D_r) = (LAT,\ LB,\ CT,\ D)$. Alternatively, the partitioned representation above can be used to construct a *singular perturbation approximation* (SPA). The matrices of the reduced model in this case are given by

$$\begin{array}{rcl} A_r &=& A_{11} + A_{12}(\gamma I - A_{22})^{-1}A_{21}, \\ B_r &=& B_1 + A_{12}(\gamma I - A_{22})^{-1}B_2, \\ C_r &=& C_1 + C_2(\gamma I - A_{22})^{-1}A_{21}, \\ D_r &=& D + C_2(\gamma I - A_{22})^{-1}B_2. \end{array} \qquad (6)$$

where $\gamma = 0$ for a continuous-time system and $\gamma = 1$ for a discrete-time system. Note that SPA formulas preserve the DC-gains of stable original systems.

Specific requirements for model reduction algorithms are formulated and discussed in [16]. Such requirements are: (1) applicability of methods regardless the original system is minimal or not; (2) emphasis on enhancing the numerical accuracy of computations; (3) use of numerically reliable procedures.

[9] Report available by anonymous ftp from `wgs.esat.kuleuven.ac.be/WGS/REPORTS/SLWN1998-2.ps.Z`

The first requirement can be fulfilled by computing L and T directly, without determining Z or Z^{-1}. In particular, if the original system is not minimal, then L and T can be chosen to compute an *exact* minimal realization of the original system [15].

The emphasis on improving the accuracy of computations led to so-called algorithms with *enhanced accuracy*. In the balancing related model reduction methods, the matrices L and T are determined from the controllability and observability gramians P and Q, respectively, which satisfy the Lyapunov equations

$$AP + PA^T + BB^T = 0, \quad A^TQ + QA + C^TC = 0$$

in the continuous-time case, or

$$APA^T + BB^T = P, \quad A^TQA + C^TC = Q$$

in the discrete-time case. The gramians can always be determined directly in Cholesky factorized forms $P = S^TS$ and $Q = R^TR$ (i.e. with S and R upper-triangular) by using numerically reliable algorithms [7]. The computation of L and T can be done on basis of the *singular value decomposition* (SVD)

$$SR^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \mathrm{diag}(\Sigma_1, \Sigma_2) \begin{bmatrix} V_1 & V_2 \end{bmatrix}^T$$

where

$$\Sigma_1 = \mathrm{diag}(\sigma_1, \ldots, \sigma_r), \quad \Sigma_2 = \mathrm{diag}(\sigma_{r+1}, \ldots, \sigma_n),$$

and $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \geq \sigma_n \geq 0$.

The so-called *square-root* (**SR**) methods determine L and T as [13]

$$L = \Sigma_1^{-1/2} V_1^T R, \qquad T = S^T U_1 \Sigma_1^{-1/2}.$$

If r is precisely the order of a minimal realization of G, then the gramians corresponding to the resulting realization are diagonal and equal. In this case the minimal realization is called *balanced*. The **SR** approach is usually very accurate for well-equilibrated systems. However if the original system is highly unbalanced, potential accuracy losses can be induced in the reduced model if either L or T is ill-conditioned.

In order to avoid ill-conditioned projections, a *balancing-free* (**BF**) approach has been proposed in [12] in which always well-conditioned matrices L and T can be determined. These matrices are computed from orthogonal matrices whose columns span orthogonal bases for the right and left eigenspaces of the product PQ corresponding to the first r largest eigenvalues $\sigma_1^2, \ldots, \sigma_r^2$. Because of the need to compute explicitly P and Q as well as their product, this approach is usually less accurate for moderately ill-balanced systems than the **SR** approach.

A *balancing-free square-root* (**BFSR**) algorithm which combines the advantages of the **BF** and **SR** approaches has been introduced in [15]. L and T are determined as

$$L = (Y^TX)^{-1}Y^T, \qquad T = X,$$

where X and Y are $n \times r$ matrices with orthogonal columns computed from the QR decompositions $S^TU_1 = XW$ and $R^TV_1 = YZ$, while W and Z are non-singular upper-triangular matrices. The accuracy of the **BFSR** algorithm is usually better than either of **SR** or **BF** approaches.

The SPA formulas can be used directly on a balanced minimal order realization of the original system computed with the **SR** method. A **BFSR** method to compute SPAs has been proposed in [14]. The matrices L and T are computed such that the system (LAT, LB, CT, D) is minimal and the product of corresponding gramians has a block-diagonal structure which allows the application of the SPA formulas.

The effectiveness of the **SR** or **BFSR** techniques depends entirely on the accuracy of the computed Cholesky factors of the gramians. Provided the Cholesky factors R and S are known, the computation of matrices L and T can be done by using exclusively numerically stable algorithms. In a next release of SLICOT we hope to perform the computation of the SVD of SR^T even without forming this product.

The **BFSR** version of the *balance & truncate* (B&T) method [11] is described in [15]. Its **SR** version [13] can be used to compute balanced minimal representations. Such representations are also useful for computing reduced order models by using the SPA formulas [10] or the *Hankel-norm approximation* (HNA) method [6]. A **BFSR** version of the SPA method is described in [14]. The B&T, SPA and HNA methods belong to the family of absolute error methods which try to minimize the absolute approximation error $\|G - G_r\|_\infty$. For all three method we have the guaranteed error bound [6, 10]

$$\|G - G_r\|_\infty \leq 2 \sum_{i=r+1}^{n} \sigma_i.$$

5.2 Model Reduction Software for Stable Systems

Both the **SR** and **BFSR** versions of the B&T and SPA have been standardized starting from the RASP-MODRED versions. The implementation of the HNA method relies also on a RASP-MODRED routine and uses the **SR** version of the B&T method to compute a balanced minimal realization of the original system. All standardized routines are applicable to both continuous- and discrete-time systems. It is worth mentioning that implementations provided in commercial software [4, 3, 1] are usually only for continuous-time systems. The following user callable main routines are presently available in SLICOT for model reduction of stable systems:

Name	Function
AB09AD	computes reduced (or minimal) order models using either the **SR** or the **BFSR** B&T method
AB09BD	computes reduced order models using the **SR** or the **BFSR** SPA method
AB09CD	computes reduced order models using the optimal HNA method based on **SR** balancing
AB09DD	computes a reduced order model by using the singular perturbation formulas

The main routines call several lower level routines, which solve particular problems, e.g. for systems with the A matrix in real Schur form. These routines will also serve for the standardization of model reduction software for unstable systems.

Andras Varga

References

[1] B.D.O Anderson and B. James. *MATRIX*$_{\mathrm{X}}$ *Model Reduction Module.* ISI, Santa Clara, CA, 1991.

[2] E. Anderson, Z. Bai, J. Bishop, J. Demmel, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK User's Guide, Second Edition.* SIAM, Philadelphia, 1995.

[3] G. Balas, J. Doyle, K. Glover, A. Packard, and R. Smith. *μ-Analysis and Synthesis Toolbox 2.0.* The MathWorks Inc., Natick, MA, 1993.

[4] R. Y. Chiang and M. G. Safonov. *Robust Control Toolbox 2.0.* The MathWorks Inc., Natick, MA, 1992.

[5] D. Enns. *Model Reduction for Control Systems Design.* PhD thesis, Dept. Aeronaut. Astronaut., Stanford Univ., Stanford, CA, 1984.

[6] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds. *Int. J. Control*, 39:1115–1193, 1974.

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

[8] G. A. Latham and B. D. O. Anderson. Frequency-weighted optimal Hankel norm approximation of stable transfer functions. *Systems & Control Lett.*, 5:229–236, 1985.

[9] Y. Liu and B. D. O. Anderson. Controller reduction via stable factorization and balancing. *Int. J. Control*, 44:507–531, 1986.

[10] Y. Liu and B. D. O. Anderson. Singular perturbation approximation of balanced systems. *Int. J. Control*, 50:1379–1405, 1989.

[11] B. C. Moore. Principal component analysis in linear system: controllability, observability and model reduction. *IEEE Trans. Autom. Control*, AC-26:17–32, 1981.

[12] M. G. Safonov and R. Y. Chiang. A Schur method for balanced-truncation model reduction. *IEEE Trans. Autom. Control*, AC-34:729–733, 1989.

[13] M. S. Tombs and I. Postlethwaite. Truncated balanced realization of a stable non-minimal state-space system. *Int. J. Control*, 46:1319–1330, 1987.

[14] A. Varga. Balancing-free square-root algorithm for computing singular perturbation approximations. In *Proc. of 30th IEEE CDC, Brighton, UK*, pp. 1062–1065, 1991.

[15] A. Varga. Efficient minimal realization procedure based on balancing. In A. El Moudni, P. Borne, and S. G. Tzafestas, editors, *Prepr. of IMACS Symp. on Modelling and Control of Technological Systems*, vol. 2, pp. 42–47, 1991.

[16] A. Varga. Numerical algorithms and software tools for analysis and modelling of descriptor systems. In *Prepr. of 2nd IFAC Workshop on System Structure and Control, Prague, Czechoslovakia*, pp. 392–395, 1992.

6 SLICOT tools for subspace identification

System identification is becoming increasingly important in optimising the quality, economical cost and environmental impact of products produced by various branches of industry, such as the process industry, the automobile and aerospace industry and consumer electronics.
The emergence of subspace identification techniques opens the possibility for significant improvements in identification in industry. This is because, on the one hand, these methods offer the computational advantages of widely accepted estimation schemes based on the linear least squares (LLS) method and the Fast Fourier Transform (FFT), such as: (1) the numerical robustness and efficiency of these schemes, and (2) their straightforward use without the requirement of getting accurate initial estimates of both parameter values and model structure and, on the other hand, they possess a number of industrially appealing properties. Three important ones are: (1) the direct identification of a MIMO Kalman filter from input-output data without initial knowledge of the system order and initial estimates of parameter values, (2) the application to classes of nonlinear systems, such as Wiener, Hammerstein and bilinear systems, while preserving their simplicity in use as holds for the linear time-invariant case, and (3) identification of accurate models based on short data length sequences.
The proposed standardisation and evaluation project aims at providing numerically robust and efficient software tools, included in SLICOT, Matlab and Scilab, for subspace identification. The toolboxes to be developed will include: A. identification of linear time-invariant systems (open and closed loop), and B. identification of nonlinear systems (in Wiener, Hammerstein and bilinear form).

In the first six months of the project an inventory is made of existing subspace identification methods that can be applied to identify linear, time-invariant systems operating in open-loop. Three routines which were publicly accessible were collected and subsequently tested on 15 real-life data sets. These data sets correspond to important representative and industrially relevant problems from DAISY[10], a database for identification of systems. The results are documented in a preliminary SLICOT working note 1998-6[11]. A detailed analysis of these tests allows to make a proposal for a new subspace identification routine that integrates the advantages of all three methods. The implementation of this integrated method will be the topic of the next six months.

Michel Verhaegen

[10]DAISY is a database for the identification of systems which is publicly available on the website `http://www.esat.kuleuven.ac.be/sista/daisy/`

[11]Report under preparation. This report will be available in October 1998 from `wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/SLWN1998-6.ps.Z`

7 SLICOT tools for robust control

Robust stability and robust performance are the most important issues in control systems design. Control systems are required to operate safely in the presence of parameter variations, external disturbances, measurement noise and unmodelled dynamics. The robust control problem is to design a controller such that the closed-loop system is stable and maintains specified performance levels in the presence of such uncertainties.

In the last two decades, there have been considerable developments in robust control theory and design methods. A variety of software packages, e.g., several toolboxes in Matlab, have been developed commercially. But the major beneficiaries so far have been academics. Because of low computational efficiency, poor numerical accuracy, and lack of ability to handle large scale problems, those software packages are not readily suitable for solving industrial design problems. The NICONET project thus aims at providing numerically robust and efficient software tools, included in SLICOT, to be used in the robust controller design.

The algorithms to be developed will include the standard state space LQG (H_2) optimal design, H_∞ sub-optimal and optimal solutions, H_∞ Loop Shaping Design Procedure (LSDP), μ-synthesis method, Linear Matrix Inequality (LMI) approach and robust pole-assignment algorithms. Numerical superiority would be an essential criterion in the selection of algorithms to perform the above methods. Information such as the condition and error estimation will be provided so that the user may be able to make a sensible decision between, for instance, the optimality of an H_∞ sub-optimal solution and the risk of large error which may be produced during the computational procedure. Demonstration examples and case studies will also be included similarly as in other NICONET topics.

The standard H_∞ sub-optimal routine will be available by the end of July, and H_∞ optimal and H_2 solution routines will be developed during the second half of 1998. Other functions will be ready in due time.

Da-wei Gu

8 SLICOT tools for nonlinear systems in robotics

An important part of the work that will be performed in NICONET is related to nonlinear systems. A task will be developed in NICONET focused on the development of standard software to allow the efficient simulation, identification and control design of nonlinear systems with main emphasis on robotic applications.

To develop standard software for nonlinear systems, the main effort will be to define generic computational tools for nonlinear systems and to develop appropriate interfaces to standard software for integration of differential equations, nonlinear programming and solving nonlinear equations. The main concern is to provide a uniform interface to a selection of routines that perform efficiently in most of the problems. Sequential and parallel tools will be included in this part of the work.

The interface to these tools will be FORTRAN-77 and Matlab compatible, which constitutes an efficient and easy framework for testing and developing codes for these problems.

The main application domain for testing and benchmarking nonlinear systems interfacing tools is robotics. Dynamic models of industrial robots will be used. Besides the application in robotics, the selected tools will be useful for most numerical problems dealing with nonlinear systems.

Vicente Hernandez

9 NICONET information corner

This section informs the reader on how to access the SLICOT library, the main product of the NICONET project, and how to retrieve its routines and documentation. In addition, information is provided on the newest NICONET reports, available via the NICONET website or ftp site, as well as information about upcoming workshops/conferences organized by NICONET or with a strong NICONET representation.

Additional information about the NICONET Thematic Network can be obtained from the NICONET homepage World Wide Web URL

```
http://www.win.tue.nl/wgs/niconet.html
```

9.1 Electronic Access to the SLICOT Library

The SLICOT routines can be downloaded from the WGS ftp site,

```
ftp://wgs.esat.kuleuven.ac.be
```

(directory `pub/WGS/SLICOT/` and its subdirectories) in compressed (gzipped) tar files. On line `.html` documentation files are also provided there. It is possible to browse through the documentation on the WGS homepage at the World Wide Web URL

```
http://www.win.tue.nl/wgs/
```

after linking from there to the SLICOT web page and clicking on the `FTP site` link in the freeware SLICOT section. The SLICOT index is operational there. Each functional "module" can be copied to the user's current directory, by clicking on an appropriate location in the `.html` image. A "module" is a compressed (gzipped) tar file, which includes the following files: source code for the main routine and its test program, test data, execution results, the associated `.html` file, as well as the source code for the called SLICOT routines. This involves duplicating some routines, but it can be convenient for a user needing only a single function. There is also a file, called `slicot.tar.gz`, in the directory `/pub/WGS/SLICOT/`, which contains the entire library. The tree structure created after applying

```
gzip -d slicot.tar
```

and

```
tar xvf slicot.tar
```

is:

```
./slicot/       - for routine source files;
./slicot/doc/   - for html files;
./slicot/tests/ - for test programs/data/results files.
```

The user can then browse through the documentation on his local machine, starting from the index file `libindex.html` from `./slicot` subdirectory.

9.2 SLICOT Library update in the period January – July 1998

Several routines were renamed (and, when possible, improved) according to the latest version of *SLICOT Implementation and Documentation Standards* (February, 1998), in order to refine the SLICOT Library Index Classification for Chapter T.

Several new user-callable routines for basic control problems have been made available on the ftp site. They include *Analysis Routines*, *Mathematical Routines*, *Synthesis Routines*, *Transformation Routines*, and *Utillity Routines*, performing the following tasks:

- model reduction for stable systems via square-root or balancing-free square-root Balance & Truncate methods;
- model reduction for stable systems via square-root or balancing-free square-root singular perturbation approximation methods;
- model reduction for stable systems via square-root or balancing-free square-root Balance & Truncate methods; optimal Hankel-norm approximation model reduction (with square-root balancing) for stable systems;
- compute a reduced order model using singular perturbation approximation formulas;
- reorder the eigenvalues of a real Schur matrix according to several reordering criteria;
- compute the block diagonal form of a square matrix;
- compute all, or part, of the singular value decomposition of an upper triangular matrix;
- solve for $X = \mathrm{op}(U)'\mathrm{op}(U)$ either the stable non-negative definite continuous-time Lyapunov equation $\mathrm{op}(A)^T X + X\mathrm{op}(A) = -\sigma^2 \mathrm{op}(B)^T \mathrm{op}(B)$, or the convergent non-negative definite discrete-time Lyapunov equation $\mathrm{op}(A)^T X \mathrm{op}(A) - X = -\sigma^2 \mathrm{op}(B)^T \mathrm{op}(B)$, where $\mathrm{op}(K) = K$ or K^T, A is square, B is rectangular, U is upper triangular, and σ is a scale factor, set less than or equal to 1 to avoid overflow in X;
- compute the terms G_1 and G_2 (in a state-space representation) of an additive spectral decomposition of a transfer-function matrix G with respect to a specified region of the complex plane;
- perform an orthogonal similarity transformation to reduce the system state matrix to an ordered real Schur form;
- perform an orthogonal similarity transformation to reduce the system state matrix to the real Schur form;
- read the coefficients of a matrix polynomial;
- read the elements of a sparse matrix polynomial;
- read the elements of a sparse matrix;
- print the coefficient matrices of a matrix polynomial.

Future changes in the library contents or routine updates—till the next SLICOT Release—are announced in the file `Release.Notes`, located in directory `pub/WGS/SLICOT/` on the WGS ftp site.

9.3 New NICONET Reports

Recent NICONET reports (available since June 1998), that can be downloaded as compressed postscript files from the World Wide Web URL

`http://www.win.tue.nl/wgs/reports.html`

or from the WGS ftp site,

`ftp://wgs.esat.kuleuven.ac.be`

(directory `pub/WGS/REPORTS/`), are the following:

- Andras Varga. *Task II.A.1—Selection of Model Reduction Routines* (file SLWN1998-2.ps.Z).

  This paper discusses the model reduction algorithms which are to be included in the SLICOT library as part of the NICONET project.

- Andras Varga. *Task I.A.1—Selection of Basic Software Tools for Standard and Generalized State-space Systems and Transfer Matrix Factorizations* (file SLWN1998-3.ps.Z).

  This paper discusses the algorithms and software for basic control problems, for the factorization of proper transfer function matrices, and for descriptor systems which are to be included in the SLICOT library as part of the NICONET project.

- Andras Varga. *Standardization of Interface for Nonlinear Systems Software in SLICOT* (file SLWN1998-4.ps.Z).

  This paper discusses the development of standardized Fortran interfaces for the description of nonlinear systems to allow an easy interfacing with standard software for integration of differential equations, nonlinear programming and solving nonlinear equations. These interfaces are used for nonlinear systems software in SLICOT and will be developed as part of the NICONET project.

- Volker Mehrmann and Thilo Penzl. *Benchmark collections in SLICOT* (file SLWN1998-5.ps.Z).

  This paper contains guidelines for setting up benchmark collections for SLICOT. The purpose of these SLICOT benchmark collections is to establish an environment for testing the subroutines within the SLICOT library and compare their performance with other numerical software. Guidelines for the submission of benchmark examples by external contributors are given, as well as guidelines for the implementation of benchmark routines within SLICOT.

- Vasile Sima, Peter Benner, Sabine Van Huffel and Andras Varga. *Improving the efficiency and accuracy of the MATLAB control toolbox using SLICOT-based gateways* (file MTNS98.ps.Z).

  The paper presents performance results for some components of the new, public-domain version of the SLICOT library in comparison with equivalent computations performed by some MATLAB functions included in the Control Toolbox. SLICOT incorporates the new algorithmic developments in numerical linear algebra, implemented in the state-of-the-art software packages LAPACK and BLAS. The results show that, at comparable or better accuracy, SLICOT routines are several times faster than MATLAB computations.

Previous NICONET/WGS reports are also posted at the same address.

9.4 Forthcoming Conferences

Forthcoming Conferences related to the NICONET areas of interest, where NICONET partners submitted or will submit proposals for NICONET/SLICOT-related talks and papers, and/or will diseminate information and promote SLICOT, are the following:

- Mathematical Theory of Networks and Systems, MTNS 98, Padova, Italy, July 6–10, 1998.
- NICONET Workshop[12], Valencia, December 4, 1998 (organized by NICONET, with external participation).This workshop is open to all end-users and industrial software companies. The workshop consists of 3 plenary talks, a demo and poster session, and a panel discussion.
- Computer-Aided Control System Design, CACSD '99, Hawaii, August 22 – 26, 1999.
- European Control Conference, Karlsruhe, Germany, 31 August – 3 September, 1999.

Vasile Sima

[12]For more detailed information and a preliminary program, see the NICONET website `http://www.win.tue.nl/wgs/niconet.html`