4

ISSN 1373-6787

January 2000

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/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

1 Editorial

Welcome to the fourth 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. In the last 6 months several important events happened which are worthwhile to be mentioned. First of all, we held our Mid-Term Assessment meeting on December 2, 1999: the NICONET activities were evaluated very positively which implies that we can continue our activities for another 2 years. On December 3, we held our second NICONET workshop at INRIA Rocquencourt near Paris-Versailles in France. The highlights of this workshop are described in Section 9. Besides this positive news we also have very sad news to share: on December 17, Thilo Penzl, who was an enthusiastic active member of NICONET and was responsible for the benchmark collection, died in a tragic avalanche accident in the Canadian Cascade Mountains. A short description of Thilo's scientific carreer is included in Section 2.

Sections 3 to 7 present as usual the new updates of the SLICOT library in subfields of systems and control. Also, new benchmark collections have been added recently to the SLICOT library (see Section 2). As announced above, Section 9 discusses the program of the second NICONET workshop held on December 3, 1999, at INRIA Rocquencourt in France. In Section 9, Jobert Ludlage and Ton Backx from IPCOS Technology, one of the industrial partners of NICONET, discuss the importance of SLICOT in model based process control, a basic technology for reproducible and predictable process operation. Finally, Section 11 gives more details about the newest additions to the SLICOT library, new reports and forthcoming events.

I hope you enjoy reading this newsletter.

Sabine Van Huffel
NICONET coordinator

2 New developments in the SLICOT benchmark library

2.1 New additions

The SLICOT benchmark library, an important tool for the development, analysis and testing of numerical methods and codes for the solution of control problems is constantly improved and updated. Currently there are 6 major collections, 3 for continous-time and 3 for discrete-time problems. Release 2.0 of the benchmark collections for Riccati equations has recently been issued, i.e.

1. *CAREX — A Collection of Benchmark Examples for Continuous-Time Algebraic Riccati Equations (Version 2.0).* A collection of benchmark examples is presented for the numerical solution of continuous-time algebraic Riccati equations. The collected examples focus mainly on applications in linear-quadratic optimal control theory. This version updates an earlier benchmark collection and includes one new example.

2. *DAREX — A Collection of Benchmark Examples for Discrete-Time Algebraic Riccati Equations (Version 2.0).* This is the second part of a collection of benchmark examples for the numerical solution of algebraic Riccati equations. This version updates an earlier benchmark collection. Some of the examples have been extended by incorporating parameters and there have been some new additions to the collection.

These collections may serve for testing purposes in the construction of new numerical methods, but may also be used as a reference set for the comparison of methods. For details see the recent SLICOT working notes SLWN1999-14, SLWN1999-16 by Jörn Abels and Peter Benner.

2.2 Thilo Penzl, in memoriam

Since Thilo Penzl was highly involved in the development of the SLICOT benchmark library, we want to highlight his life in this section.

On December 17, Thilo Penzl died in a tragic avalanche accident in the Canadian Cascade Mountains while pursuing his favourite hobby of mountain climbing. Thilo was currently a post-doc at the Department of Mathematics and Statistics at the University of Calgary. He was about to return to the Technische Universitaet Chemnitz to take a position as assistant professor ("Hochschulassistent") at the Department of Mathematics in the research group "Numerical Linear Algebra" and the Sonderforschungsbereich 393 "Numerical Simulation on Massively Parallel Computers" on January 1, 2000.

Thilo was born in June 1968 in Plauen, Germany. He received his PhD in mathematics in 1998 at the Technische Universitaet Chemnitz with his thesis "Numerical Solution of Large Lyapunov Equations". He was an active member of the Working Group on Software participating in the development of the Subroutine Library in Control Theory (SLICOT). His latest research was devoted to model reduction and optimal control for large sparse control systems arising from discretized PDEs. He will always be remembered as a good friend and colleague who was liked and respected by everybody. We miss him a lot.

Volker Mehrmann and Peter Benner

3 Basic numerical SLICOT tools for control

Basic mathematical software tools are the basic building blocks for most other activities of this network. This is why this activity was scheduled as Task I. Its first part, Task I.A, was finalized in December 1999, and is described below. After that, we briefly describe the planning of the second part, Task I.B, started halfway 1999 and to be finalized in December 2000.

3.1 Task I.A : Standard and generalized state space systems and transfer matrix factorizations

I.A.1 : List of Routines

Task I.A.1 consisted of the selection and standardization of basic numerical routines for systems and control. There are now 45 user-callable routines ready that have been standardized in the first 2 years and can be grouped in the following chapters :

- **Mathematical Routines**: Routines for Hamiltonian, symplectic and various other eigenvalue and singular value problems
- **Transformation Routines** : Routines for various state space transformations
- **Analysis Routines** : Routines for transfer function norm calculations
- **Synthesis Routines** : Routines for Lyapunov and Riccati equations
- **Factorization Routines** : Routines for coprime factorizations and state space representations.

In addition to these user-callable routines, a large number of auxiliary routines have been written, standardized and documented, such as Lyapunov, Sylvester and Riccati solvers. 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. Together with the user-callable routines the SLICOT library contains more that 100 standardized and documented routines. The user-callable routines are listed in the working note SLWN1999-17.

I.A.2 : Interfacing Toolboxes

Task I.A.2 makes the above-mentioned basic software tools more "accessible" by implementing them in a user-friendly environment, so that little technical background is required to use the tools to almost full functionality. We have integrated top level routines of SLICOT in MATLAB via mex-files. Since such mex-files are rather big, we minimized 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). The integration in `Scilab` was done similarly as for MATLAB.

The routines have been grouped in the following groups :

- **Linear Matrix Equations**: Routines for Lyapunov and Stein equations
- **Generalized Linear Matrix Equations** : Routines for generalized Lyapunov and Stein equations

- **Riccati Equations** : Routines for discrete and continuous time algebraic Riccati equations
- **Realizations** : Routines for controllability, observability and minimal realizations
- **Transformation Routines** : Routines for balancing, Schur form and block diagonal forms.
- **Coprime Factorization Routines** : Routines for constructing state space representations of left and right coprime factorizations.

Most of these routines are described in the Working Note SLWN1999-11 and the files are available via ftp. The list of all the mex files and their corresponding m-files is reported in the Working note SLWN1999-17.

I.A.3 : Benchmarks

Tasks I.A.3 is the selection of benchmarks for task I.A. Six collections of benchmarks have been put together for this task and guidelines for such benchmark collections have been issued:

- Benchmark collections in SLICOT (Working Note SLWN1998-5)
- CTDSX, a collection of benchmarks for state-space realizations of continuous-time dynamical systems (Working Note SLWN1998-9)
- DTDSX, a collection of benchmarks for state-space realizations of discrete-time dynamical systems (Working Note SLWN1998-10)
- CTLEX, a collection of benchmark examples for continuous-time Lyapunov equations (Working Note SLWN1999-6)
- DTLEX, a collection of benchmark examples for discrete-time Lyapunov equations (Working Note SLWN1999-7)
- CAREX, a collection of benchmark examples for continuous-time algebraic Riccati equations (Working Note SLWN1999-14)
- DAREX, a collection of benchmark examples for discrete-time algebraic Riccati equations (Working Note SLWN1999-16)

These collections contain as well examples from real systems as artificial examples that test numerical reliability of the subroutines of our library. The details of the different benchmark collections are described in the respective notes. These collections will also be of valuable help for the other tasks of the project.

I.A.4 : Examples

This task is the selection of industrial design problems. The routines of Task I.A are basic numerical routines that are not directly called in industrial applications but that are needed indirectly through the more advanced Tasks of this Network. For this reason we consider two sets of examples.

The first set are test examples from the benchmark collections. They test the reliability of the numerical methods implemented in the software library. These examples indeed contain a set of examples of which the sensitivity of the computed quantities varies, and they can therefore check if our routines react appropriately to such "difficult" cases. The second set of examples are borrowed from Task II.A since this task uses basic routines from Task I.A to build reduced order models. We refer to NICONET Report 1999-8 *Model Reduction Routines for SLICOT*, for more details about these examples.

Some used industrial benchmark examples for model reduction are:

- PS: Continuous-time power system model ($n = 7$)
- PSD: Discrete-time power system model ($n = 7$)
- TGEN : Nuclear plant Turbo-generator model ($n = 10$)
- ACT: Actuator model ($n = 5$)
- ATTAS: Linearized aircraft model ($n = 55$)
- CDP: CD-player finite element model ($n = 120$)
- GAS: Gasifier models linearized at 0%, 50%, 100% loads ($n = 25$)

Extensive numerical tests are reported in Working Notes 1999-8 and 1999-11. Here we report only a few selected examples which show the improved speed of the SLICOT routines with respect to the corresponding MATLAB routines from the Control Toolbox. These results are based on test examples from the CTDSX Benchmark collection. They compare the SLICOT-based mex-files for controllability, observability and minimality of a state space system (slconf, slobsf, slminr), with the corresponding m-files of the MATLAB Control Toolbox (ctrbf, obsvf, minreal). The table shows the comparison of execution times.

n	m	p	Time		Time		Time	
			slconf	ctrbf	slobsf	obsvf	slminr	minreal
39	20	19	<0.01	0.05	0.05	0.05	0.05	0.11
100	1	100	<0.01	2.91	0.06	0.11	0.05	6.75
421	211	211	5.66	22.13	6.05	17.354	15.16	6694.39

For these examples the numerical accuracy of the compared routines does not differ substantially (one digit of accuracy unless the routines yield full accuracy). But the speed of the routines is clearly in favor of the SLICOT routines. For the last minimum realization problem with dimensions $n = 421$, $m = p = 211$, SLICOT needed 15.16 sec, while MATLAB took 6694.39 sec, just for saying that the system is already minimal !

The comparisons made in the Working Notes 1999-8 and 1999-11 show that SLICOT is in general faster, more accurate and more reliable than the comparable MATLAB routines. As a consequence, Mathworks showed interest in including SLICOT software in an improved Control Toolbox (negotiations are still in progress).

I.A.5 : Toolbox

The deliverable for this task is a Basic Software Toolbox, containing all implemented new routines, the accompanying documentation, mex-files developed in this task and a demonstration script. This demo file uses the benchmark examples described in the Working Notes of this Task (1998-9 and 10, 1999-6,7,14 and 16) and the interfacing mex-files described in the Working Note 1999-11. This Demo was used at the European Control Conference of August 1999 in Karlsruhe, where many attendees learned to appreciate the reliability and speed of the numerical tools of NICONET. The Demo is also available from the `wgs.esat.kuleuven.ac.be` ftp site at `pub/WGS/SLICOT/MatlabTools/Windows/SLToolboxes/basic.mex.zip`.

3.2 Task I.B : Structured matrix decompositions and perturbations

Structured matrix problems arise in many problems of systems and control. The typical matrix structures encountered are Toeplitz, Hankel, Hamiltonian, symplectic and patterned matrices. In the context of systems and control these are found in three areas. The largest is definitely identification, where data are collected and arranged in structured matrices, whose decompositions yield the parameters of the system to be identified. The second is analysis and design where structured eigenvalue and singular value problems occur. The third is that of robustness, where structured optimization problems are found.

This task just started in July 1999. The first subtask I.B.1 consists in selecting the routines to be standardized. A first report was delivered that proposed a selection of basic routines for each of the relevant topics in these areas (Working Note SLWN1999-9).

Andras Varga and Paul Van Dooren

4 SLICOT tools for model reduction

We are pleased to announce that a recently developed model reduction toolbox has now been included in the SLICOT package. The development of this toolbox was the subject of Task II.A[1] of the NICONET project. The main functionalities of the toolbox include the reduction of stable and unstable continuous- and discrete-time linear systems. In the development of the toolbox, computational reliability and efficiency, as well as enhanced numerical accuracy have been our main concerns. The coding of all subroutines uses extensively the linear algebra standard package LAPACK and has been done according to the SLICOT implementation and documentation standards. All developed subroutines have been extensively tested on various test examples and fully documented. For the easy use by MATLAB and `Scilab` users, *mex*- and *m*-functions have been developed so that the major functionality of SLICOT routines is available within these user-friendly environments. A unique, functionally reach and flexible *mex*-function `sysred` covers practically the complete functionality of all implemented model reduction routines. To provide a convenient interface to work with *control objects* defined in the MATLAB Control Toolbox or in `Scilab`, 9 easy-to-use *m*-functions have been additionally implemented explicitly addressing some of supported features.

The completed model reduction toolbox is described in detail in the NICONET Report 1999-8.[2] The test results and performance comparisons reported here show the superiority of SLICOT model reduction tools over existing model reduction software. In what follows we illustrate the application of the new model reduction tools in three industrial case studies: a linearized aircraft model, a CD-player finite element model and industrial gasifier model.

4.1 ATTAS: Linearized aircraft model

This model describes the linearized rigid body dynamics of the DLR Advanced Technology Testing Aircraft System (ATTAS) during the landing approach. The nonlinear model of ATTAS used for linearization has been obtained using the object oriented modelling tool Dymola [1]. Besides flight dynamics, this model includes actuators and sensors dynamics, as well as engine dynamics. Several low pass filters to eliminate structure induced dynamics in outputs are also included. The total order of the model is 51. The linearized model has an unstable spiral mode. Moreover, because of presence of position states, there are three pure integrators in the model and an additional one for the heading angle. There are 6 control inputs and 3 wind disturbance inputs, and 9 measurement outputs. This model serves basically for the evaluation of linear handling criteria in a multi-model based robust autopilot design.

To speed-up the evaluation of different handling quality criteria, lower order design models have been obtained by using model reduction techniques. A 15-th order approximation has been computed using the *Balance & Truncate* (B&A) model reduction method followed by minimal realization which fits almost exactly the original 51 order model both in time as well as in frequency domain. Figure 1 shows a very good agreement obtained between the frequency responses of the original and reduced model for element g_{22} of the corresponding transfer function matrix.

For longitudinal flight, a minimal order stable model has been derived by combining model reduction and minimal realization techniques. The reduced longitudinal ATTAS model has 7

[1] see also at `http://www.win.tue.nl/niconet/NIC2/NICtask2A.html`

[2] available at `ftp://wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/NIC1999-8.ps.Z`

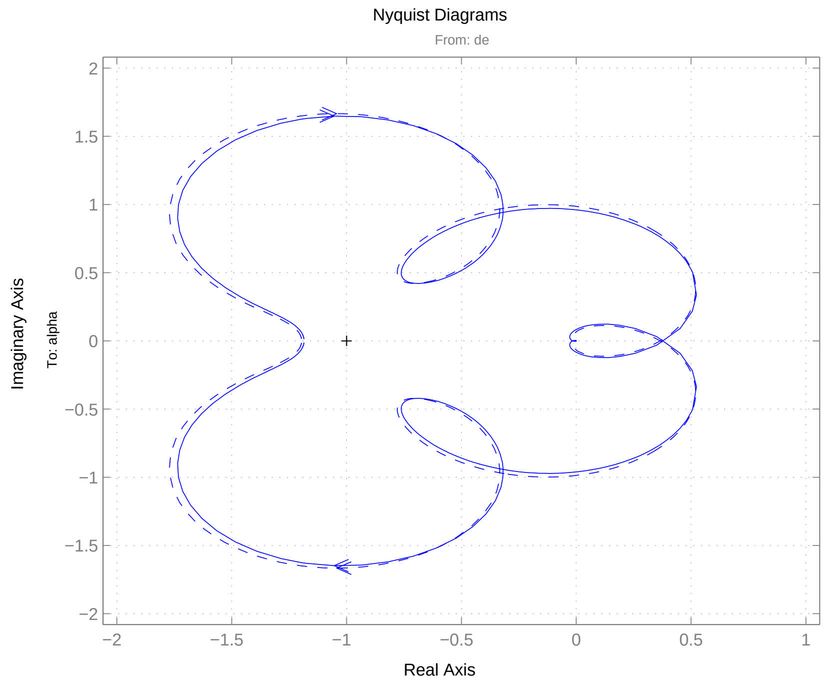

Figure 1: Comparison of frequency responses for element $g_{22}(s)$ of ATTAS.

states, 4 inputs and 4 outputs. For lateral flight, a minimal order model has been computed having 10 states, 2 inputs and 5 outputs. Both these models approximate practically exactly the corresponding parts of the dynamics of the original 51 order model. Note that handling this model raises several difficulties for currently available model reduction software such as the presence of unstable modes or of redundant dynamics (non-minimal model). For instance, this model is intractable with standard model reduction tools available in the Control Toolbox of MATLAB.

4.2 CDP: CD-player finite element model

This is a 120-th order single-input single-output system which describes the dynamics between the lens actuator and radial arm position of a portable compact disc player discussed in [2]. Due to physical constraints on the size of the systems's controller, a reduced model with order $r \leq 15$ is desired. For testing purposes, three 10-th order models have been determined using the B&T, *singular perturbation approximation* (SPA) and *Hankel-norm approximation* (HNA) methods. Figure 2 compares the performance of different computed approximations on basis of Bode plots. All methods approximate satisfactorily the central peek at frequency about 120 Hz, but have different approximation properties at low and high frequencies. Both SPA and HNA approximations seems to be inappropriate, although the stationary error for the SPA method is zero. However, the B&T method appears to provide a good 10-th order approximation.

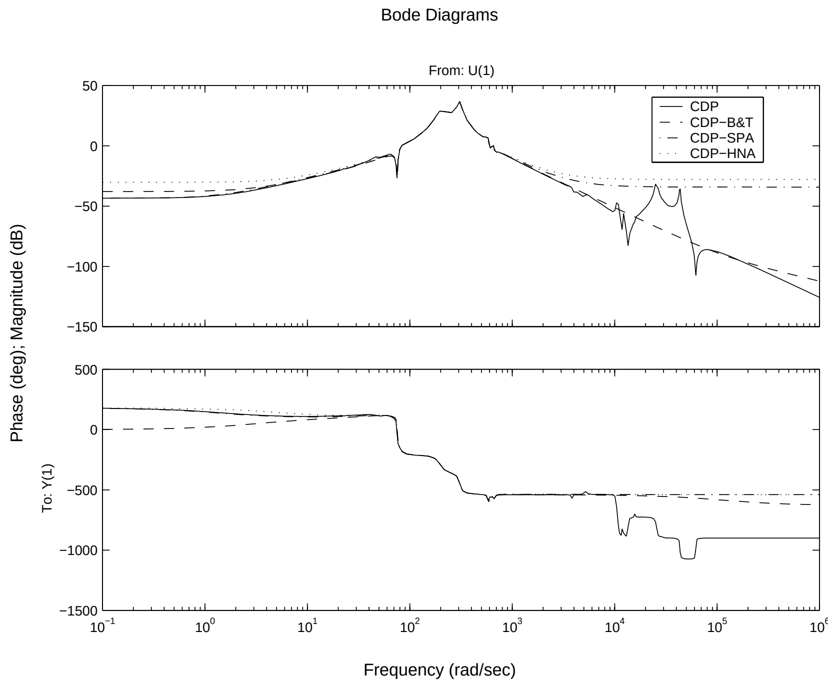

Figure 2: Comparison of frequency responses for CDP.

4.3 GAS: Gasifier model

A detailed nonlinear gasifier model has been developed by GEC ALSTHOM, in October 1997, as a benchmark problem for simulation and robust control. The model includes all significant effects; e.g., drying of coal and limestone, pyrolysis and volatilisation of coal, the gasification process itself and elutriation of fines. This model has been validated using measured time histories from the British Coal CTDD experimental test facility and it was shown that the model predicts the main trends in fuel gas quality. Linearized models at 0%, 50% and 100% have been generated to be used for a multi-model based robust controller design. Some analysis results on the 100% load models are discussed in [3]. Here, numerical difficulties with respect to using MATLAB model reduction tools, but also of the symbolic manipulation tools in `Mathematica`, have been reported.

The cause of reported numerical difficulties lies in the poor scaling of the model. The GAS model has order 25 and is non-minimal. The norm of state matrices for the three models ranges between $7.64 \cdot 10^8$–$1.03 \cdot 10^9$, but after scaling with the SLICOT routine TB01ID, all norms can be reduced below 100. Such a preliminary scaling is not necessary for using the model reduction software, being an optional feature of all user callable routines and implicit feature for the *mex*- and *m*-functions. Still, for simulations we used the scaled models to avoid numerical difficulties with MATLAB plotting functions and to make the comparison more reliable. All three models are non-minimal as can be seen by examining the smallest Hankel singular values. For the 100% load model the last 10 Hankel singular values are

$$\sigma_{16-25} = \{0.64046, 1.0852 \cdot 10^{-4}, 0, 0, 0, 0, 0, 0, 0, 0, \}$$

Note that the Hankel-norm (the largest singular value) for this model is $3.4078 \cdot 10^5$. The

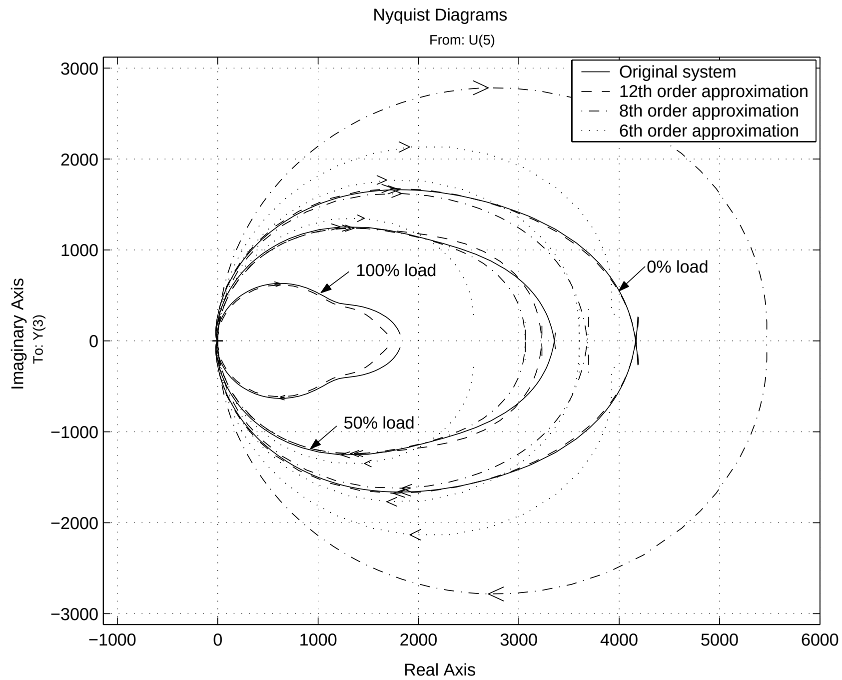

Figure 3: Comparison of frequency responses for $g_{35}(s)$ elements of GAS models.

same qualitative results are true for the other two models.

We computed three 16 order reduced models, which practically, can be not distinguished from the original models on basis of time or frequency responses. Several lower order approximations have been also computed of orders 6, 8 and 12. The 12 order models represent very good approximations of the original models and can serve as basis for designing a unique robust controller ensuring satisfactory performance for all three models. A comparison on basis of elements $g_{35}(s)$ of the corresponding transfer-function matrices is shown in Figure 3.

References

[1] H. Elmquist. Object-oriented modeling and automatic formula manipulation in Dymola. Scandinavian Simulation Society SIMS'93, Konsberg, Norway, June 1993.

[2] P. Wortelboer. *Frequency-weighted Balanced Reduction of Closed-loop Mechanical Servo-systems: Theory and Tools.* PhD thesis, Techn. Univ. Delft, 1994.

[3] N. Munro. Control system analysis and design using Mathematica. In *Proc. CDC'98, Tampa, FL*, pp. 3681–3685, 1998.

Andras Varga

5 SLICOT tools for subspace identification

The standardization of the selected subspace identification tools according to the SLICOT standards in Fortran was planned to be completed by December 1999. However because of the unforeseen investment into our study of new modifications of the implementations of existing subspace identification routines, as reported in SLICOT Working Note 1999-3, and the priority to standardize first the basic routines in Topic I and Topic II, the standardization was delayed. The study of new implementations lead to dramatic improvements in numerical efficiency and robustness. These properties make the schemes far more attractive to industry than their ancestors. The current status of the toolbox is the following:

1. The standardization of the building blocks with the features of concatenating data sets and the ability to use the basic building blocks in a stand-alone fashion according to the guidelines described in SLICOT Working Note 1999-3 will be completed by March 1st, 2000 (subtask III.A.1).

2. The documentation and standardization of the benchmark examples used in SLICOT Working Note 1998-6 and 1999-3 will be finalized by June, 2000 (subtask III.A.3).

Further improvement of the computational efficiency is still an important issue for getting the software widely accepted in industry. Therefore a special effort is devoted to further exploit the structure of the data matrices handled by the subspace identification routines in order to improve the initial data compression with an order of magnitude. This work is currently being performed at KUL-SISTA by Drs. Nicola Mastronardi in cooperation with Prof. P. Van Dooren (UCL-CESAME) and Prof. S. Van Huffel (KUL-SISTA). As a result it is still possible that the industrial testing and integration in the CAD friendly environments can start in February 2000 and be completed at the end of the first quarter of 2000 (subtasks III.A.2 and 4). At that time task III.A will be completed.

Michel Verhaegen

6 SLICOT tools for robust control

In the second half of 1999, the SLICOT robust control toolbox was expanded to include a new member, a set of the $\mathcal{H}_\infty$ Loop Shaping Design Procedure (LSDP) routines for the continuous-time case. The user-callable routines are listed in the following table.

Loop shaping design

SB10ID.F	Loop shaping design of output controllers (main subroutine)
SB10JD.F	Transformation of a descriptor system into regular form

Furthermore, the gateway functions and corresponding mex-files for the design of continuous-time and discrete-time $\mathcal{H}_\infty$ and $\mathcal{H}_2$ (sub) controllers, and for the calculation of the $\mathcal{H}_\infty$ norm of continuous-time systems in PC-MATLAB have been modified and tested on the SLICOT benchmark examples. They have been renamed and are listed in the following table.

Test routines for $\mathcal{H}_\infty$ and $\mathcal{H}_2$ design

TSB10FD.F	Design of $\mathcal{H}_\infty$ suboptimal continuous-time output controllers
TSB10HD.F	Design of $\mathcal{H}_2$ optimal continuous-time output controlers
TSB10DD.F	Design of $\mathcal{H}_\infty$ suboptimal discrete-time output controllers
TSB10ED.F	Design of $\mathcal{H}_2$ optimal discrete-time output controlers
TAB13HD.F	Computation of the $\mathcal{H}_\infty$ norm of a continuous-time system

In addition, four routines, intended for solving matrix algebraic Riccati and Lyapunov equations with condition and accuracy estimates, have been delivered to SLICOT.

SB02RD.F	Solution of the continuous-time matrix Riccati equation with condition and forward error estimation
SB02PD.F	Solution of the continuous-time matrix Riccati equation by the matrix sign function method
SB03RD.F	Solution of the continuous-time matrix Lyapunov equation with condition and forward error estimation
SB03PD	Solution of the discrete-time matrix Lyapunov equation with condition and forward error estimation

All subroutines have been standardized according to the SLICOT convention.

During the period, the following three NICONET/SLWN reports were produced, including a very interesting study on the comparison of the MATLAB and SLICOT continuous-time algebraic Riccati equation solvers, which showed the superiority of routines in the latter with respect to accuracy and efficiency.

Niconet Report, **No. 1999-10** P.Hr. Petkov, M.M. Konstantinov, D.-W. Gu and V. Mehrmann. "Numerical Solution of Matrix Riccati Equations: A Comparison of Six Solvers".

Niconet Report, **SLWN1999-12** D.-W. Gu, P.Hr. Petkov and M.M. Konstantinov. "$\mathcal{H}_\infty$ and $\mathcal{H}_2$ Optimization Toolbox in SLICOT ".

Niconet Report, **No. 1999-13** A.A. Stoorvogel. "Numerical Problems in Robust and $\mathcal{H}_\infty$ Optimal Control".

Da-Wei Gu

7 SLICOT tools for nonlinear systems in robotics

One of the objectives of the NICONET project is to provide the SLICOT numerical software library with tools for nonlinear control systems. In this sense, the objective of this task is to implement a standard interface to the most used integrator packages (ODEPACK, DASSL, DASPK, RADAU5, DGELDA).

SLICOT will deal with the simulation of non-linear control systems which can be described in terms of *ordinary differential equations* (ODEs):

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

or DAEs,

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

where $x(t)$ is the state vector, $u(t)$ is the input vector, $y(t)$ is the output vector, p is the parameter vector.

As it was presented in previous newsletters, an interface has been designed to compile all the integrator packages on a single entry point. This standard interface has been implemented in both Fortran and MATLAB systems.

The implementation in MATLAB has been quite complex. The integrator packages are fully written in FORTRAN, and changing the code would lead to difficulties in subsequent updates of the integrator versions. A two-way gateway was implemented to capture the communication flow from the integrator package to the routines that the user supplies to define the structure of the system. These routines can be written in MATLAB code now. In this sense, MATLAB is also complemented with the capabilities of these packages.

To validate the correctness of the system, a test-battery has been proposed. This test-battery comprises many cases coming from the packages and from the IVPTestSet[3]. The use of a standardised interface has allowed to compare different integrators by just changing the code number of the integrator package.

As an example, results obtained with LSODE ODEPACK test problem are shown in figure 4. The expression of the problem is :

$$\begin{aligned} \dot{y}_1 &= -0.4y_1 + 10000y_2y_3 \\ \dot{y}_2 &= 0.4y_1 - 10000y_2y_3 - 0.0000003y_2^2 \\ \dot{y}_3 &= 0.0000003y_2^2 \end{aligned}$$

$$y_1(0) = 1, y_2(0) = y_3(0) = 0, t_0 = 0, t_f = 40$$

As figure 4 shows, there are no differences among the solvers used (RADAU5, DA2SSL, LSODE). Also, results with a package developed at the UPV are shown.

In the next period, the interface will be migrated to `Scilab` and industrial cases will be tested. After the completion of the standard package for nonlinear control systems, standard interfaces for solving nonlinear systems of equations and for optimisation codes will be implemented using a similar procedure.

[3]see http://www.cwi.nl/cwi/projects/IVPtestset/

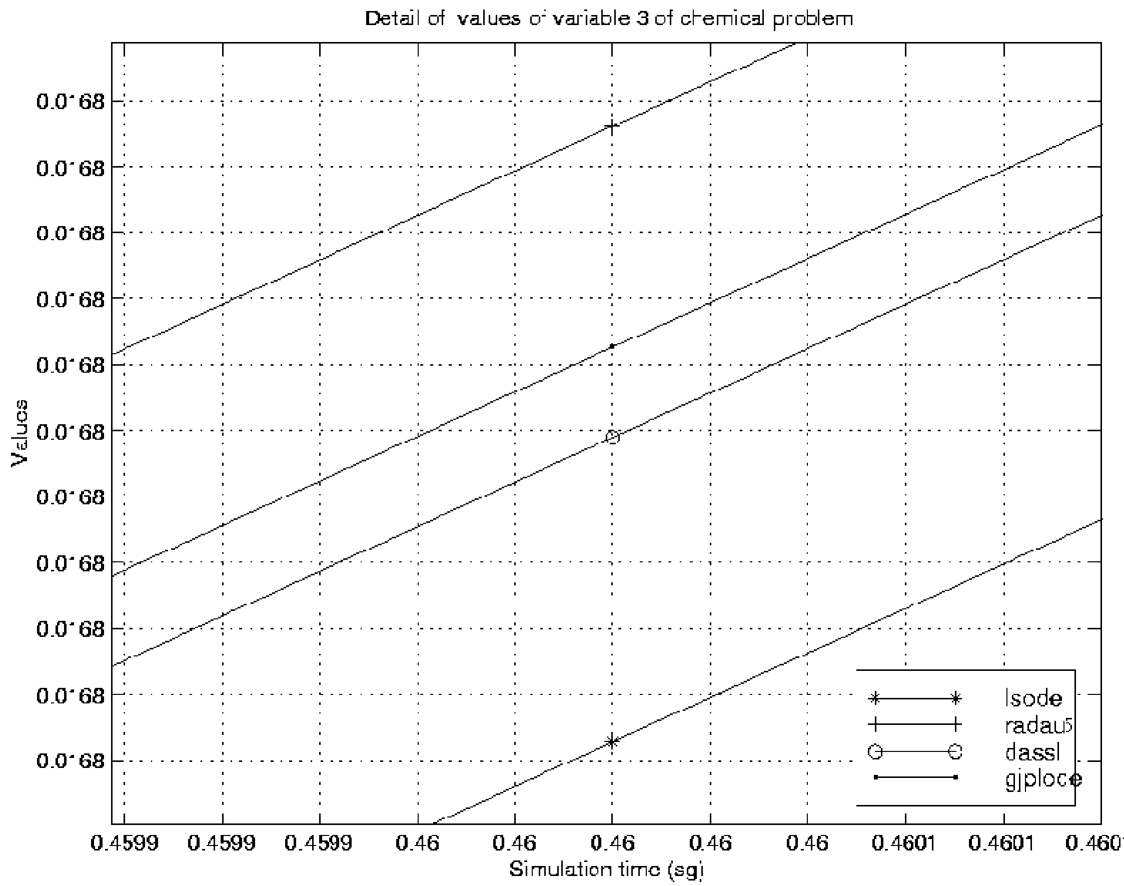

Figure 4: Comparison among the results obtained with several packages.

8 SLICOT: a useful tool in industry?

Model Based Process Control: A basic technology for reproducible and predictable process operation

8.1 Introduction

In the last decades a major change has started in process industry. Competition is drastically increased and environmental legislation has been significantly tightened. The strong growth in production capacity and the environmental concern have resulted in a market that has become complex quickly changing and saturated. One of the major reasons for these changes is the globalization of the market. As a result process industry is nowadays confronted with a strong competitive market. The market is developing from a supplier driven market to a demand driven market. These changes have far reaching consequences for these industries. In this market it becomes difficult to sell already produced products. Good trade-margins can only be obtained for products for which a momentary demand exists. Moreover customer dictated markets are capricious. Hence producers have to be able to quickly respond on the momentary market demands. More and more industries will be forced to operate their production facilities flexible. The ability to produce small series of a large variety of products at demand, on the existing production installations becomes a prerequisite to survive in this market.

Tight control of production processes over a broad operation range will become increasingly important. Process operation has to enable a fast completely *predictable* and *reproducible* change-over to different operating points that correspond to different raw materials and semi manufactured products, product types and different economic objectives (minimize costs, maximize production rate,...). Hence the freedom in process operation to produce specific products that have demand has to be used to predictably produce precisely what is asked, with the best company result. Or differently stated: From all potential operation scenarios that production strategy has to be selected that results in the best company results. For this selection a thorough knowledge of process behavior and process operation is needed. In this paper it will be discussed how process models and model based control systems can help industry to meet the new requirements. It is moreover discussed how the current generation of control systems has to be improved to indeed cope with these future requirements.

8.2 Model Based Control Concepts

Detailed knowledge and understanding of the behavior of the process forms the key to obtain the intended improved process operation. In general processes consist of a large number of manufacturing steps. In each of these production steps one or more in general quite diverse manipulations are performed on the raw materials or semi-manufactured products. Examples of these steps are separation of components (distillation, filtration, crystallization,...), mixing of components, sterilization, pasteurization, coocking, freezing, drying (evaporation, spray drying , freeze-drying), fermentation, chemical reactions,... All these manipulations are performed in process installations, which are able to create the proper conditions for a successful process operation. The sub-processes that take place in the installations have to fulfill certain requirements to ensure that the resulting products fulfill specifications. In each processing step a number of variables (e.g. residence time, temperature profile in the reactor, maximum temperature, concentration of the reactants and their purity, concentration of inerts, pressure, catalyst activity and so on) determine the course of the process. These critical process variables and a number of product properties have to be kept within specified tolerance limits, or have to be brought within these regions during a process change over, to guarantee product quality. These process variables are the so- called process *outputs or CV's* (controlled variables).

In order to keep the CV's in their predefined region, a number of process variables are available for manipulation, a predefined region, by the operator or control system. These manipulated variables, the so-called process *inputs or MV's*, are used at one hand to compensate for external disturbances and changes in the process behavior and at the other hand to change-over the operation point.

The third category of process variables is the so-called disturbance. Examples of these variables are impurities of the raw materials, composition of feeds to sub-processes, humidity, ambient temperature, and reactor fouling,... These variables influence the process like the manipulated variables. In contrast to the manipulated variables, disturbances can not be manipulated. Hence one just has to accept the presence of the disturbance and try to use the MV's to compensate for their effects. In the best case disturbances are measurable and predictable over a certain time horizon. In model based terminology these measurable disturbances are erroneously called the *DV's* (disturbance variables). Figure 5 gives a general overview of a process and the variables defined above. The process installations as well as the

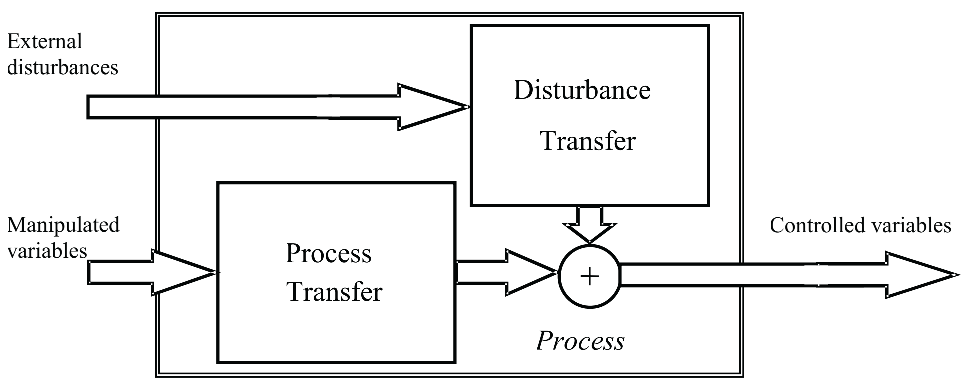

Figure 5: General representation of a process and the defined variables

processes that take place in the installation display in general inertia. This dynamic behavior of the process and installation can be described with a dynamic model. The step response is a well-known example of such a dynamic model. The step response is the response of process variables and product parameters on a step adjustment of a manipulated variable. Figure 6 gives an example of such a step response. The step response model is a specific model representation of the process dynamics. Other model types that represent dynamic behavior are impulse responses, transfer functions, Differential- Algebraic Equations (DAE's) and state space models.

Process models can within certain limits be used for simulation and prediction of the expected response of the process for arbitrary input signals, which are applied to manipulated variables and/or disturbance variables of the process. Consequently these models enable prediction of the process outputs in the near future on the basis of known adjustments on the manipulated

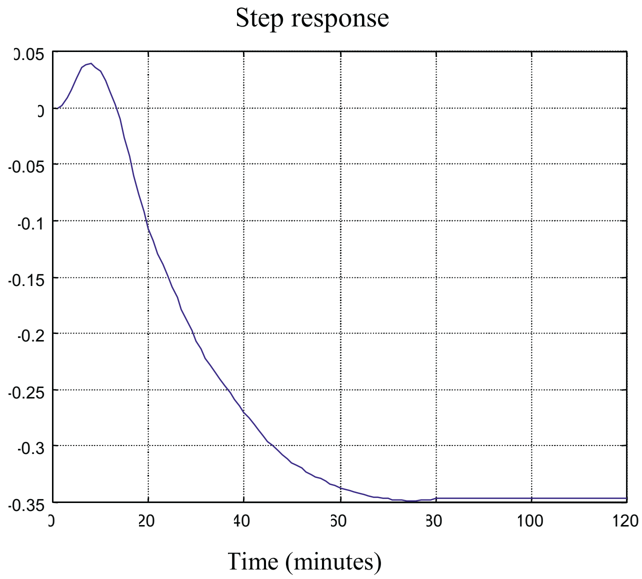

Figure 6: A step response representation of a process transfer

variables and disturbance variables in the recent past. The process models can also be used to determine which manipulated variable adjustments are to be applied to the process in the future to bring the process efficiently, i.e. in accordance with predefined goals, in a desired state. Hence models make process behavior predictable, controllable and optimizable. In fact model based control systems explicitly use the dynamic process behavior described by the models to determine the best possible control strategy under the given circumstances.

8.3 Model Predictive Control

Model Predictive Control (MPC) is a model-based controller well suited for control of multi-variable processes. Multivariable processes are processes whose inputs influence more process outputs at the same time. Characteristic for MPC is that the control strategy is determined at each new calculation of the control action. As a result MPC is very flexible for changing situations, like changing requirements, switching-off or failure of sensors and actuators. Moreover MPC can deal with constraint type of requirements, i.e. it can keep both manipulated as well as controlled variables in certain predefined regions, if feasible. MPC has been developed in industry, based on the need to operate processes with best achievable performance within the operational and physical limitations of the process and process installation. From its initial development [1, 2] MPC has grown nowadays to a standard technology in refinery and petrochemical industry to help operate processes such that the added value is maximized. For refineries this in general implies maximizing throughput of a certain product mix. The success of MPC within industry is for a major part due to the fact that MPC meets industrial requirements. These requirements can be roughly discerned in three hierarchical groups:

- *Operational requirements.* Processes have to be operated within a predefined region (Safety, wear, fooling,...)

- *Product Quality requirements.* Products have to fulfill certain specifications.

- *Economic requirements.* Products must be produced in such a way that the added value is maximized, without violation of the above limitations.

Figure 7 shows a block diagram of a Model Based Controller. Initially MPC did not take

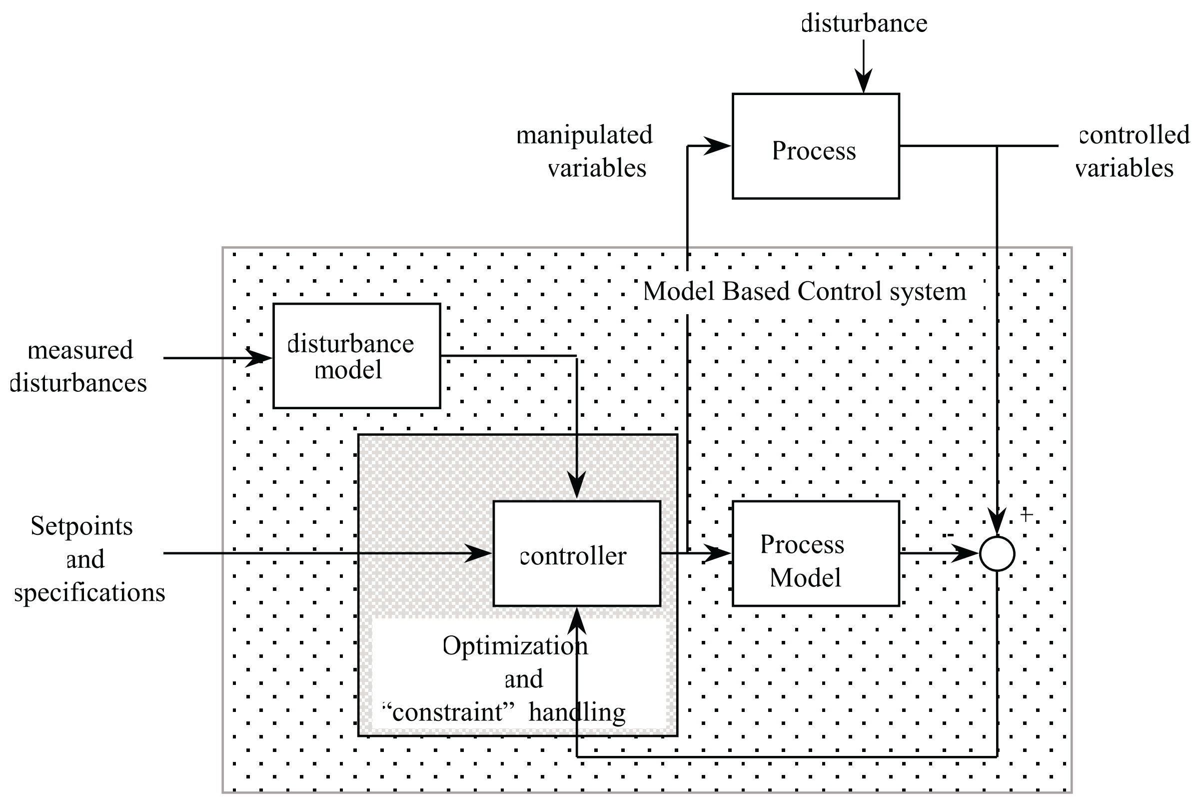

Figure 7: Schematic representation of a model based process control system

constraints into consideration explicitly. Later refinements of the technology allow constraints on both input and output variables to be considered in the formulation of the control strategy. The introduction of these constraints enables the definition of complex control hierarchies that closely resemble the discussed operational hierarchy. A recent paper of Qin en Badgewell [5] gives a good overview of the current generation of MPC technology applied in industry.

The basic principle of MPC can best be illustrated on the basis of the unconstrained situation. The finite impulse response (FIR) model, describing the dynamic behavior of a process with m inputs and p outputs, can be used to describe how input manipulations $u(t)$ applied to the process at discrete time instances in the past $t = k - i$, have on the process output $y(t)$ at the current discrete time instance $t = k$:

$$y(k) = \sum_{i=1}^{N} M_i u(k-i), \quad \text{where } M_i \in \Re^{p \times m}$$

are the so-called Markov parameters. Hence the FIR model can be used to describe the process output $y(t)$ at discrete time instances $t = k - i$ in the past. More interesting the model can also be used to describe the evolution of the process output $y(k + i)$ at discrete time instances in the future:

$$\begin{bmatrix} y(k) \\ y(k{+}1) \\ y(k{+}2) \\ y(k{+}3) \\ \vdots \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{\infty} M_i u(k-i) \\ \sum_{i=0}^{\infty} M_i u(k{+}1{-}i) \\ \sum_{i=0}^{\infty} M_i u(k{+}2{-}i) \\ \sum_{i=0}^{\infty} M_i u(k{+}3{-}i) \\ \vdots \end{bmatrix} = \begin{bmatrix} \dots & M_2 & M_1 & M_0 & 0 & 0 & 0 & 0 & \dots \\ \dots & M_3 & M_2 & M_1 & M_0 & 0 & 0 & 0 & \dots \\ \dots & M_4 & M_3 & M_2 & M_1 & M_0 & 0 & 0 & \dots \\ \dots & M_5 & M_4 & M_3 & M_2 & M_1 & M_0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \vdots \\ u(k{-}1) \\ u(k) \\ u(k{+}1) \\ \vdots \end{bmatrix}$$

Note that the future behavior of the process outputs is therefore determined by both the input manipulations applied to the process in the past ($u(k-i), i = 1, 2, \dots$) as well as by the future input manipulations ($u(k+j), j = 0, 1, 2, \dots$). Define $Y_{fp}(t, N_f, N_p)$ as the influence that past input manipulations over the horizon $[t - N_p, t - 1]$ have on the future outputs over the time horizon $[t, t + N_f]$ at time instant t. Define $Y_{ff}(t, N_f, N_c)$ as the influence that future input manipulations over the time horizon $[t, t + N_c]$ have on the outputs over the time horizon $[t, t + N_f]$. Hence the at time instant t predicted future behavior at the process outputs over the time horizon $[t, t + N_f]$, say $Y_f(t, N_f)$, is determined by:

$$\begin{aligned} Y_f(t, N_f) &= Y_{fp}(t, N_f, N_p) + Y_{ff}(t, N_f, N_c) \\ &= H(N_f, N_p)U_p(t, N_p) + T(N_f, N_c)U_f(t, N_c) \end{aligned}$$

where:

$$H(N_f, N_p) = \begin{bmatrix} M_{N_p} & \dots & M_2 & M_1 \\ M_{N_p+1} & \dots & M_3 & M_2 \\ \vdots & \vdots & \vdots & \vdots \\ M_{N_p+N_f-1} & \dots & M_{N_f+1} & M_{N_f} \end{bmatrix} \in \Re^{(N_f p)\times(N_p m)}$$

$$T(N_f, N_c) = \begin{bmatrix} M_0 & 0 & \dots & 0 \\ M_1 & M_0 & 0 & \vdots \\ \vdots & \vdots & \dots & 0 \\ M_{N_c-1} & M_{N_c-2} & \dots & M_0 \\ M_{N_c} & M_{N_c-1} & \dots & M_1 \\ \vdots & \vdots & \vdots & \vdots \\ M_{N_f-1} & M_{N_f-2} & \dots & M_{N_f-N_c-1} \end{bmatrix} \in \Re^{(N_f p)\times(N_c m)}$$

Three vectors, $Y_f(t, N_f) \in \Re^{(N_f p)\times 1}, U_p(t, N_p) \in \Re^{(N_p m)\times 1}$ and $U_f(t, N_c) \in \Re^{(N_c m)\times 1}$ are defined as:

$$\begin{aligned} Y_f(t, N_f) &= [y(t)\ y(t+1) \dots y(t + N_f - 1)]^T \\ U_p(t, N_p) &= [u(t - N_p) \dots u(t - 2)\ u(t - 1)]^T \\ U_f(t, N_c) &= [u(t) \dots u(t + N_c - 2)\ u(t + N_c - 1)]^T \end{aligned}$$

In MPC terminology the horizon $t + [0, N_f - 1]$ is called the prediction horizon. The control horizon equals the time horizon $t + [0, N_c - 1]$. The above distinction between the influence that past and future input manipulations have on the predicted future behavior of the process outputs is visualized in figure 8. This distinction between the influence that the past and future input manipulations have on the future outputs, respectively $Y_{fp}(t, N_f, N_p)$ and $Y_{ff}(t, N_f, N_c)$, is relevant for MPC since:

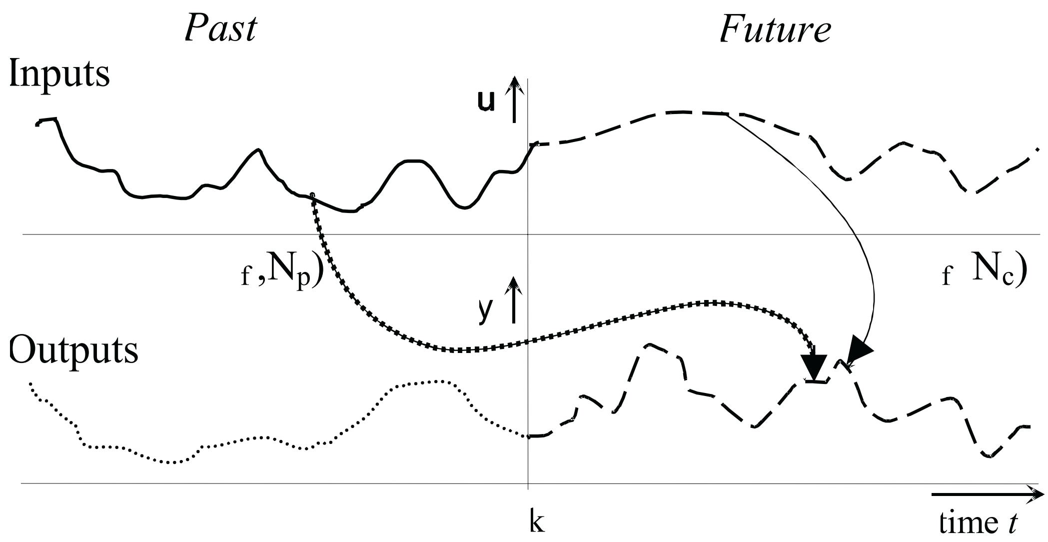

Figure 8: Relation between the past and future process inputs and the future process outputs.

- Past input manipulations have already been applied to the system and are therefore fixed.
- Future input manipulations have not yet been applied and are therefore still free to be chosen.

In MPC these future inputs are chosen such that the future behavior at the process outputs and inputs is as good as possible in accordance with the specified behavior for these variables. Hence the future input manipulations are the degrees of freedom in the optimization formulation.
The quadratic criterion function for this minimization is defined as:

$$\min_{U_f(t,N_c)} \{\|W_{sp}(Y_{ref}(t,N_f) - Y_f(t,N_f))\|_2^2 + \|\rho\Delta U_f(t,N_c)\|_2^2\}$$

with $\Delta U_f(t,N_c) = \begin{bmatrix} u(t) \\ u(t+1) \\ \vdots \\ u(t+N_c-1) \end{bmatrix} - \begin{bmatrix} u(t-1) \\ u(t) \\ \vdots \\ u(t+N_c-2) \end{bmatrix}$, $Y_{ref}(t,N_f)$ the desired process behavior over the prediction horizon, and W_{sp} and ρ are weighting functions used to tune the controller.

The above optimization problem is solved at each sampling instant, since at each subsequent sampling instant new information, i.e. measurements, comes available from the process. This information can be used to refine the solution. This is called the receding horizon principal. Hence although the input manipulations are determined over the complete control horizon, only the first sample, $u(t)$, is actually sent to the process. Note that the application of the *receding horizon* principle in MPC enables one to change the control specification at each sampling moment. This makes MPC very flexible and attractive. A second essential difference between MPC and other multivariable control technology is the ability to include constraints in the problem formulation. Constraints can be defined on inputs, outputs and additional variables that have a linear relation with the process inputs:

$$\min_{U_f(t)}\{\|W_{sp}(Y_{ref}(t)-Y_f(t))\|_2^2+\|\rho\Delta U_f(t)\|_2^2\}$$

subject to:

$$\begin{aligned}\alpha_L(i)\le u(t+i)\le\alpha_U(i) \quad &\text{for} \quad i=1,2,3,\dots\\ \gamma_L(i)\le \Delta u(t+i)\le\gamma_U(i) \quad &\text{for} \quad i=1,2,3,\dots\\ \beta_L(i)\le y(t+i)\le\beta_U(i) \quad &\text{for} \quad i=1,2,3,\dots\end{aligned}$$

In these expressions $\alpha_L, \beta_L, \gamma_L$ and $\alpha_U, \beta_U, \gamma_U$ represent the lower and upper limits defined on input variables and output variables and the change in the input variable between subsequent sample instances, respectively at each sample instant i over the prediction horizon. Constraints are frequently used to define the operational requirements, i.e. to define the operational region in which the process has to stay. A further refinement of the hierarchy can be obtained by recursive application of the above optimization. It is this combination of flexibility and the ability to define a control problem that closely resembles the actual operational problem that makes MPC that attractive for industry. The price one has to pay is that the approach results in a large optimization problem that has to be solved on-line at each sample instant.
Observe the dominant and explicit role the model plays in the formulation of MPC. In the prediction of the future it is possible to include disturbance models that describe the relation between measurable disturbances and the process outputs. Including the effect these disturbances will have on the future of the output behavior enables the optimization to account for their effect on the output in the calculation of the future input moves. In fact this is a feedforward control action, i.e. the controller already accounts for the effect before it actually occurs at the process output. Inclusion of these models in the controller may drastically improve the controller performance. Note however that the actual improvement is completely determined by the quality of the model. It will therefore not come as a surprise that the attainable performance of the controller is closely related to the quality of the process models.

8.4 Limitations of the current MPC technology

The current generation of MPC systems has a number of essential limitations that restrict their more general industrial applicability. These restrictions are caused at one hand by the way the criterion function is minimized. At the other hand the models currently applied in MPC and the way they are currently obtained also limit a more general industrial application. The first restriction is related to the fact that the solution of the criterion function subject to constraints over the complete future horizon at each subsequent sampling instant still requires significant computational power. A generally applied approach is to split the original formulation into two sub-problems: a steady state problem and a dynamic problem that are subsequently solved. The steady state problem is used to rigorously determine an optimal solution that fulfills all constraints and minimizes the criterion function in steady state. The solution for the input and output variables obtained from this optimization is then used as

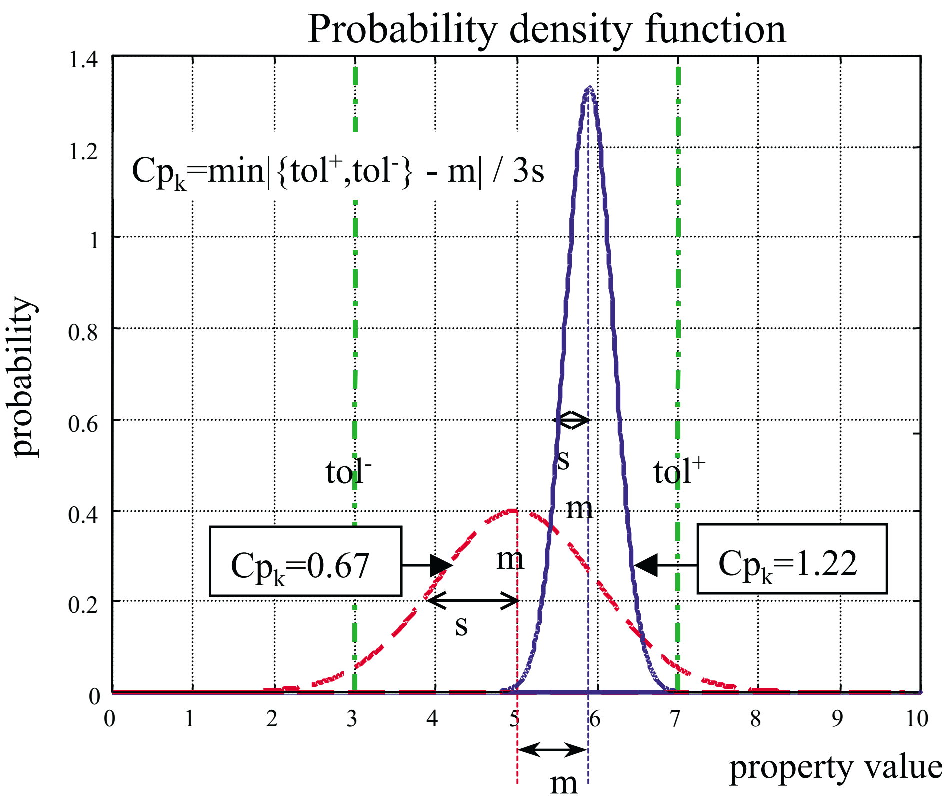

Figure 9: Optimization of the "capability" (Cpk) of important process variables and product parameters using model predictive control

targets for the dynamic optimization. The dynamic optimization has to bring the process variables from their current values to the defined targets. A rigorous implementation of this dynamic optimization problem is computationally demanding. It is therefore specifically in this step that a number of simplifications are applied. These simplifications may significantly limit the bandwidth of the controller and hence deteriorate the dynamic performance of the controller.

The second source that limits the performance are the models used in the current generation MPC and the identification techniques used to determine the models. The model types currently most frequently applied are finite step response models, finite impulse response models. In general the models describe only part of the process dynamics relevant for control. Only the low frequency behavior, i.e. the slow process responses and the steady state behavior process behavior, is well described. The restricted dynamic validity of the model is a direct consequence of the identification techniques and tests used. The fact that the models do not accurately describe the faster process dynamics relevant for. The restricted dynamic range of these models has a direct impact on the performance of the MPC. It limits the bandwidth of the MPC. Quality improvement of critical process variables and product properties is therefore restricted with the current MPC generation. This is important for problems where quality control, i.e. control of the so-called C_{pk} is an important objective (see figure 9).

Application of process identification requires extended on-site testing. The high costs related

to testing severely restrict the area of application of the current generation of MPC.
In process identification currently almost only linear dynamic models are used. Sometimes simple static non-linear functions at the input and output are used to still describe to a certain extend the non-linear process behavior. Current generation of MPC is therefore restricted in their ability to control non-linear process behavior, which is frequently observed:

- During fast change-overs between different operation points of the process
- In batch processes

8.5 Developments in MPC technology within IPCOS Technology

IPCOS Technology is a supplier of model predictive control technology that is widely applicable in processing industry. They have developed **INCA**, a range of product modules that enable efficient industrial application of model predictive control technology. IPCOS Technology is strongly innovation oriented. The aim of the company is to bring new technology to the market that enables a significant improvement of company results for their customers. **INCA** tools have a modular (object oriented) structure that makes it relatively easy to incorporate new developments in the product. Future versions of **INCA** will have to be able to cope with the above-discussed problems. Development therefore is targeted at enabling operation of processes at their physical, chemical, biological limits, such that the demands posed in the introduction with respect to flexibility, predictability and complete reproducibility of process operation conform predefined specifications is possible. In this field IPCOS Technology is cooperating in a number of international precompetitive research and development projects to achieve these objectives. These projects concentrate on the use of a more rigorous type of models for process operation and process control. It is expected that specifically hybrid models, i.e. models obtained from the integration of rigorous modeling and empirical process identification techniques, will be applied in future MPC systems. These hybrid models are expected not only to increase the accuracy, but also drastically reduce the cost of the modeling phase.
An important aspect of an industrial controller is its reliability. As a consequence the numerical routines to be applied in these general applicable controllers have to be efficient, accurate and also extremely robust and reliable. IPCOS Technology is a user and not a developer of basic numerical algorithms that are guaranteed to satisfy all of the above conditions. It is therefore important to have access to an easily accessible library of reliable implementations of basic control technology related numerical functions, for companies like IPCOS Technology. The NICONET initiative to develop a freely available library of basic control routines is therefore very valuable for not only suppliers of this technology, but also the process industry that applies this type of technology.
SLICOT routines are expected to be integrated in future versions of **INCA** modules. A number of basic routines from the library that have specific advantages above current implementations, e.g. exploit the Hankel or Toeplitz structure of matrices [6] or are more efficient, are expected to replace existing routines in future versions of the controller. A number of main SLICOT functions are evaluated for integration in **INCA**, e.g. observer technology, model reduction routines and subspace model identification.
Integration of observers in the model predictive controller enables a more realistic estimation of the disturbance and better control of marginally stable processes. Observers may improve the dynamic performance of MPC drastically. In IPCOS Technology a three step

identification method has been developed to obtain low order MIMO state space models that accurately describe the overall dynamics of a process [3, 4]. In this approach first a FIR model is estimated. In the second step the FIR model is used to obtain a low order state space model. This low order model serves as the initial model in a non-linear optimization to obtain the final state space model in the last step [3]. The model approximation applied in the second step turns out to be critical for the performance of the last step. It is expected that the performance of this step can be further improved, specifically for large and stiff systems. Different model reduction and estimation techniques, available in the library [8], are currently under evaluation for this purpose. A second approach under consideration is to replace the first two steps by a subspace identification technique [7]. Advantages of this approach are that it is very efficient (fast) and that it directly results in a state space model of low order. A potential disadvantage is that the optimization is not based on a criterion that has a direct relation with the use of the model, which makes the interpretation of the results potentially more difficult.

8.6 Concluding remarks and conclusions

This paper has given an overview of model predictive control, which is more and more applied in process industry. The current generation of MPC technology is in particular suited for refinery and petrochemical applications. The break-even point of these applications is in general reached within one year. Further development of the technology is needed to increase its applicability in smaller scale process industry. These developments have to result in MPC technology that enables design of:

- Robust high performance control systems that can reduce the variance of critical product parameters and process variables to desired level, (production at a desired Cpk value maximizing the added value).
- Control systems that perform changeovers from one operation point to another along a trajectory in a completely predictable and reproducible way (maximum flexibility with respect to product changes).
- Control systems based on a good balance between development and maintenance cost at one hand and profitability at the other hand.

A number of development and research projects are defined within IPCOS Technology to achieve these requirements in the future.
Numerical routines used in model predictive controller packages have to be reliable, accurate and efficient. The majority of companies do neither have the numerical expertise nor the means to develop implementations of these algorithms that are guaranteed to have the above properties. The NICONET initiative to develop a freely available library of basic control routines is therefore very valuable for not only suppliers of this technology, but also the process industry that applies this type of technology.

References

[1] J. Richalet, A. Rault, J.L. Testud and J. Papon. Model predictive heuristic control: Applications to Industrial processes. Automatica, Vol.14, 1978. pp.413-428.

[2] C.R. Cutler and B.L. Ramaker. Dynamic matrix control - a computer control algorithm. Proceedings of the joint Automatic Control Conference, 1980, paper WP5-B. pp.1-6.

[3] T.Backx and A. Damen. Identification of industrial MIMO processes for fixed controllers. Part 1: Journal A Vol.30, no.1, 1989, pg.3-12. Part 2: Journal A Vol.30, no.2, 1989, pg.33-43.

[4] H.M. Falkus. Parametric Uncerntainty in System Identification. Phd. Thesis Eindhoven University of Technology, 1994.

[5] S.J. Qin and T.A. Badgwell. An overview of industrial model precictive control technology. Preprints of the proceedings of the Chemical Process Control-V , Tahoe City 7-12 January, 1996.

[6] P. van Dooren. Selection of basic software tools for structured matrix decompositions and perturbations. NICONET Report 1999-9, May 1999.

[7] B.R.J. Haverkamp.Efficient implementation of Subspace method identification Algorithms. NICONET Report 1999-3, March 1999.

[8] A. Varga. Model Reduction Routines for Slicot. NICONET Report 1999-8, May 1999.

Jobert Ludlage[4] *and Ton Backx*[5]

[4]IPCOS Technology, Bosscheweg 145a, NL-5282 WV Boxtel, The Netherlands

[5]IPCOS Technology, Bosscheweg 145a, NL-5282 WV Boxtel, The Netherlands , Eindhoven University of Technology, Faculty of Electrical Engineering, Group Measurement and Control systems, P.O. box 513, NL-5600 MB Eindhoven, The Netherlands.

9 Highlights of the second NICONET workshop at INRIA-Rocquencourt, France

The second NICONET workshop was held on Friday, December 3, 1999, at Rocquencourt, France, organized by INRIA, one of the NICONET partners. The objectives of this second workshop were

- To promote the advantages of using the numerical software library SLICOT in control engineering and industrial problems.
- To investigate the current interest of users in numerical software tools.
- To get feedback on good candidate areas (those with high cost, large problem sizes, real-time constraints,...) for applying SLICOT.
- To demonstrate the feasibility of using SLICOT in industrial applications.

The workshop has been attended by about fifty participants, coming from various areas: engineers, mathematicians, computer scientists and practitioners from industry. The attendance for the meeting was totally free. Attendees received a proceedings booklet containing extended abstracts and full papers.

The workshop included 3 plenary sessions on advanced topics in control.

- *SLICOT, introductory presentation*, A. van den Boom and V. Sima: 9h-10h
- *A large industrial application*, E. Demay from EDF: 10h-10h30
- *SLICOT, presentation of specific topics with toolbox demos*: 11h-13h
  - Topic I: Basic software tools: 11h-11h30
  - Topic II: Model reduction: 11h30-12h
  - Topic III: Subspace identification: 12h-12h30
  - Topic IV: Robust control: 12h30-13h
- *Industrial use of SLICOT presented by Niconet partners*: 14h-15h30
  - H_∞ control presented by A. Coville, SFIM, France.
  - Subspace identification of Multi Axis Durability Test Rigs using SLICOT presented by J. De Cuyper,LMS, Belgium.
  - A steel cooling application presented by R. Wohlgemuth, TBZ Pariv, Chemnitz.

The final part of the program was devoted to demos and poster presentations. The poster and demos session included 23 presentations, most of them related to SLICOT:

- **E.A.Antunez, V.H. Garcia, I.B. Espert, J.J.I. Gonzalez**: "Non singular jacobian free piecewise linearization of ODE"
- **P. Benner, E.S. Quintana-Orti, G.Quintana-Orti** : "Efficient balanced truncation model reduction methods using the matrix sign function"

- **S. Bingulac, N.F. Al-Muthairi**: "Determination of Markov parameters directly from noise free input-output samples of MIMO systems"
- **R. Chmurny, J. Stein**: "Evaluation of frequency response function of harmonically exited dynamic System"
- **K. Dackland, B. Kågström**: "Blocked algorithms for reduction of a regular matrix pair to generalized Schur form"
- **F. Delebecque**: "The SCILAB SLICOT environment"
- **E. Elmroth, P. Johansson, B. Kågström**: "StratiGraph – computation and presentation of graphs displaying hierarchies of Jordan and Kronecker structures"
- **H. Fassbender, P. Benner** : "SLICOT drives tractors"
- **J. Garloff**: "Software for solving robust performance problems based on Bernstein expansion"
- **Y. Genin, Y. Hacker,Y. Nesterov, P. Van Dooren**: "Convex optimization over positive polynomials"
- **Y. Genin, R. Stefan, P. Van Dooren**: "Real stability radii of polynomial matrices"
- **D.W. Gu, P.H. Petkov, M.M. Konstantinov**: "H-infinity loop shaping design procedure routines in SLICOT"
- **D.W. Gu, P.H. Petkov, M.M. Konstantinov, V. Mehrmann**: "Comparison of Riccati equation solvers in Matlab and SLICOT"
- **I. Ioslovich**: "Numerical software for redundancy determination and presolving analysis of large scale LP problems, using MATLAB5.2"
- **I. Jonsson and B. Kågström**: "Recursive blocked algorithms for solving triangular Sylvester-type matrix equations"
- **N. Mastronardi, S. Van Huffel, P. Van Dooren**: "Fast implementation and stability property of the QR factorization in subspace identification"
- **P.M. Ndiaye, S. Steer** : "Model reduction of large systems with SLICOT"
- **R.N. Nikoukhah**: "The SCILAB SCICOS simulation environment"
- **J.P. Quadrat**: "Dynamic programming and MAX+ algebra in SCILAB"
- **A. Varga**: "A SLICOT-based descriptor system toolbox for MATLAB"
- **V.Sima** : "Accurate computation of eigenvalues and real Schur form of 2x2 real matrices"
- **V.Sima** : "Cholesky or QR factorization for data compression in subspace-based identification"
- **D. Kressner**: "Web Computing with Slicot"

- **A. Steinbrecher**: "Slicot in Robotics"

A panel discussion in which the partners of NICONET exposed their opinions and needs for control problems ended the workshop . Copies of the proceedings are available upon request from INRIA[6]. The next workshop will be organized in Louvain-la-Neuve, Belgium.

Francois Delebecque, Serge Steer and Sabine Van Huffel

[6]Christine.Bren@inria.fr

10 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. Recent updates of the library are also described. 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 found from the NICONET homepage World Wide Web URL

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

10.1 Electronic Access to the 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 example program, example data, execution results, the associated `.html` file, as well as the source code for the called SLICOT routines.

The entire library is contained in a file, called `slicot.tar.gz`, in the SLICOT root directory `/pub/WGS/SLICOT/`. The tree structure of the subdirectories created after applying

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

and

```
tar xvf slicot.tar
```

and their contents is summarized below:

`slicot`	contains the files `libindex.html`, `make.inc`, `makefile`, and the following subdirectories:
`benchmark_data`	contains benchmark data files for Fortran benchmark routines (`.dat`);
`doc`	contains SLICOT documentation files for routines (`.html`);
`examples`	contains SLICOT example programs, data, and results (`.f`, `.dat`, `.res`), and `makefile`, for compiling, linking and executing these programs;
`src`	contains SLICOT source files for routines (`.f`), and `makefile`, for compiling all routines and creating an object library;
`SLTools`	contains MATLAB `.m` files, mex files (optionally), and data `.mat` files (optionally);
`SLmex`	contains Fortran source codes for mexfiles (`.f`).

Note that the tree structure has been changed from what has been used previously, in order to better separate the various files of the SLICOT large collection.

Another, similarly organized file, called `slicotPC.zip` (previously called `slicotPC.tar.gz`), is available in the SLICOT root directory; it contains the MS-DOS version of the source codes of the SLICOT Library, and can be used on Windows 9x or NT platforms.

Currently, there are no makefiles for the PC version, and no executable MATLAB files (e.g., `.dll` files) have been included. Executable MATLAB files for Windows platforms may, however, be separately taken from the ftp site, subdirectory `MatlabTools/Windows/SLToolboxes` of the SLICOT root directory.

After downloading and decompressing the appropriate SLICOT archive, the user can then browse through the documentation on his local machine, starting from the index file `libindex.html` from `slicot` subdirectory.

10.2 SLICOT Library updates in the period July 1999 — December 1999

There have been three SLICOT Library updates during the period July 1999 — December 1999: on September 1, November 27, and December 18. Each time, known bugs have been corrected out. These resulted in updating 8 routines and 2 example programs. Details are given in the files `Release.Notes` and `Release3.History`, located in the directory `pub/WGS/SLICOT/` of the ftp site. Several changes have also occured in the documenting comments of some routines.

On November 27, 1999, Release 4.0 of the SLICOT Library has been announced. All library source files then contained the Release 4.0 statement notice. Several routines from SLICOT Release 3.0, together to the associated documentation files, example program files, data and results, have been removed. The removed routines have better counterparts, and their removal has been announced in the previous versions of the file `Release.Notes`; see the file `Release3.History`, which describes the history of all updates performed in the Release 3.0 since October 1997. A table indicating the removed routines, as well as the functionally equivalent routines, is included in `Release3.History`.

Several new user-callable routines for basic and robust control problems have been made available on the ftp site in the period July 1999 — December 1999. They include *Analysis Routines*, *Mathematical Routines*, *Synthesis Routines*, and *Transformation Routines*, performing the following main computational tasks:

- H_∞ norm of a continuous-time stable system;
- distance from a real matrix to the nearest complex matrix with an eigenvalue on the imaginary axis, using either bisection or bisection and SVD;
- extracting, from a system pencil $S(\lambda)$, a regular pencil having the finite Smith zeros of $S(\lambda)$ as generalized eigenvalues;
- fast recursive least-squares filter using a QR-decomposition based approach;
- minimum-norm solution to a linear least squares problem, given a rank-revealing QR factorization;
- minimum norm least squares solution of $\mathrm{op}(R)X = \alpha B$, or $X\mathrm{op}(R) = \alpha B$, with R upper triangular, using singular value decomposition ($\mathrm{op}(R)$ is R or R^T);

- H_∞ (sub)optimal state controller for a continuous-time or for a discrete-time system;
- H_2 optimal state controller for a continuous-time or for a discrete-time system;
- closed-loop system matrices for a system with robust controller;
- normalization of a system for H_∞ controller design;
- state feedback and output injection matrices for an H_∞ (sub)optimal state controller (continuous-time);
- H_∞ (sub)optimal controller matrices using state feedback and output injection matrices (continuous-time);
- H_2 optimal controller matrices for a normalized discrete-time system;
- H_2 optimal controller matrices for a discrete-time system;
- normalization of a system for H_2 controller design;
- state feedback and output injection matrices for an H_2 optimal state controller (continuous-time);
- H_2 optimal controller matrices using state feedback and output injection matrices (continuous-time);
- balancing the matrices of the system pencil corresponding to a descriptor triple $(A - \lambda E, B, C)$;
- orthogonal reduction of a descriptor system pair $(A - \lambda E, B)$ to the QR-coordinate form;
- orthogonal reduction of a descriptor system pair $(C, A - \lambda E)$ to the RQ-coordinate form;
- orthogonal reduction of a descriptor system $(A - \lambda E, B, C)$ to an SVD coordinate form;
- orthogonal reduction of a descriptor system $(A - \lambda E, B, C)$ to an SVD-like coordinate form;
- orthogonal reduction of a descriptor system to the controllability staircase form;
- orthogonal reduction of a descriptor system to a system with the same transfer-function matrix and with no uncontrollable finite eigenvalues;
- orthogonal reduction of a descriptor system to the observability staircase form;
- finding a reduced (controllable, observable, or irreducible) descriptor representation $(A_r - \lambda E_r, B_r, C_r)$ for an original descriptor representation $(A - \lambda E, B, C)$.

In addition, nine new mexfiles and several MATLAB `.m` files have been added in the subdirectory `MatlabTools/Windows` (previously called `Mexfiles/PC`), and its subdirectories. They can be used for the solution of standard and generalized Sylvester and Lyapunov equations, or of Riccati equations, for computing system transformations, or canonical forms, as well as for designing H_∞ (sub)optimal and H_2 optimal state controllers for continuous-time and discrete-time systems, and for computing the H_∞ norm of a continuous-time stable system. These mexfiles can be directly called on Windows 9x or NT platforms.

The latest changes in the library contents or routine updates, as well as in the ftp site organization, are announced in the file `Release.Notes`, located in directory `pub/WGS/SLICOT/` on the WGS ftp site. Previous updates of the current release are described, in reverse chronological order, in the file `Release.History`, at the same address. The history of all the changes performed in the Release 3.0 since October 1997 are listed in the file `Release3.History`.

The tree structure of the ftp site has been slightly changed at the latest SLICOT update, by adding three subdirectories and several subsubdirectories, and deleting other two subdirectories. The new subdirectories and their contents are listed below.

`FD/`	Fast Recursive Least Squares Filters.
`FD/FD01/`	
`MatlabTools/`	MATLAB files and mexfiles, contained in its subdirectories:
`MatlabTools/Unix/`	MATLAB files and mexfiles for Unix platforms.
`MatlabTools/Unix/SLTools/`	MATLAB source and data files (`.m` and `.mat`).
`MatlabTools/Unix/SLmex/`	Fortran source codes for MATLAB mexfiles based on SLICOT library (`.f`).
`MatlabTools/Windows/`	MATLAB files and mexfiles for Windows platforms.
`MatlabTools/Windows/SLToolboxes/`	MATLAB executable files (`.dll`) grouped in several archives, according to their functionality.
`MatlabTools/Windows/SLTools/`	MATLAB source and data files (`.m` and `.mat`).
`MatlabTools/Windows/SLdemos/`	MATLAB source and executable files (`.m`, `.mat`, and `.dll`), for a GUI-based demo.
`MatlabTools/Windows/SLmex/`	Fortran source codes for MATLAB mexfiles based on SLICOT library (`.f`).
`contrib/`	routines which were previously proposed for being included in the SLICOT Library, but which could not be included yet. These routines do not follow the latest SLICOT Implementation and Documentation Standards, but could be of interest to some users. Included are G. Miminis' routines for solving the Pole Placement Single-input or Multi-input problem, and A.J. Geurts and C. Praagman's routines for computing a unimodular polynomial matrix $U(s)$ such that $R(s) = P(s)U(s)$ is column reduced, given a polynomial matrix $P(s)$.

Note that the former subdirectory `Mexfiles/` (under `SLICOT/` directory) and all its subsubdirectories have been removed, their contents being replaced by the contents of the new subdirectory `MatlabTools/` and its subsubdirectories. In addition, the subdirectory `LAPACK_N/` of `/pub/WGS/SLICOT/`, containing used LAPACK files not included in LAPACK Release 2.0, has been removed, because the needed files are now included in the LAPACK Release 3.0.

10.3 New NICONET Reports

Recent NICONET reports (available after July 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:

- Petko Petkov, Mihail Konstantinov, Da-Wei Gu and Volker Mehrmann. *Numerical solution of matrix Riccati equations: a comparison of six solvers* (file `nic1999-10.ps.Z`).

  This report presents results from the evaluation of six solvers intended for the numerical solution of continuous-time matrix algebraic Riccati equations. The solvers include the MATLAB functions from different toolboxes and two Fortran 77 solvers developed by the authors. The comparison implements two benchmark problems each comprising 1600 6-th order Riccati equations with known solutions. For each solver and each equation the relative forward and backward errors are computed, and, for two of the solvers, the accuracy of condition and error estimates is investigated. Some conclusions concerning the numerical behaviour of the solvers are given.

- Volker Mehrmann, Vasile Sima, Andras Varga and Hongguo Xu. *A* MATLAB*MEX-file environment of SLICOT* (file `SLWN1999-11.ps.Z`).

  Several MEX-files are developed based on SLICOT Fortran subroutines. The MEX-files provide new tools for the numerical solution of some classical control problems, such as the solution of linear or Riccati matrix equations computations in the MATLAB environment. Numerical tests show that the resulting MEX-files are equally accurate and much more efficient than the corresponding MATLAB functions in the Control System Toolbox and the Robust Control Toolbox. In order to increase user-friendlyness the related m-files are also developed so that the MEX-file interface to the corresponding SLICOT routines can be implemented directly and easily.

- Da-Wei Gu, Petko Hr. Petkov and Mihail Konstantinov. *H_∞ and H_2 optimization toolbox in SLICOT* (file `SLWN1999-12.ps.Z`).

  This report summarizes the progress made in the sub-task IV.A of the NICONET project. Selected routines to implement H_∞ and H_2 (sub) optimization syntheses are listed, which have all been standardized and included in the SLICOT package. The integration of those routines in MATLAB has also been completed; the mex files are attached in the appendices. This report discusses the selection and testing of benchmark problems with regard to the developed routines, and the comparisons made between these routines and others available in MATLAB. In particular, two industrial benchmark case studies, namely the controller design of a Bell 205 helicopter and a distillation column design, are introduced and the design results, obtained using the developed routines, are analysed.

- Anton Stoorvogel. *Numerical problems in robust and H_∞ optimal control* (file `nic1999-13.ps.Z`).

  After formulating the H_∞ control problem for linear, time-invariant and finite-dimensional systems, the difficulties in the computation of the optimal performance are discussed, as well as the problems encountered in computing controllers.

- Jörn Abels and Peter Benner. *CAREX — A Collection of Benchmark Examples for Continuous-Time Algebraic Riccati Equations (Version 2.0)* (file `SLWN1999-14.ps.Z`).

  A collection of benchmark examples is presented for the numerical solution of continuous-time algebraic Riccati equations. This collection may serve for testing purposes in the construction of new numerical methods, but may also be used as a reference set for the comparison of methods. The collected examples focus mainly on applications in linear-quadratic optimal control theory. This version updates an earlier benchmark collection and includes one new example.

- D.W. Gu, P.Hr. Petkov and M.M. Konstantinov. *H_∞ Loop Shaping Design Procedure Routines in SLICOT* (file `nic1999-15.ps.Z`).

  This report briefly introduces the H_∞ Loop Shaping Design Procedure (LSDP) and its implementation in the software package SLICOT. The developed routines are tested in a design example and are included as appendices.

- Jörn Abels and Peter Benner. *DAREX — A Collection of Benchmark Examples for Discrete-Time Algebraic Riccati Equations (Version 2.0)* (file `SLWN1999-16.ps.Z`).

  This is the second part of a collection of benchmark examples for the numerical solution of algebraic Riccati equations. After presenting examples for the continuous-time case in Part I (CAREX), our concern in this paper is discrete-time algebraic Riccati equations. This collection may serve for testing purposes in the construction of new numerical methods, but may also be used as a reference set for the comparison of methods. This version updates an earlier benchmark collection. Some of the examples have been extended by incorporating parameters and there have been some new additions to the collection.

- Andras Varga and Paul Van Dooren. *Summary report of topic I.A.* (file `SLWN1999-17.ps.Z`).

  This report surveys the deliverables of Task I.A. of the NICONET project. A brief description of the control problems that are solved by the basic numerical tools developed in this Task is first given, and the different routines of SLICOT that correspond to these control problems, which are available via ftp, are listed. Then, the toolboxes that give interactive access via MATLAB or Scilab to those routines, as well as the benchmark problems for this Task, are described. Finally, a few numerical examples exhibiting the accuracy and speed of the new tools are given and a demo for the routines of this Task is described.

- Andras Varga. *Task II.B.1 - Selection of Software for Controller Reduction* (file `SLWN1999-18.ps.Z`).

  This working note presents a short overview of methods suitable for controller reduction. A first class of methods considered are general purpose methods for reduction of unstable systems, as for example, absolute and relative error methods or frequency weighted methods, both in combination with modal separation or coprime factorization techniques. Special frequency weighted controller reduction methods able to preserve closed-loop stability and even closed-loop performance are also discussed. A selection of user callable and supporting routines to be implemented for controller reduction is proposed. The new routines will be included in the SLICOT library.

- Ad van den Boom and Ton Backx. *Benchmarks for Identification* (file `nic1999-19.ps.Z`, soon available).

- Michel Verhaegen. *Symbolic and computational pre-processing in physical parameter estimation of multi-body mechanical systems* (file `SLWN1999-20.ps.Z`).

  The objective of this note is to highlight the scope and computational (symbolic and/or arithmetic) tasks of turning a physical parameter estimation problem into a (constraint) optimization problem. Concrete examples show the need for symbolic (object-oriented) modeling environments for defining the structure of the physical system to be used in the parameter optimization step. Without this (interactive) software environment for compiling a physical parameter estimation problem into an optimization problem, standardization of commercial optimization routines is of little or no interest.

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

10.4 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, include the following:

- The 3rd Mathematical Modelling (MATHMOD) conference, Vienna, Austria, February 2–4, 2000.

- AspenWorld 2000, Orlando, USA, February 7–11, 2000.

- Mathematical Theory of Networks and Systems (MTNS 2000), Perpignan, France, June 19–23, 2000.

- UKACC International Conference CONTROL 2000, University of Cambridge, United Kingdom, September 4–7, 2000.

- IEEE International Symposium on Computer-Aided Control System Design, CACSD 2000, Anchorage, Alaska, USA, September 25–27, 2000.

Vasile Sima