

NICONET Newsletter

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

Contents:	Page
1. Editorial	2
2. New developments in the SLICOT benchmark library	3
3. Basic numerical SLICOT tools for control	4
4. SLICOT tools for model reduction	8
5. SLICOT tools for subspace identification	13
6. SLICOT tools for robust control	14
7. SLICOT tools for nonlinear systems in robotics	15
8. SLICOT controls microwaves – a case study	18
9. SLICOT: a useful tool in industry?	24
10. Second NICONET workshop in INRIA-Rocquencourt, France	28
11. NICONET information corner	31

*Contact address of the WGS: Mrs. Ida Tassens, Secretary of WGS
 Katholieke Universiteit Leuven
 Dept. of Electrical Engineering (ESAT-SISTA/COSIC)
 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

©NICONET NEWSLETTER. Parts of this Newsletter may be reproduced,
 provided the source is mentioned.

1 Editorial

Welcome to the third issue of the NICONET newsletter which informs you about the evolution of the SLICOT library and its integration in user-friendly environments such as **Scilab** and **MATLAB**, as well as about other NICONET activities related to CACSD software developments. To assess the performance, reliability and versatility, of the new routines, we are developing benchmarks and select industrial design examples, which should reveal the numerical robustness and computational efficiency of the SLICOT software. More detailed information on the present status of the SLICOT tools is given in Sections 3 to 7. In addition, much attention is paid to the selection and development of industrially oriented benchmarks. The newest developments in the SLICOT benchmark library are described in Section 2.

To illustrate the industrial relevance of SLICOT, we describe in Sections 8 and 9 two different industrial examples in which SLICOT is being used and shown to improve the performance.

Finally, we would like to invite all of you to our second NICONET workshop which will be held at INRIA Rocquencourt near Paris-Versailles in France on December 3, 1999. This workshop includes plenary talks discussing the importance of numerics in control software in industry, demo sessions as well as a training course on the use of SLICOT and SLICOT based toolboxes. A detailed program, as well as registration information, is included in Section 10. Finally, Section 11 announces the publication of new NICONET reports and the new updates added to the SLICOT library in the last 6 months.

Sabine Van Huffel

Chairperson of WGS and Coordinator of NICONET.

2 New developments in the SLICOT benchmark library

Two new benchmark collections have been added to the SLICOT benchmark library. These concern continuous-time and discrete-time Lyapunov equations. The routines are BB03AD (continuous-time) and BB04AD (discrete-time) and the documentation is given in SLICOT working notes:

References

- [1] D. Kressner, V. Mehrmann und T. Penzl, ‘CTLEX– a Collection of Benchmark Examples for Continuous-Time Lyapunov Equations’, *SLICOT Working Note SLWN1999-6*, see <http://www.esat.kuleuven.ac.be/pub/WGS/REPORTS/>
- [2] D. Kressner, V. Mehrmann und T. Penzl, ‘DTLEX– a Collection of Benchmark Examples for Discrete-Time Lyapunov Equations’, *SLICOT Working Note SLWN1999-7*, see <http://www.esat.kuleuven.ac.be/pub/WGS/REPORTS/>

Volker Mehrmann and Thilo Penzl

3 Basic Numerical SLICOT Tools for Control

Developing basic software tools for standard and generalised state space systems and transfer matrix factorisations is required for most other NICONET activities since they rely on it to provide them with basic building blocks. The selection, standardization and documentation of basic routines has been completed as originally planned. Care has been put in selecting numerical methods that are state of the art : the numerical reliability, efficiency and flexibility of the selected routines have indeed been demonstrated in the literature. It soon appeared to be useful to even extend this activity beyond the original plans, and the routines completed at this stage also include routines that were not in the originally proposed list. These were included because of their relevance to other tasks of our network.

The following 37 user-callable routines have been standardized in the first 18 months :

Mathematical Routines

Name	Function
MB01TD	computes the product of two upper quasi-triangular matrices
MB01UD	computes the matrix expressions $\alpha \text{op}(H)A$ or $\alpha A \text{op}(H)$, with H an upper Hessenberg matrix ($\text{op}(A) = A$ or A^T)
MB02PD	solves the matrix equation $\text{op}(A)X = B$, providing error bounds and condition estimates
MB02QD	solves a linear least squares problem, optionally considering specified free elements
MB03QD	reorders the eigenvalues of a real Schur matrix according to several reordering criteria
MB03RD	computes the block diagonal form of a square matrix
MB03SD	computes the eigenvalues of a square-reduced Hamiltonian matrix
MB03UD	computes all, or part, of the singular value decomposition of an upper triangular matrix
MB03VD	computes the periodic Hessenberg decomposition of a matrix product
MB03WD	computes the periodic Schur decomposition and eigenvalues of a matrix product in a periodic Hessenberg form

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 transformation to reduce the system state matrix to an ordered real Schur form
TB01UD	computes the orthogonal controllability staircase form of a state-space model
TB01RD	computes the orthogonal observability staircase form of a state-space model
TB01WD	performs an orthogonal similarity transformation to reduce the system state matrix to the real Schur form
TB01ZD	computes a controllable realization for single-input systems

Analysis Routines

Name	Function
AB13AD	computes the Hankel norm and the Hankel singular values of the stable projection of a transfer-function matrix
AB13BD	computes the L_2 - or l_2 -norm of a transfer-function matrix
AB13CD	Computation of the \mathcal{H}_∞ norm of a continuous-time system

Synthesis Routines

Name	Function
SB01BD	performs multi-input pole assignment using the Schur method
SB01DD	performs parametric multi-input pole assignment using the Hessenberg method
SB02QD	condition and forward error estimation for continuous-time Riccati equation (CARE) solution
SB02RD	solves the CARE or DARE using refined Schur vectors method, with condition and forward error bound estimates
SB02SD	condition and forward error estimation for discrete-time Riccati equation (DARE) solution
SB03OD	solves for $X = \text{op}(U)' \text{op}(U)$ either the stable non-negative definite continuous-time Lyapunov equation $\text{op}(A)' X + X \text{op}(A) = -\sigma^2 \text{op}(B)' \text{op}(B)$ or the convergent non-negative definite discrete-time Lyapunov equation $\text{op}(A)' X \text{op}(A) - X = -\sigma^2 \text{op}(B)' \text{op}(B)$, where $\text{op}(K) = K$ or K'
SB03PD	Solution of the discrete-time matrix algebraic Lyapunov equations with condition and accuracy estimates
SB03QD	condition and forward error estimation for continuous-time Lyapunov equation solution
SB03SD	condition and forward error estimation for discrete-time Lyapunov equation solution
SB03TD	solves continuous-time Lyapunov equation, with condition and forward error estimation
SB03UD	solves discrete-time Lyapunov equation, with condition and forward error estimation

Factorization Routines

Name	Function
SB08CD	computes the state-space representations of the factors of a LCFID of a TFM
SB08DD	computes the state-space representations of the factors of a RCFID of a TFM
SB08ED	computes the state-space representations of the factors of a LCF with prescribed stability degree
SB08FD	computes the state-space representations of the factors of a RCF with prescribed stability degree
SB08GD	computes the state-space representation of the TFM corresponding to a LCF
SB08HD	computes the state-space representation of the TFM corresponding to a RCF

Descriptor System Routines

Besides the above (extended) list of routines, a whole new section of generalised state space system routines has been added to this activity. These routines were added because of their relevance to other tasks of the NICONET project. The following 9 user-callable routines have therefore been implemented and standardized in this context and will be added soon to SLICOT library :

Transformation Routines

Name	Function
TG01AD	performs the scaling of a descriptor system model
TG01CD	reduces a descriptor system to the QR-coordinate form with $\tilde{E} = QE$ upper triangular and $Z = I$
TG01DD	reduces a descriptor system to the RQ-coordinate form with $\tilde{E} = EZ$ upper triangular and $Q = I$
TG01ED	reduces a descriptor system to the singular value coordinate form with $\tilde{E} = QEZ$ diagonal, having its singular values as diagonal elements
TG01FD	reduces a descriptor system to a singular value like coordinate form with $\tilde{E} = QEZ$ in a complete orthogonal decomposition form
TG01HD	computes the controllability staircase form of a descriptor system
TG01ID	computes the observability staircase form of a descriptor system
TG01JD	computes an irreducible descriptor representation from a non-minimal one

Analysis Routines

Name	Function
AG08BD	computes the zeros and the Kronecker structure of a descriptor system pencil

A large number of auxiliary routines have also been written, standardized and documented. Although these routines are not user callable, they are very valuable and can still be called in their own right. For this reason, the same documentation standards were followed as for user-callable routines.

These basic software tools are more “accessible” when implemented in a user-friendly environment, so that little technical background is required to use the tools to almost full functionality. For this reason a set of *mex*-function and *m*-functions were developed to access these routines in a user friendly manner. Since such *mex*-files are rather big, we minimize their number by grouping routines which require similar basic routines into one *mex*-file with multiple functionality (several *m*-files will call these *mex*-files). These easy-to-use interfaces are provided in MATLAB as well as in Scilab.

Mex-files and m-files

Name	Function
<code>linmeq</code>	mex-file used for the following m-files
<code>sllyap</code>	for general Lyapunov equations
<code>slstly</code>	for stable semi-definite Lyapunov equations
<code>slstei</code>	for general Stein equations
<code>slstst</code>	for stable semi-definite Stein equations
<code>slsylv</code>	for Sylvester equations
<code>genleq</code>	mex-file used for the following m-files
<code>slgely</code>	for generalized Lyapunov equations
<code>slgsly</code>	for generalized stable semi-definite Lyapunov equations
<code>slgest</code>	for generalized Stein equations
<code>slgsst</code>	for generalized stable positive Stein equations
<code>slgesg</code>	for generalized Sylvester equations
<code>aresol</code>	mex-file used for the following m-files
<code>slcares</code>	for continuous-time algebraic Riccati equations
<code>sldares</code>	for discrete-time algebraic Riccati equations
<code>slcaregs</code>	for generalised continuous-time algebraic Riccati equations
<code>sldaregs</code>	for generalised discrete-time algebraic Riccati equations
<code>sldaregsv</code>	for generalised discrete-time algebraic Riccati equations
<code>syscom</code>	mex-file used for the following m-files
<code>slconf</code>	for the controllability form of a state space system
<code>slobsf</code>	for the observability form of a state space system
<code>slminr</code>	for the minimal realization of a state space system
<code>systra</code>	mex-file used for the following m-files
<code>slsbal</code>	for balancing a state space system
<code>slsrsf</code>	for the real Schur form of a state space system
<code>slsorsf</code>	for the ordered real Schur form of a state space system
<code>slsdec</code>	for the additive spectral decomposition of a state space system

Preliminary versions of these routines have been implemented and are available for the moment. The first two *mex*-functions and the associated *m*-functions have been posted on the WGS ftp site.

Paul Van Dooren and Andras Varga

4 SLICOT tools for model reduction

The basic model reduction algorithms implemented and standardized in SLICOT belong to the class of methods based on or related to balancing techniques and are intended to solve model reduction problems for stable or unstable, continuous- or discrete-time linear systems. The selected routines for standardization¹ come partly from the RASP-MODRED library of the DLR² and partly are new implementations. All implemented methods rely on guaranteed mathematical error bounds and have particular features which recommend them for use in specific applications. The underlying computational algorithms represent the latest developments of various procedures for solving computational problems appearing in the context of model reduction.

At this moment, the standardization of routines for model reduction has been terminated. The Model Reduction Toolbox contains a total of 9 user-callable routines standardized for reduction of stable and unstable systems:

Name	Function
AB09AD	computes reduced (or minimal) order balanced models using either the SR or the BFSR B&T method
AB09BD	computes reduced order models using 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
AB09ED	computes reduced order models for unstable systems using the optimal HNA method in conjunction with additive stable/unstable spectral decomposition
AB09FD	computes reduced order models for unstable systems using the BFSR B&T method in conjunction with left/right coprime factorization methods
AB09GD	computes reduced order models for unstable systems using the BFSR SPA method in conjunction with left/right coprime factorization methods
AB09MD	computes reduced order models for unstable systems using the B & T method in conjunction with additive stable/unstable spectral decomposition
AB09ND	computes reduced order models for unstable systems using the SPA method in conjunction with stable/unstable additive spectral decomposition

All these routines perform optionally scaling of the initial system. Each of routines handles both continuous-time as well as discrete-time systems. An important number of user-callable and auxiliary routines have been implemented for the special needs of the model reduction. These routines have been standardized as part of Task I.A.

One of the main objectives of the NICONET project is to provide, additionally to standardized Fortran codes, high quality software embedded into user-friendly environments for *computer aided control system design* (CACSD). Two target CACSD environments have been envisaged: the popular commercial numerical computational environment MATLAB and the public domain MATLAB-like environment Scilab. Both allows to easily add external functions implemented in general purpose programming languages like C/C++ or Fortran. In case of

¹Report available by anonymous ftp from [wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/SLWN1998-2.ps.Z](ftp://wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/SLWN1998-2.ps.Z)

²German Aerospace Center (DLR), Oberpfaffenhofen

MATLAB, the external functions are called *mex*-functions and have to be programmed according to precise programming standards. In **Scilab**, external functions can be similarly implemented and only several minor modifications were necessary to the MATLAB *mex*-functions to use them in **Scilab**.

A unique *mex*-function **sysred** provides interface to all model reduction routines in SLICOT. To provide a convenient interface to work with control objects defined in the MATLAB Control Toolbox, several easy-to-use interface functions have been additionally implemented explicitly addressing some of supported features. The following table contains the list of implemented *m*-functions. Similar functions will be available also within **Scilab**.

Name	Function
bta	for balancing-free square-root B & T method in combination with stable/unstable additive spectral decomposition
btabal	for square-root B & T method in combination with stable/unstable additive spectral decomposition
bta_cf	for balancing-free square-root B & T method in combination with stable left/right coprime factorizations
btabal_cf	for square-root B & T method in combination with stable left/right coprime factorizations
spa	for balancing-free square-root SPA in combination with stable/unstable additive spectral decomposition
spabal	for square-root SPA in combination with stable/unstable additive spectral decomposition
spa_cf	for balancing-free square-root SPA in combination with stable left/right coprime factorizations
spabal_cf	for square-root Singular Perturbation Approximation in combination with stable left/right coprime factorizations
hna	for HNA in combination with stable/unstable additive spectral decomposition

Extensive testing of the implemented software has been performed using several benchmark problems. The benchmark problems for model reduction include industrial models taken from control literature as well as real examples, as for example, the linearized model of an aircraft (ATTAS), the finite element model of a CD-player or several linearized models of an industrial gasifier plant. Model reduction results on these models are presented in a separate report³.

In what follows we present some results computed for a continuous-time linearized state space model PS of a two-area interconnected power system. The model has the form

$$\begin{aligned}\dot{x} &= Ax + B_1u + B_2w \\ y &= Cx\end{aligned}$$

where $x \in \mathbb{R}^7$ is the state vector, $u \in \mathbb{R}^2$ is the command input vector, $w \in \mathbb{R}^2$ is the disturbance input vector and $y \in \mathbb{R}^3$ is the measurable output vector. Note that the partial model (A, B_1, C) has been used as test example by several SLICOT test programs of the

³Report available by anonymous ftp from wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/nic1999-8.ps.Z

model reduction routines for stable systems. The matrices of this model are

$$A = \begin{bmatrix} -0.04165 & 0 & 4.92 & -4.92 & 0 & 0 & 0 \\ -5.21 & -12.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.33 & -3.33 & 0 & 0 & 0 & 0 \\ 0.545 & 0 & 0 & 0 & -0.545 & 0 & 0 \\ 0 & 0 & 0 & 4.92 & -0.04165 & 0 & 4.92 \\ 0 & 0 & 0 & 0 & -5.21 & -12.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.33 & -3.33 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 & 0 \\ 12.5 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 12.5 \\ 0 & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} -4.92 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -4.92 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad D = 0.$$

The PS model is stable, minimal and has the Hankel-singular values

$$\{3.9137, 3.5944, 2.5277, 1.0888, 0.6526, 0.0276, 0.0275\}.$$

Taking into account the gap between the 5-th and 6-th singular values, a 5-th order model seems to be appropriate for a lower order approximation. The B&T, SPA and HNA methods produced reduced order models of order 5, PS₁, PS₂ and PS₃, respectively, which approximately preserve the dominant poles of the original system

Poles of PS	Poles of PS ₁	Poles of PS ₂	Poles of PS ₃
$-0.5181 + 3.1259i$	$-0.5053 + 3.1206i$	$-0.5186 + 3.1228i$	$-0.5118 + 3.1221i$
$-0.5181 - 3.1259i$	$-0.5053 - 3.1206i$	$-0.5186 - 3.1228i$	$-0.5118 - 3.1221i$
$-1.3550 + 2.1866i$	$-1.2923 + 2.1162i$	$-1.3598 + 2.1692i$	$-1.3250 + 2.1430i$
$-1.3550 - 2.1866i$	$-1.2923 - 2.1162i$	$-1.3598 - 2.1692i$	$-1.3250 - 2.1430i$
-1.6916	-1.4233	-1.6578	-1.5351
-13.1438			
-13.1617			

However, the three methods approximate differently the zeros of the original system, the B&T and SPA methods producing even non-minimum phase zeros:

Zeros of PS	Zeros of PS ₁	Zeros of PS ₂	Zeros of PS ₃
∞	1.4438	1.4438	-9.2069
∞	∞	0	∞
	∞	∞	∞
	∞		

There is little difference in step responses for different input-output channels and also the Nyquist plots show good agreements. In Figures 1 and 2 the corresponding plots for the u_1 - y_1 channel are presented. Each of the computed 5-th order approximate models is suitable to perform controller synthesis.

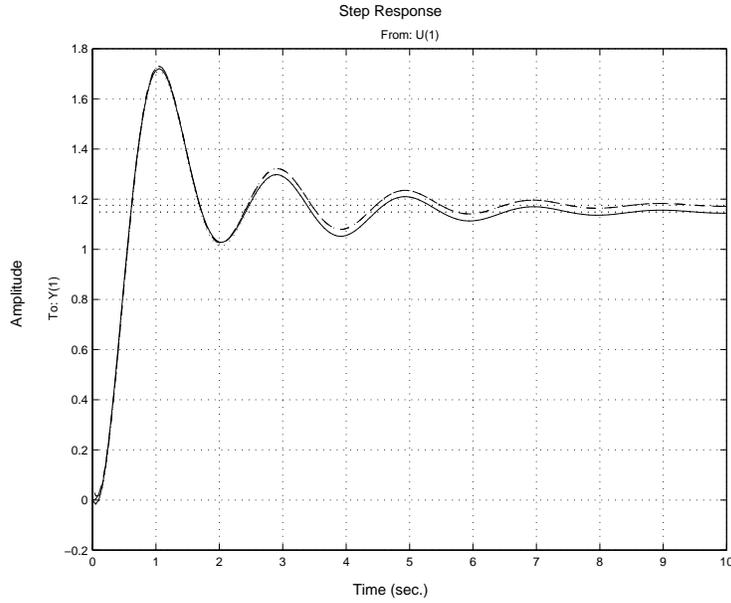


Figure 1: Comparison of step responses of element $g_{11}(s)$ of PS.

The already mentioned report also presents a thorough comparison of performance and capabilities of the SLICOT model reduction tools with those available in other packages. A summary of comparisons of available model reduction tools is presented in the following table:

Software	<i>RASP/ANDECS</i>	<i>SLICOT/Scalab</i>	<i>Control Toolbox</i>	<i>Robust Toolbox</i>	<i>μ-Toolbox</i>	<i>HTOOLS</i>	<i>MatrixX</i>	<i>WOR-Toolbox</i>
Provided features								
continuous-time	+	+	+	+	+	+	+	+
discrete-time	+	+	+	-	-	+	-	-
unstable	+	+	-	+	-	-	-	+
non-minimal	+	+	-	+	+	+	+	+
Method								
balancing	+	+	+	+	+	+	+	+
balancing-free (BF)	+	+	-	+	-	+	+	-
square-root (SR)	+	+	-	-	+	+	-	+
BF-SR	+	+	-	-	-	+	-	-
Problem classes								
additive error	+	+	+	+	+	+	+	+
relative error	+	-	-	+	+	+	+	-
frequency weighted	+	-	-	-	+	-	+	+

It can be seen from this table that the new standardized model reduction routines of SLICOT cover a broader area of applications than those available in the model reduction tools of

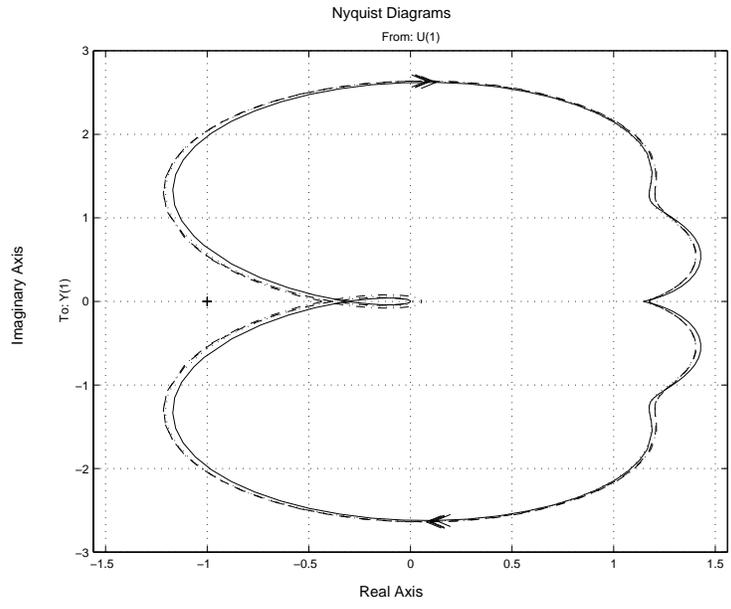


Figure 2: Comparison of frequency responses for element $g_{11}(s)$ of PS.

commercial packages. This functional richness is doubled by a high computational efficiency and numerical robustness.

Andras Varga

5 SLICOT tools for subspace identification

A report⁴ has been finished (with report number nic1999-3) documenting the modular definition and implementation in the MATLAB language of the basic building blocks of the 3 existing publicly available Subspace Identification routines known under the acronyms N4SID (developed in the group of Prof. B. De Moor (KU Leuven)), CVA (developed in the group of Prof. M. Deistler (TU Wien)), MOESP (developed in the group of Prof. M. Verhaegen (TU Delft)). The report highlighted the substantial savings in computational complexity by newly defined combinations of these basic building blocks. In addition it was recommended to exploit the block-Hankel structure of the matrices in the RQ factorization to further speed up the calculations.

In the future planning the following is foreseen:

1. A β -version of the Fortran library with subspace identification routines will be finalized by **Sept. 1st 1999**.
2. The documentation of the benchmark examples used in the report nic1999-3 will be finalized by **Nov. 1st 1999**.
3. By the end of the year 1999 a preliminary test period evaluating the novel subspace identification software following the recommendation given in the report nic1999-3 is planned. The partners that initially will take part in this evaluation are LMS, IPCOS, INRIA-Rocquencourt, TU-Chemnitz, TU-Eindhoven and TU-Delft.

Michel Verhaegen

⁴Report available by anonymous ftp from `wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/nic1999-3.ps.Z`

6 SLICOT tools for robust control

The SLICOT robust control section has now included subroutines for the H_2 and H_∞ controller synthesis in the discrete-time case as well as subroutines for multi-input systems pole assignment, \mathcal{H}_∞ norm calculation, etc.

The following tables summarise the user-callable routines developed so far concerning the above tasks (the routines AB13CD, SB03PD, SB03SD and SB01DD are already mentioned in section 3 and are therefore not repeated here):

Name	Function
SB10ED	Driver subroutine for discrete-time \mathcal{H}_2 optimization
SB10SD	Computation of the \mathcal{H}_2 optimal controller for the standard system
SB10TD	Computation of the \mathcal{H}_2 optimal controller for the general system
SB10DD	Computation of the \mathcal{H}_∞ sub-optimal controller

In addition, the following gateway function routines have been developed.

Name	Function
DHINFS	Design of the central \mathcal{H}_∞ sub-optimal controller for a discrete-time system
DH2	Design of the \mathcal{H}_2 optimal controller for a discrete-time system
HFNORM	Computation of the \mathcal{H}_∞ norm of a continuous-time system
DEIGEN	Pole assignment of a multi-input system

This software will be available on the WGS web and ftp site in September 1999.

Da-Wei Gu

7 SLICOT tools for nonlinear systems in robotics

7.1 Tools for nonlinear dynamical systems simulation

One of the objectives of the NICONET project is to include tools for nonlinear control systems into the SLICOT numerical software library. To achieve this goal, a standard interface to the most commonly used integrator packages (ODEPACK, DASSL, DASPK, RADAU5, DGELDA) is needed.

Problems dealing with the simulation of nonlinear control systems can be described in terms of *ordinary differential equations* (ODEs):

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

or *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.

An interface has been designed in which the user can specify all the information required by any of the packages including also the user-callable functions, which have been also standardised. This interface has the advantages that it requires very few modifications for changing the solver, and also it provides a reliable tool for the comparison of the different solvers. The interface translates the syntax and values of all parameters to the appropriate features of each package.

The implementation of the interface for ODEs is being finished and the interface for DAEs is under development. Although they are still subject to changes, the syntax of the driver ODE solver is:

```
SUBROUTINE ODESolver (ISOLVER,
&                      NRP, NIP, RPAR, IPAR,
&                      NX, NY, NU, NP, X, Y, U, P
&                      TINI, TOUT,
&                      LTOL, RTOL, ATOL,
&                      LDWORK, DWORK, LIWORK, IWORK,
&                      IERR )

SUBROUTINE ODEDER ( NRP, NIP, RPAR, IPAR,
&                  NX, NU, NP, X, U, P, T, F, LDF, IERR)
SUBROUTINE ODEOUT ( NRP, NIP, RPAR, IPAR,
&                  NX, NY, NU, NP, X, XDOT, Y, U, P, T,
&                  IERR)
SUBROUTINE JACFX ( NRP, NIP, RPAR, IPAR,
&                 NX, NU, NP, X, U, P, T, J, LDJ, IERR)
SUBROUTINE JACFU ( NRP, NIP, RPAR, IPAR,
&                 NX, NU, NP, X, U, P, T, F, LDJ, IERR)
SUBROUTINE JACFP ( NRP, NIP, RPAR, IPAR,
&                 NX, NU, NP, X, U, P, T, F, LDJ, IERR)
```

The parameters describe the solver identifier, the mode real and integer parameter vectors and their lengths, the state, input, output and problem parameter arrays and their sizes, time interval, tolerances, work arrays and error codes. The user must also provide subroutines that define the main function (f) of the system and the output (g or y) if different. Functions for computing the jacobians are also requested for improving the performance of most packages. Not all the features are supported by all the packages. Similarly, an interface for DAE problems has been defined.

The interface will be available as a linkable library (coded in Fortran, but available to all languages) and as executable modules for MATLAB and Scilab. The interface for MATLAB and Scilab is also very similar but simpler, since some information can be automatically obtained from these environments. The syntax of the MATLAB/ Scilab call is:

```

SUBROUTINE ODESolver (ISOLVER, CODEDER_, CODEOUT_,
&                   CJACFX_, CJACFU_, CJACFP_,
&                   RPAR, IPAR,
&                   X, Y, U, P
&                   TINI, TOUT,
&                   RTOL, ATOL,
&                   DWORK, IWORK,
&                   IERR )

SUBROUTINE ODEDER ( RPAR, IPAR, X, U, P, T, F, IERR)
SUBROUTINE ODEOUT ( RPAR, IPAR, X, XDOT, Y, U, P, T, IERR)
SUBROUTINE JACFX ( RPAR, IPAR, X, U, P, T, J, IERR)
SUBROUTINE JACFU ( RPAR, IPAR, X, U, P, T, F, IERR)
SUBROUTINE JACFP ( RPAR, IPAR, X, U, P, T, F, IERR)

```

in which the parameters have the same meaning.

With this interface, SLICOT can be used also to simulate nonlinear control problems. The advantage is that the users can continue with their favourite packages but also can benefit of the features of other packages. With this interface, SLICOT will be complemented with implicit and explicit methods for sparse, banded and dense storage modes exploiting different problem formulations, offering a wider range of possibilities than single packages do.

7.2 Tools for nonlinear parameter identification

An evaluation of existing optimization tools for the parameter identification of multi-body dynamical systems lead to the threefold observation:

1. The existence of a number of dedicated software packages with user-friendly command interfaces for the purpose of parameter identification of mechatronic systems. An Example is the EASY-FIT commercial software package for parameter estimation from Prof. K. Schittkowski of the university of Bayreuth, Germany.
2. The existence of a vast number of dedicated software packages for the purpose of solving parameter optimization problems.
3. The substantial requirement necessary to transform a multi-body parameter identification problem into a 'standard' parameter optimization problem.

Since most of the software packages are commercial they cannot be integrated into SLICOT and furthermore because of the restricted budget in NICONET no performance evaluation of these packages can be made. This in combination with the third point of the list above motivated the decision to develop by the end of the year 1999 a simple standard interface for the parameter identification of a multi-body system of a prespecified complexity making use of different nonlinear least squares solvers that enable to take linear and nonlinear inequality constraints into account. The standardization will be defined for both commercial software packages, such as NAG, and free-ware software packages to allow to make a comparison later on in the study for the optimization part only.

Ignacio Blanquer-Espert, Vicente Hernandez, Michel Verhaegen

8 SLICOT Controls Microwaves – a Case Study

Motivation

Recent technological advances in microwave household appliances have created the possibility of a more refined control of the heating process in combination ovens. This includes the control of both a microwave heating source and a forced convection (air) heating source. Optimal control of a combination oven will give a higher quality end-product with a more uniform temperature distribution and, hence, a better cook-quality. This short article reports on the application of the (recently available) SLICOT reduction routines for calculation of an optimal control history for a microwave combination oven. We present a simple, finite horizon linear quadratic regulator solution to the heating problem, which was calculated using the reduced set of model equations. The example involves the calculation of an optimal heating profile for a container of mashed potato. The original (high dimensional) system model was reduced using the SLICOT routine AB09CD, which includes optimal Hankel norm approximation with square-root balancing. This routine allows a substantial reduction of computation time for the LQ design. The reduction routines were found to be more efficient than MATLAB routines. Recent progress is reported.

Fourier's Law of Heat Diffusion

A simplified model of heat conduction can be described by Fourier's heat equation which, in Cartesian coordinates $\underline{r} = (x, y, z)$, reads

$$\rho(\underline{r})c(\underline{r})\frac{\partial T(\underline{r}, t)}{\partial t} = \vec{\nabla}\{k(\underline{r}) \cdot \vec{\nabla}T(\underline{r}, t)\} + Q(\underline{r}, t) \quad (3)$$

where $T(\underline{r}, t)$ is the temperature field ($^{\circ}C$), $Q(\underline{r}, t)$ is the heat flux which drives the system to a desired state or final temperature field, $\rho(\underline{r})$ is the density of the body to be heated, and $c(\underline{r})$ is the heat capacity. We note that in reality heat conductance and heat conductivity are temperature dependent and, hence, the above model is only a good approximation of the heat diffusion process on a restricted temperature domain. Further, note that the heat flux $Q(\underline{r}, t)$ in the heat diffusion PDE consists of two heating sources, namely a convection boundary condition for nodal points $r_i \in \partial\Omega$ and internal heating through microwaves for the points $r_i \in \Omega$. For boundary points the surface boundary condition expresses the heat transfer from the air to the object, i.e.

$$\forall \underline{r} \in \partial\Omega : \quad -k(\underline{r})\frac{\partial T(\underline{r}, t)}{\partial n} = h_q(T(\underline{r}, t) - T_{\infty}(t)) \quad (4)$$

where h_q is the surface heat transfer coefficient, n is the (outwards orientated) normal vector to the surface $\partial\Omega$, and $T_{\infty}(t)$ is the air temperature in the oven cavity.

The temperature field $T(\underline{r}, t)$ is approximated using finite element (Galerkin) approximation techniques which transform the system model to a set of ordinary differential equations. Assuming spatial uniformity of densities, heat capacities, and conductivities, this gives

$$C_p\frac{dx(t)}{dt} + Kx(t) = f(\underline{r}) \begin{pmatrix} P(t) \\ T_{\infty}(t) \end{pmatrix} \quad (5)$$

with C_p the heat capacitance matrix ($N \times N$), K the conductivity matrix ($N \times N$), $f(\underline{r})$ the heat input vector ($N \times m$), and $P(t)$ and $T_\infty(t)$ are the μ -wave power field and the air temperature in the oven cavity respectively. The N nodal temperatures $x_i(t)$ are defined on a grid κ . The dimension of the input vector (m) equals two in this case. Note that the driving term on the right hand side of (5) has been separated into a spatial dependent and a temporal dependent component.

SLICOT Reduction

The above set of ordinary differential equations in the *nodal temperatures* $x(t)$ on a grid κ is, in fact, the starting point for the control analysis. The dimension N of the nodal temperature vector can be quite large (order of magnitude is several thousands of nodes for a realistic case study) and the system matrices C_p and K are *sparse*. Calculation of the optimal, finite horizon, linear quadratic regulator for the full system is very laborious and involves too many equations. Hence, SLICOT's routine AB09CD was used to reduce the original set of equations (5) to a more workable size, i.e.

$$\dot{x}_r(t) = A_r x_r(t) + B_r u(t) \tag{6}$$

$$y(t_k) = C_r x_r(t_k) \tag{7}$$

with $\dim(x_r) \ll \dim(x)$. It was found that the efficiency of the AB09CD routine outperforms similar reduction commands in MATLAB, especially for large dimensional problems with several thousands of nodes. A comparison (expressed in CPU time) was made between a reduction using MATLAB's command 'ohklmr' and the SLICOT routine AB09CD. It was found that, for this particular problem, SLICOT performed 18 times faster (41.5 seconds, instead of 750.3 seconds for MATLAB's ohklmr). Also, the memory usage for the mashed potato problem (see below) was approximately a fourth of the size MATLAB used to solve the same problem. In order to find an optimal control history, the original quadratic cost function in the nodal temperatures $x(t)$ was transformed to a cost in the reduced state vector x_r and the associated algebraic Riccati equation in the reduced state was simply swept back to t_0 .

An Example Case Study

As an example case study the optimal profiles for a cylindrical container (figure 3) of mashed potato will be presented. First, the matrices C_p , K , and the input heating vector $f(\underline{r})$ were calculated with Galerkin finite element approximation techniques on the basis of the physical properties of mashed potato⁵. The microwave power distribution was approximated using Lambert's law which states that the attenuation of microwave power is exponentially decreasing in space (see figure 4 for the microwave distribution on the surface Ω_1) with an attenuation factor β . The Lambert approximation is not a very good one for realistic case studies but it demonstrates the method used here. We are currently working on examples which include more realistic values for the microwave power distribution, based on (finite difference) simulations of Maxwell's electro-dynamical laws in three dimensions.

The original system for the mashed potato problem involves 441 ordinary differential equations. SLICOT's AB09CD routine reduced this set to 41 states which substantially reduced the computations involved in a backward sweep of the algebraic Riccati equations for

⁵These matrices will be available on the SLICOT website.

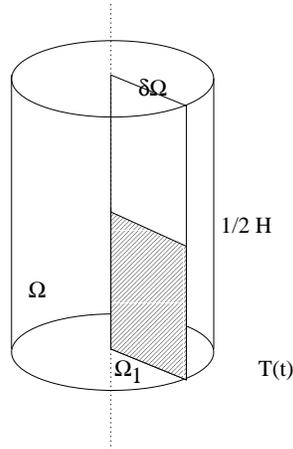


Figure 3: Cylindrical container of mashed potato. The simulations in this study involve the half vertical squared surface Ω_1 .

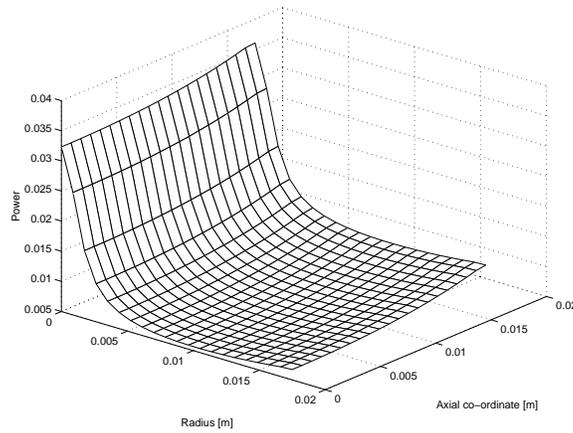


Figure 4: Microwave Power Distribution following Lambert's attenuation law.

calculation of the optimal control history $\{u(t), 0 < t < t_f\}$. It was found that the calculation of an optimal control history using SLICOT's reduced system model took approximately 2000 times less CPU time when compared to a similar calculation for the full 441 dimensional model. This, of course, is solely due to the model reduction. The calculated control inputs are shown in figure 5. Note that, for this specific case study, the microwave power controls the heating process almost completely. The fairly rapid decrease just before the final time instant $t_f = 90s$ is to allow heat diffusion from the center of the cylinder to the boundaries (figure 6).

Although this may not seem plausible on the basis of Lambert's law, the power distribution of the microwaves *decreases* from $r = 0$ to $r = R$ since the circular boundary introduces a focussing effect of microwave power in the center of the cylinder causing a rapid increase in temperature in the center. Indeed, this example demonstrates that the geometry of the problem contributes substantially to the heating process. An interesting additional exercise

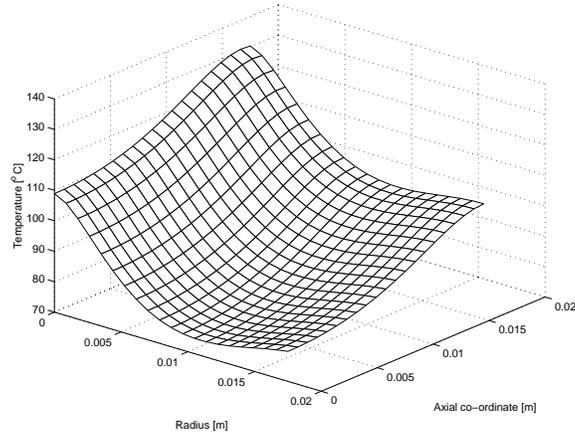


Figure 5: Final temperatures on the surface Ω_1 after 90 seconds of optimally combined hot-air/microwave heating.

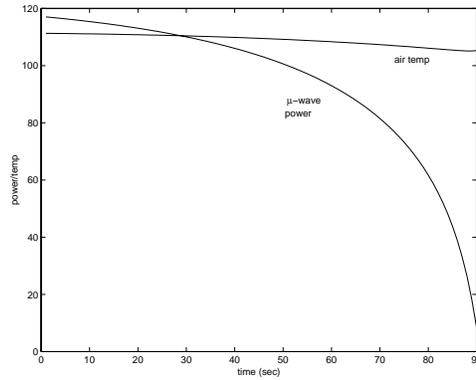


Figure 6: Optimal input profiles for combined μ -wave/forced convection heating for the mashed potato example.

is to calculate an optimal heating profile for convective control only. This was performed for the mashed potato example using a heating time of approximately 25 minutes. For this case SLICOT reduced the system to 29 states. The resulting input heating profile shows a ΔT -cooking profile, meaning that initially the heating temperature needs to be increased to maintain a constant gradient between the center and the boundary of the potato (figure 7) in order to transfer the boundary heat input most efficiently. The oscillation after the initial ΔT profile is to ‘fine-tune’ the temperature to reach exactly a uniform final temperature field of $100\text{ }^{\circ}\text{C}$. In figure 8 one can see that uniformity is reached within a range of approximately $0.1\text{ }^{\circ}\text{C}$ which is much better than the case of combined μ -wave/forced convection heating.

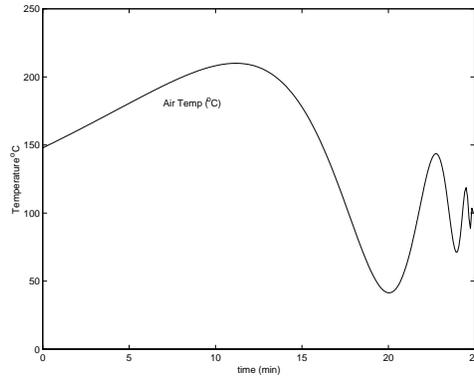


Figure 7: Optimal input profiles for convection heating only.

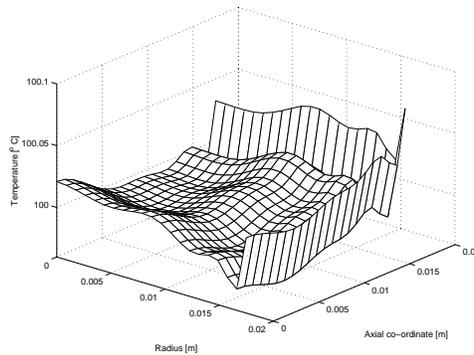


Figure 8: Final temperature distribution for convection heating only.

Closing Remarks

This short note has demonstrated an interesting application of the SLICOT library for finding optimal heating strategies for multimode combination ovens (air/microwave oven). Our experience has shown that the SLICOT routines are very efficient and fast solvers for model reduction and are a useful tool in this application. Future work will involve the application of the same routines to systems with a dimension of approximately 4000 nodes (a test case with a 1500 node model has already been reduced to 61 states in approximately 1 hour of CPU time which is very good).

Acknowledgements

This work was supported by the European Committee (FAIR CT96-1192), and by the Flemish Community under contract number COF-97-08.

References

- [1] A.E. Bryson, Jr. *Dynamic Optimization*. Addison Wesley, 1999.
- [2] A.E. Bryson, Jr. and Y. Ho. *Applied Optimal Control: Optimization, Estimation, and Control*. John Wiley & Sons, 1975.
- [3] K. Åström and B. Wittenmark. *Computer Controlled Systems: Theory and Design*. Prentice-Hall, 1984.
- [4] D. Chen, R. Sing, K. Haghighi, and P. Nelson. Finite element analysis of temperature distribution in microwaved cylindrical potato tissue. *Journal of Food Engineering*, 18:351–368, 1993.

Hans Stigter, Nico Scheerlinck, Bart Nicolai, and Jan van Impe

9 SLICOT: a useful tool in industry?

Subspace Identification of Multivariable Durability Test Rigs using SLICOT

Introduction: Durability Test Rigs

When designing a new vehicle, car manufacturers make extensively use of test drives on public roads and test tracks. During these test drives, the service loads at some specific points of the car are measured. Those service loads can be accelerations, forces or displacements, and are referred to as '*target*' signals. On the one hand, the test drive measurements are indispensable to optimize and sign-off the design of the car for aspects such as durability and comfort. On the other hand, test drives on the road are expensive, time-consuming and uncomfortable for the driver who has to drive for hours and hours on often severe roads.

An interesting alternative to this expensive and time-consuming test drive procedure is a *simulation* of the measured service loads in a laboratory environment. This *service load simulation* approach starts from only one test drive on the road, during which the target signals are measured, and reproduces those targets on a multi axial test rig. The simulation is evaluated by comparing the laboratory measurements with the target signals; the goal is to drive the test rig such that the laboratory measurements approach the target signals as close as possible. Figure 9 shows an industrial test rig.



Figure 9: Industrial durability test rig

The number of shakers of an industrial test rig is minimally equal to 4 and can go up to 16 allowing to impose on each wheel axis a displacement in all three directions and a breaking torque. Each of the shakers is separately controlled by means of a PID controller.

A Multivariable Tracking Control problem

System description. Controlling multi axial test rigs like the one shown in figure 9 is a multivariable tracking problem: the goal is to calculate appropriate input signals, such that the outputs of the dynamic system follow given reference signals. From a control point of view, the *dynamic system* consists of the complete test rig including PID controllers, shakers and the car; the *inputs* are the voltages sent to the shakers; *outputs* are the measured accelerations or forces or displacements on the vehicle and the *reference signals* are the targets measured on the vehicle during a test drive on the road.

Nonlinear test rig behaviour. In the case of durability applications, the service load simulation process typically involves reproducing high-amplitude loading conditions, as these are most damaging. These high-amplitude loading conditions cause *non-linear vehicle and test rig behaviour*, hence advanced methods are needed to perform a successful service load simulation.

Classical approach. This control problem is addressed with a model-based approach leading to two phases: a *System Identification* phase and a *Target Simulation* phase. During System Identification, the test engineer tries to model the complete test rig and vehicle. Classically *linear frequency domain models*, i.e. *Frequency Response Function (FRF) models*, are used. In the next phase, Target Simulation, the identified model is used to calculate those input signals which reproduce the target signals on the test rig. Since FRF models are linear and the vehicle on the test rig can in some cases behave extremely nonlinear, the input calculation is done with an off-line, iterative procedure which aims at systematically reducing the tracking error during the consecutive iterations.

Besides the fact that FRF models are linear, the use of FRF models has also the disadvantage that the identification signals sent to the system need to be uncorrelated, i.e. linearly independent of each other. In practice this requirement is met by using a random excitation. However, on many test rigs, it is very dangerous to apply a MIMO random excitation at high level, because this can damage the vehicle seriously. So, when using FRF models, the input amplitudes need to be limited, which makes it impossible to excite the system at operational level. The consequence is that the nonlinearities in the system are not excited during the System Identification phase, which can lead to large model errors, and thus slows down the Target Simulation phase substantially.

Exciting the system at operational level, already during the System Identification phase, greatly improves the model accuracy. As explained above, this is not possible with FRF models due to the limitation of the uncorrelated input signals. Exciting the system at operation level is possible however, when *time domain* models are used. The reason is that the calculation of a time domain model does not need the input signals to be uncorrelated. This can not only lead to a more accurate system model, it certainly is a safer approach than the classically used MIMO random excitation.

LMS integrates state space model identification algorithms

State space models Within the different time domain models, LMS has put emphasis on the use of so called *state-space models*. State space models have the advantage over for

example ARMAX models, that systems with multiple inputs and outputs are described and identified in the same way as SISO systems. A state space model is described by

$$x(k + 1) = Ax(k) + Bu(k) \tag{8}$$

$$y(k) = Cx(k) + Du(k), \tag{9}$$

and can be identified using *Subspace Identification* algorithms.

To make an easy use of Subspace identification algorithms possible, LMS has developed the "State Space Modeling (SSM)" product, a stand-alone identification module, completely integrated in the standard LMS CADA-X software. Most of the mathematical calculation algorithms within SSM are based on the SLICOT software, wich is freely available through WWW: (<http://www.win.tue.nl/wgs/>) or anonymous ftp: (<ftp://wgs.esat.kuleuven.ac.be/pub/WGS-SLICOT/>). SSM allows a flexible and user-friendly interface for the calculation of multivariable state space models, based on given input-output data sets. Figure 10 shows the LMS SSM module. The upper left dialog is the main dialog of SSM guiding you through the complete identification process, starting from Data selection over the Identification and finally to the model validation step. The lower right window shows the stabilization diagram allowing you to select the main dynamics of the test rig as explained below.

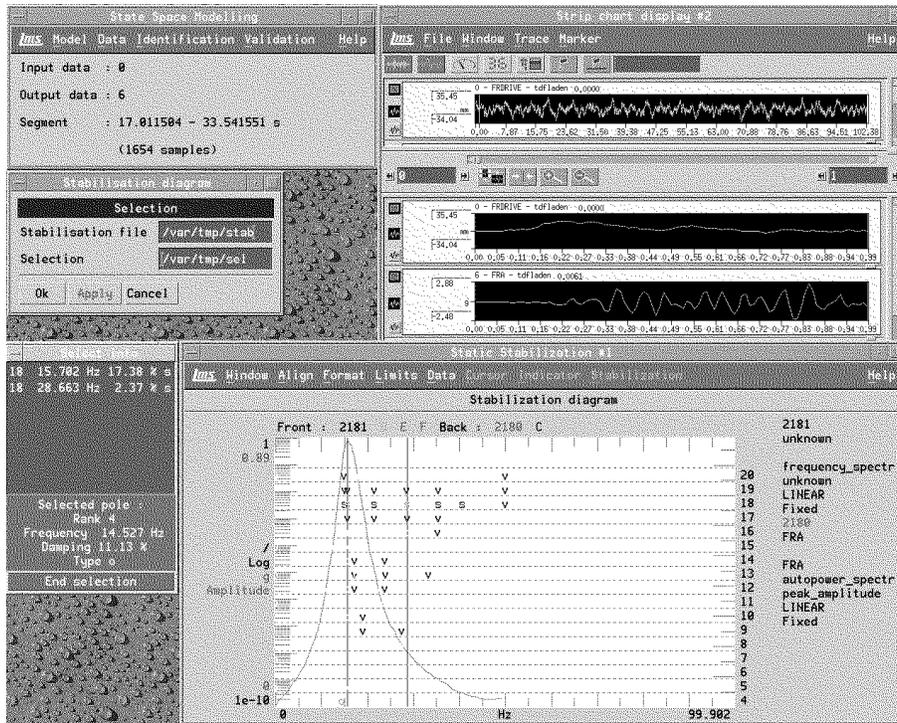


Figure 10: State Space Modeling module

Stabilization diagrams When modeling complex systems like test rigs, it is necessary to integrate stabilization diagrams in the identification process. Stabilization diagrams give a

very effective graphical representation of the evolution of the identified poles of the state space model when the model order is increased. This allows to distinguish between 'mathematical' poles - which only describe noise in the data - and 'physical' poles - which really describe the dynamics of the system - leading to increased model accuracy with lower order models.

At this moment, the use of the state space models still requires some insight in the dynamics of the system under investigation and some parameters - like the delay in the system - need to be tuned experimentally. The final goal is to develop a identification module which allows to identify accurate models in an almost completely automated manner.

Conclusion

The integration of the SLICOT software routines in the LMS software allows a easy and user-friendly subspace identification of complex multivariable systems like durability test rigs. In the future, emphasis will be put on the automation of the identification process.

Joris De Cuyper

10 Second NICONET workshop in INRIA-Rocquencourt, France

We are happy to invite all of you to our second NICONET workshop announced below.

Second NICONET WORKSHOP:
**NUMERICAL CONTROL SOFTWARE:
a useful tool in industry**

Friday, December 3, 1999
INRIA Rocquencourt, France.

First Announcement and Call for Posters

This workshop organized by the European Numerics in Control thematic Network (NICONET⁶) aims to bring together engineers, mathematicians, computer scientists and practitioners from industry and academia dealing with numerical software in systems and control and their implementation and use in industrial practice.

Recent advances on the use of numerical software libraries especially designed for solving systems and control engineering problems in a numerically reliable and efficient way will be discussed. The current status of the *freely available* SLICOT library will be extensively discussed, as well as industrial control applications and future extensions, comprising *parallel versions* and *practically oriented benchmarks*. SLICOT is a valuable tool for the reliable solution of many control problems, and for large-scale, computer-intensive control problems and real-time applications, SLICOT can lead to significant performance improvements.

Chairpersons: François Delebecque, Serge Steer (local organizers) and Sabine Van Huffel (project coordinator).

Organizing Committee: T. Backx, P. Benner, A. van den Boom, J. De Cuyper, A. Coville, D.W. Gu, S. Hammarling, V. Hernández, B. Kågström, M. Konstantinov, V. Mehrmann, A. Moner, P. Petkov, V. Sima, A. Stoorvogel, P. Van Dooren, A. Varga, M. Verhaegen, R. Wohlgenuth.

Workshop Program:

The morning session consists of a tutorial training course including demos of the newly developed SLICOT toolboxes which enable the use of SLICOT routines within the user-friendly CACSD packages MATLAB and Scilab. In the afternoon, plenary talks are scheduled describing the use of numerical control software in diverse industrial control applications. In addition, a poster session of contributed papers illustrates new developments in control software and their performance in engineering practice and industrial applications.

- 9h-10h: “*The numerics in control library SLICOT: an introductory presentation*” by Prof. Ad van den Boom (Eindhoven University of Technology) and Prof. Sabine Van Huffel (Katholieke Universiteit Leuven)
- 10h30-12h30: *Introductory presentations of the newly developed SLICOT toolboxes including demos*
 - “Basic software tools” presented by Prof. Paul Van Dooren (Université Catholique de Louvain)

⁶<http://www.win.tue.nl/wgs/niconet.html>

- “Model reduction” presented by Dr. Andras Varga (DLR, Oberpfaffenhofen)
- “Subspace identification” presented by Prof. Michel Verhaegen (Delft University of Technology)
- “Robust control” presented by Dr. Dawei Gu (University of Leicester)
- 14h-16h: *Plenary session (tentative schedule)*
 - Presentation of Scilab through a large industrial application by Eric Demay from Electricité de France.
 - Industrial control applications of SLICOT presented by the NICONET industrial partners:
 - * Ir. Joris De Cuyper from LMS International, Leuven, Belgium.
 - * Ir. Arnaud Coville from SFIM Industries, Massy, France.
 - * Dr. ir. Jobert Ludlage from IPCOS, Best, the Netherlands.
- 16h-17h30: *Demos and poster session*
- 17h30-18h: *Closing discussion*
- 18h: *Reception*

The preliminary, as well as the final program, will be announced on the NICONET website¹.

Participants of the workshop receive:

- A copy of the workshop program and proceedings book
- A copy of the last version of the SLICOT Software Library and Toolboxes (upon request)
- Documentation on SLICOT and NICONET

Prospective authors are invited to submit two copies of a camera-ready paper (1 to 5 pages long), describing the contents of the poster contribution, to the workshop secretariat (see address below) for review. Address and e-mail should be provided if possible. All accepted contributions, as well as documentation for the tutorial training course and the plenary session, will be published in the workshop proceedings.

Author’s Schedule:

Submission of camera-ready paper (1-5 pages):	<i>November 10, 1999</i>
Notification of acceptance as poster presentation	<i>November 15, 1999</i>

Registration Information:

Please register your attendance to the workshop at the address below before *November 15, 1999*.

The attendance to the workshop, the documentation and the software are free.

Second NICONET workshop Registration Form,
Friday December 3 1999

Name:

Title:

Company/Institution:

Area:

Address:

Phone:

Fax:

e-mail:

To the attention of Mrs Christine Bren at the workshop secretariat

By Fax: + 33-1-39635638

By Mail: INRIA, Relations Extérieures

Domaine de Voluceau Rocquencourt B.P. 105

78153 Le CHESNAY CEDEX (France)

By WWW: <http://www-rocq.inria.fr/Niconet/register.html>

By e-mail: symposia@inria.fr

11 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 web-site 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
```

11.1 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 root 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.
```

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

The latest library update took place on June 26, 1999. Several additional files (`.dat`, `.m`, and `.mat`), associated to the newly posted benchmark collections, have been added in the subdirectory `./slicot`. Another, similarly organized file, called `slicotPC.tar.gz`, has been put in the same directory `/pub/WGS/SLICOT/`; it contains the MS-DOS version of the Fortran source codes of the SLICOT library.

11.2 SLICOT Library update in the period January – July 1999

Known bugs have been corrected out. This resulted in updating over 20 routines. Most changes were needed for setting some output parameters before a Quick return (when some problem dimensions are zero). Details are given in the file `Release.Notes`, located in root directory `pub/WGS/SLICOT/` on the WGS ftp site.

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

- model reduction via square-root or balancing-free square-root Balance & Truncate methods applied to the α -stable part of a system;
- model reduction via square-root or balancing-free square-root Singular Perturbation Approximation methods applied to the α -stable part of a system;
- benchmark examples of (generalized) continuous-time or discrete-time Lyapunov equations;
- benchmark examples for standard or generalized time-invariant continuous-time or discrete-time dynamical systems;
- efficient evaluation of various matrix expressions (where $\text{op}(A) = A$ or A^T):
 - $\alpha R + \beta \text{op}(A)X \text{op}(A)^T$, with R and X symmetric matrices (simpler MB01RD variant);
 - a specified triangle of $\alpha R + \beta \text{op}(A)B$, or $\alpha R + \beta B \text{op}(A)$ (useful, e.g., when the result should be a symmetric matrix);
 - a specified triangle of $\alpha R + \beta \text{op}(H)B$, or $\alpha R + \beta B \text{op}(H)$, with H an upper Hessenberg matrix;
 - $\alpha \text{op}(H)A$, or $\alpha A \text{op}(H)$, with H an upper Hessenberg matrix;
 - $\alpha \text{op}(H)A$, or $\alpha A \text{op}(H)$, overwriting A , with H an upper Hessenberg matrix;
- solution (if well-conditioned) of the matrix equations $\text{op}(A)X = B$, providing error bounds on the solution and a condition estimate (a simplified version of the LAPACK Library routine `DGESVX`, useful when several sets of matrix equations with the same coefficient matrix A and/or A^T should be solved);
- finding a solution, optionally corresponding to specified free elements, to a real linear least squares problem;
- periodic Hessenberg form: reducing a product of p real general matrices to upper Hessenberg form, $H = H_1 H_2 \cdots H_p$, where H_1 is upper Hessenberg, and H_2, \dots, H_p are upper triangular, using orthogonal similarity transformations;
- computing the p real orthogonal matrices defined as the product of elementary reflectors used in the reduction to periodic Hessenberg form;
- periodic Schur form: finding the eigenvalues and Schur decomposition ($H = H_1 H_2 \cdots H_p$, with H_1 in a real Schur form and H_2, \dots, H_p upper triangular) of a product of matrices given in periodic Hessenberg form, without evaluating the product;

- computing the eigenvalues of a product of matrices, $T = T_1 T_2 \cdots T_p$, where T_1 is an upper quasi-triangular matrix and T_2, \dots, T_p are upper triangular matrices;
- applying a real elementary reflector to a real rectangular matrix from either the left or the right, exploiting the LAPACK structure of reflectors, and using in-line code for orders less than or equal to 10;
- pole assignment for a given matrix pair (A, B) , using and delivering a real Schur form of the closed-loop system matrix;
- choosing a real eigenvalue or a pair of complex conjugate eigenvalues at minimal distance to a given real or complex value;
- eigenstructure assignment for a controllable matrix pair (A, B) in orthogonal canonical form;
- condition number and forward error bound estimates for the solution of a continuous-time algebraic Riccati equation;
- condition number and forward error bound estimates for the solution of a discrete-time algebraic Riccati equation;
- accurate solution of a continuous-time or discrete-time algebraic Riccati equation (Schur vector method), with condition number and forward error bound estimates (a functionally extended and more accurate version of the SLICOT Library routine SB02MD);
- efficient and accurate construction of the $2n \times 2n$ Hamiltonian or symplectic matrix associated to a linear-quadratic optimization problem, used to solve a continuous- or discrete-time algebraic Riccati equation, respectively (a functionally extended and, for the discrete-time case, more accurate version of the SLICOT Library routine SB02MU);
- condition number and forward error bound estimates for the solution of a continuous-time Lyapunov equation;
- forward error bound estimate for the solution of a continuous-time Lyapunov equation;
- estimating the separation between $\text{op}(A)$ and $-\text{op}(A)^T$ and the 1-norm of the Θ operator for a continuous-time Lyapunov equation;
- condition number and forward error bound estimates for the solution of a discrete-time Lyapunov equation;
- forward error bound estimate for the solution of a discrete-time Lyapunov equation;
- estimating the “separation” between $\text{op}(A)$ and $\text{op}(A)^T$ and the 1-norm of the Θ operator for a discrete-time Lyapunov equation;
- solution of a real continuous-time Lyapunov matrix equation, with condition number and forward error bound estimates;
- solution of a real discrete-time Lyapunov matrix equation, with condition number and forward error bound estimates;

In addition, *benchmark data files*, possibly called by the benchmark routines, at user's request, as well as the associated MATLAB m- and mat-files, have been added.

Moreover, several components of the new, SLICOT-based MATLAB toolboxes have been posted on the ftp site. At the time being, there are two mexfiles for model reduction, directly callable on Windows 9x or NT platforms (using Fortran 90 memory allocation scheme), and two mexfiles (source form only) for solving standard or generalized Sylvester and Lyapunov equations by various methods, including methods for stable non-negative definite (generalized) Lyapunov equations, which can be used on various platforms (Fortran 77). These mexfiles are invoked by a series of m-files, also provided. Included are some MATLAB test programs and mat files with test data. The work is in progress, and new mexfiles will be added soon. Performance results and comparisons with MATLAB functions will be reported and/or made available.

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. Previous updates are described, in reverse chronological order, in the file `Release.History`, at the same address.

11.3 New NICONET Reports

Recent NICONET reports (available after January 1999), 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:

- Peter Benner, Enrique S. Quintana-Orti, Gregorio Quintana-Orti. *A portable subroutine library for solving linear control problems on distributed memory computers* (file `nic1999-1.ps.Z`).

This paper describes the design of a software library for solving the basic computational problems that arise in analysis and synthesis of linear control systems. The library is intended for use in high performance computing environments based on parallel distributed memory architectures. The portability of the library is ensured by using the BLACS, PBLAS, and ScaLAPACK as the basic layer of communication and computational routines. Preliminary numerical results demonstrate the performance of the developed codes on parallel computers. The suggested library can serve as a basic layer for PSLICOT, a parallel extension of the Subroutine Library in Control Theory (SLICOT).

- Peter Benner, Heike Faßbender. *SLICOT drives tractors* (file `nic1999-2.ps.Z`).

This paper describes the successful application of a SLICOT subroutine in a control engineering problem. Based on GPS data it is possible to automatically steer farm vehicles along a prescribed trajectory. The bottleneck for the successful on-line implementation of a LQG regulator is the numerical solution of a discrete-time algebraic Riccati equation in real-time and at high accuracy. This is achieved employing a Fortran 77 subroutine from the Subroutine Library in Control Theory—SLICOT.

- Bert Haverkamp. *Efficient implementation of subspace method identification algorithms* (file nic1999-3.ps.Z).

This paper summarises the results of a study to improve existing Subspace Method Identification (SMI) algorithms. Significant improvements in calculation speed can be achieved by combining components from existing algorithms namely N4SID and MOESP. A second improvement can be achieved by more efficient implementation of critical parts of the algorithms.

- Da-Wei Gu, Petko Petkov, Mihail M. Konstantinov. *An Introduction to H-infinity Optimisations Designs* (file nic1999-4.ps.Z).

This Niconet report is prepared for users of the software package SLICOT who are not familiar with the H-infinity optimisation design approach. Together with some previous Niconet reports it is hoped that the reader would have a general idea about the H-infinity method, know how to use the algorithms available in SLICOT to synthesize a controller for a standard H-infinity optimisation problem and, furthermore, be aware of some difficulties such as singularity in the H-infinity controllers design.

- Petko Petkov, Da-Wei Gu, Mihail Konstantinov. *Fortran 77 routines for H_∞ and H_2 design of linear discrete-time control systems* (file nic1999-5.ps.Z).

This paper presents Fortran 77 subroutines intended for state-space design of H_∞ (sub)optimal controllers and H_2 optimal controllers for linear discrete-time control systems. The subroutines make use of LAPACK and BLAS libraries and produce estimates of the condition numbers of the matrices which are to be inverted and estimates of the condition numbers of the matrix algebraic Riccati equations which are to be solved in the computation of the controllers. The subroutines will be included in the SLICOT library.

- Daniel Kressner, Volker Mehrmann, Thilo Penzl. *CTLEX — a collection of benchmark examples for continuous-time Lyapunov equations* (file SLWN1999-6.ps.Z).

This paper describes the benchmark collection CTLEX, that contains test examples of continuous-time algebraic Lyapunov equations. The main focus of this collection is on scalable benchmark examples depending on parameters, which affect the conditioning of the equation. Such examples are particularly useful for the assessment of the complexity and the accuracy of numerical solution methods.

- Daniel Kressner, Volker Mehrmann, Thilo Penzl. *DTLEX — a collection of benchmark examples for discrete-time Lyapunov equations* (file SLWN1999-7.ps.Z).

This paper describes the benchmark collection DTLEX, that contains test examples of discrete-time algebraic Lyapunov equations. These matrix equations are also known as Stein equations. The main focus of DTLEX is on scalable benchmark examples depending on parameters, which affect the conditioning of the equation. Such examples are particularly useful for the assessment of the complexity and the accuracy of numerical solution methods.

- Andras Varga. *Model reduction routines for SLICOT* (file nic1999-8.ps.Z).

This paper reports on the newest developments of model reduction software for SLICOT. Three enhanced accuracy model reduction algorithms belonging to the class of methods

based on or related to balancing techniques form the basis of model reduction software in SLICOT. These methods are primarily intended for the reduction of linear, stable, continuous- or discrete-time systems. However, in combination with additive spectral decomposition or coprime factorization techniques, the basic methods can be employed to reduce unstable systems too. The implemented computational methods for reduction of stable and unstable systems, and the associated software available in SLICOT are presented. Performance comparisons performed using appropriate interface software to user-friendly environments like MATLAB and Scilab show the superiority of SLICOT model reduction tools over existing model reduction software.

- Daniel Kressner, Volker Mehrmann, Thilo Penzl. *DTDSX — a collection of benchmark examples for state-space realizations of discrete-time dynamical systems* (file SLWN1998-10.ps.Z, revised in June 1999).

This paper describes a benchmark collection for state-space realizations of time-invariant discrete-time dynamical systems. The collection is intended to provide a means for testing the correctness, accuracy, and speed of numerical methods for several problems arising in control theory. It has been implemented in Fortran and MATLAB.

- Paul Van Dooren. *Selection of basic software tools for structured matrix decompositions and perturbations* (file SLWN1999-9.ps.Z).

In this note a survey is given of areas of systems and control where structured matrix problems are important. In identification we mention four different types of data collection: impulse response, input-output pairs, frequency response and covariance data. In each of those, the identification problem can be rewritten in terms of structured matrix problems for which there exist fast decompositions. The use of structured matrix decompositions should yield an improvement in speed of computations. In analysis and design one encounters eigenvalue problems with specific structure such as cyclic, Hamiltonian and symplectic matrices. For those problems it is important to use structure preserving decompositions, mainly to improve the numerical accuracy of the computations, although these algorithms typically yield improved computational complexities as well. We also list the key numerical routines that should be provided in the SLICOT library in order to tackle most of the problems mentioned in this note.

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

11.4 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 disseminate information and promote SLICOT, are the following:

- Fourteenth Householder Symposium on Numerical Linear Algebra, Chateau Whistler, Whistler B.C., Canada, June 14–18, 1999.
- The Fourth International Congress on Industrial and Applied Mathematics, Edinburgh, Scotland, ICIAM99, July 5–9, 1999.
- 19th IFIP TC7 Conference on System Modelling and Optimization, Cambridge, England, July 12–16, 1999.

- 1999 IEEE International Symposium on Computer-Aided Control System Design, CACSD '99, Hawaii, August 22–26, 1999.
- European Control Conference, Karlsruhe, Germany, 31 August – 3 September, 1999.
- Second NICONET Workshop⁷, Paris, December 3, 1998 (organized by NICONET, with external participation). This workshop is open to all end-users and industrial software companies. The workshop consists of a tutorial training course (including demos of the newly developed SLICOT toolboxes), several plenary talks, demo and poster session, and a closing discussion.

Vasile Sima

⁷For more detailed information and a preliminary program, see the NICONET website <http://www.win.tue.nl/wgs/niconet.html>