91-44

International Federation of Automatic Control

COMPUTER AIDED DESIGN IN CONTROL SYSTEMS

Preprints of the IFAC Symposium, Swansea, UK
15-17 July 1991

Edited by

H.A. BARKER
University of Wales, Swansea, UK

PREPRINTS

PERGAMON PRESS

SLICOT,

A SUBROUTINE LIBRARY IN CONTROL AND SYSTEMS THEORY.

Ad van den Boom [1], Alan Brown [2], Frits Dumortier [3], Aloys Geurts [1], Sven Hammarling [2], Ruth Kool [1], Michel Vanbegin [4], Paul Van Dooren [4] and Sabine Van Huffel [5].

Affiliations:

1) Eindhoven University of Technology, Eindhoven, NL
2) Numerical Algorithms Group (NAG) Ltd., Oxford, UK
3) State University of Ghent, Gent, B
4) Philips Research Laboratory Belgium, Louvain la Neuve, B
5) Katholieke Universiteit Leuven, Leuven, B

Abstract

Past initiatives with respect to realising control libraries are analyzed. A new collaborative initiative is introduced. The design of a new control library SLICOT is presented and the first release of this library is discussed.

Keywords

Computer-aided control system design, interactive computing, control engineering computer applications, control libraries.

1. Introduction and history

In recent years there has been an increasing interaction between the fields of numerical linear algebra and of control and systems theory. System theorists had to turn their attention to advanced numerical software for solving their problems as they found that their straightforward methods did not work well when applied on harder problems such as large scale systems and ill-posed problems. Numerical analysts have started to focus on the linear algebra problems arising in linear systems theory. This cross-fertilisation has resulted in a considerable number of novel techniques both in terms of speed and accuracy; see Van Dooren (1989). But many problems are still waiting for an adequate algorithm which is both fast and accurate and at the same time respects the structure of the problem. The growing number of people involved in this interdisciplinary area of research shows the relevance of the area. A more explicit indication of this is the importance that the field has acquired in industrial environments: advanced control techniques are not only being applied in production processes but are also built into actual products such as compact discs and medical equipment.

User-friendly and powerful tools for the analysis and design of complex systems became available with the advent of CACSD programs in the last decade; see Van den Boom (1988) and Rimvall c.s. (1985). Such a CACSD tool is a complicated hierarchical software structure, where the libraries containing the mathematical and control algorithms are usually placed on lower levels of the hierarchy. Moreover, the libraries constitute only a small part of the total software volume of the CACSD program. Nevertheless the quality of the outcome of the CACSD program is fully determined by the quality of these libraries.

Over the last ten years several efforts have been initiated to produce software libraries of basic subroutines dealing with various analysis or design problems in control and systems theory. The Scandinavian control library (Elmqvist c.s., 1976; Wieslander, 1977) and the Swiss control library AUTLIB (Cellier c.s., 1977) were the first initiatives in the field. However, these libraries did not reach maturity. A British cooperation led to the realization of the control library SLICE which contains a set of almost 40 control routines, see Denham c.s. (1981), Denham (1985) and Williams (1986). A revised version of SLICE was issued by NAG in 1987. The Benelux Working Group on Software (WGS) has realised the prototype control library SYCOT, see WGS (1983). The libraries SLICE and SYCOT form the basis of the SLICOT library to be discussed in the sequel. The German control library RASP offers about 250 subroutines classified in 7 chapters, mainly for control, and some 20 main programs for several chapters of control and systems theory; see Grübel (1983) and Grübel c.s. (1990). RASP is a large library and has indeed reached maturity.

RASP relies heavily on its main programs, which include the interactive facilities. RASP has found ample application in industry in Germany. A Rumanian development is the control library BIMASC (Varga, 1983 and Varga c.s. 1985) which is based on the mathematical library BIMAS (Varga c.s. 1982). BIMASC provides many subroutines for analysis, modelling, simulation and design of linear multivariable systems. A Bulgarian development in this field is LISPACK. An inventory of all subroutines contained in 16 libraries and packages for solving problems in control and systems theory has been issued by the Benelux Working Group on Software (WGS); see WGS (1985). Most of these libraries are incomplete (with respect to the large field of control theory) and mostly of limited scope, i.e. covering mainly the specific research interests of the developers. As a rule these developments are not ongoing activities since there are no plans for maintenance and extension of these libraries.

2. Motivation behind SLICOT

The short history of the development of control libraries given above reveals the importance that has been given to the topic at various moments in time and at different places. Nevertheless the large amount of work that has been devoted to developing useful control subroutines has not resulted so far in a coherent entity. This is due to several factors which will be discussed here briefly.

First of all, the developers of the control libraries were (mostly excellent) specialists in a limited part of the whole control field. The high quality of their methods and related software caused much interest from elsewhere, resulting in a (often small) specialised control library, sometimes with additional main programs. Only a few of these libraries were extended to cover a wider area of the whole field. The motivation to develop routines for (already existing) methods and areas of the control field was low. This is quite understandable as the effort involved in such an extension is very often at the expense of further research by the group in the original field, which has proved to be successful so far. Moreover, many of the activities are more or less a duplication (in some modified form) of work that has already been done elsewhere. It is obvious that the amount of work associated with realising a control library, which covers the whole field of control and systems theory, is an immense task. Such a task exceeds by far the capabilities of a single systems and control group at universities. This strongly suggests an organisational set-up where several groups are cooperating in such a way that they are contributing algorithms and prototype software, especially in their own, specific fields of competence.

Further, most of these control libraries were based on own mathematical libraries or a collection of ad hoc written routines rather than standard mathematical libraries such as EISPACK and BLAS. The work associated with the development of basic mathematical routines on an ad hoc basis did increase the task even more. In general, control engineers do not consider themselves to be the specialists in basic mathematical software. Consequently the need for developing a substantial amount of basic mathematical software did not stimulate starting a project as discussed here.

Another reason is that a control library as such was only of interest to a limited number of very specific users, namely those who were themselves skilled in developing software for their own use mainly at research institutes of universities and large industries. This implied a relatively small potential market for the control library, which meant a limited interest in commercialising the product. In the meantime, several user-friendly interactive CACSD packages entered the market. These packages were very attractive for a much larger group of users. This was mainly due to their flexible interactive facilities and the fact that no programming skill is required to work with it. Nowadays, some of these packages, like MATLAB, are intensively used by people with different levels of skill, working on problems with various levels of complexity at many institutions.

Due to the success of some of these interactive packages and the fact that many people have familiarised themselves with interactive packages, the interest in control libraries was low for a while. However it is becoming apparent now that, in particular for research purposes, still more flexibility is needed with respect to choice of methods, of course without loss of the nice interactive facilities of the CACSD package. Therefore, interfacing software, which allows a flexible and user-friendly combination of (control) libraries and powerful interactive packages, is becoming an attractive issue now.

An example of this is the MATLAB Gateway Compiler (MGC) as described in Renes (1990) and Renes c.s. (1991). The use of such software has also attractive advantages with respect to accuracy and speed. Such an interface strongly renews the interest in control libraries. This will give control libraries their own place among CACSD packages.

In spite of the advent of CACSD packages the importance of a control library was recognised by WGS and NAG and led to a collaboration based on each other's experience and expertise. Such a collaboration is meant to combine expertise on control issues and expertise on development, maintenance and distribution of scientific software.

3. SLICOT Implementation and Documentation Standards

Essential characteristics that turn a collection of routines in a subroutine library are that each routine is well documented and that routines and their documentation are uniformly shaped. Uniformity facilitates the use of the routines in the library and thus fulfils one of the aims of a library, i.e. economization of the user's effort to solve a particular problem with algorithmic tools. Uniformity means that routines and documentation are written conform whether or not explicitly formulated standards. To establish a standard and subsequently to meet it is for an individual programmer

a tough task which demands a great deal of discipline. This applies even more to a group cooperating on a library project. In the latter case, explicitly formulated implementation and documentation standards are mandatory. This holds particularly for SLICOT: the first release has been realised in cooperation of NAG and WGS, and for future extensions the designers aim at cooperating with external contributors. In parallel with the development of the first release of SLICOT, standards have been established and finally formulated in a report; see WGS (1990). In this section we take the opportunity to comment on some of the chosen directives.

3.1. Naming conventions

Mnemonic suggestive names have the advantage of easy recognition. A disadvantage is that with a growing number of routines it is difficult if not impossible to maintain such naming without running into conflicts. On the other hand it is common experience that with frequent use the actual name does not play an important role. That is why for SLICOT, aiming at a large number of routines, a naming convention similar to the NAG library has been chosen. The experience of NAG Library is that it works satisfactorily, possibly after a period of habituation for the users. A minor difference between the NAG and the SLICOT naming convention is that in SLICOT the first character is an abbreviation of the name of the chapter, so the names contain a bit of mnemonics.

3.2. Argument list

A common external structure of the routines is realised by a uniform order of the types of the arguments in the argument list. For SLICOT routines this order is strictly prescribed. Contravention of this rule is possible though if this is logically obvious.

3.3. Routine termination

A major point for the designers of SLICOT has been the absolute demand that a routine never ends with a dynamic stop. Termination of a routine must always take place regularly with an error flag set in case of an unsuccessful call. Consequently the set of possible values for the input parameters must be defined exactly in the documentation and the routine must check the validity of as many arguments as possible, before attempting to solve the user's problem. This holds particularly for array arguments of which the leading dimensions must be checked and for mode parameters of other than logical type.

3.4. Internal structure

Quality software is not only numerically reliable but is also designed in a surveyable and justifiable style. SLICOT routines are composed of modules with a clearly described effect. Consequences of this style are:

1. explicit typing of all arguments in the argument list and all local variables. No IMPLICIT type statement,
2. no GO TO statement to jump out of (and terminate) a DO statement,
3. a programmed WHILE statement (as long as FORTRAN has no such statement) to implement a repetition with an unforeseen number of executions,
4. IF THEN, ELSE and ELSE IF statement if possible. No COMMON, EQUIVALENCE, BLOCK DATA, assigned and computed GO TO,
5. (internal) COMMENT statements to describe the effect of a module,
6. wherever possible usage of infrastructural packages such as BLAS and NAG. Instead of LINPACK and EISPACK routines equivalent NAG routines are used.

3.5. Numerical aspects

A numerically reliable algorithm should not only be numerically stable, but should also provide full warning of results which are likely to contain large errors because, for example, the problem is ill-conditioned and should fail gracefully if it is provided with a problem for which the algorithm was not designed. A prime objective in the development of the SLICOT Library has been to make available, to the control community at large, a collection of numerically reliable software for control system analysis and design. The SLICOT routines in turn make use of a kernal of tried and tested software from the NAG Fortran Library for the underlying operations, such as, solving linear equations and eigenvalue problems and condition number estimation.

The routines also possess various other attributes that good mathematical software should have, such as, portability, which is ensured by writing routines in a standard subset of ANSI Fortran 77 and using the portability verifier from Toolpack (Pollicini, 1989); maintainability which is encouraged by documenting all user-callable routines fully and to a prescribed standard; efficiency in both computational time and storage requirements; and a well defined user interface. Finally each routine has been thoroughly tested on a wide range of examples, in a similar manner to the other NAG Libraries.

3.6. Data structures

Many of the library subprograms deal with vectors and matrices. Also polynomials and vector/matrix polynomials are used in library subprograms. Therefore special attention has been paid to a standard for data structures used in connection with these entities.

4. Library Index

Chapter A: Analysis Routines

- AA : Auxiliary Routines
- AB : State-space Analysis
- AC : Polynomial Matrix Analysis
- AD : Rational Matrix Analysis
- AE : Frequency Response Analysis
- AF : Time Response Analysis

Chapter C: Adaptive Control

CA : Auxiliary Routines
CB : Self-tuning Control
CC : Model Reference Adaptive Control
CD : Parameter Estimation

Chapter D: Data Analysis

DA : Auxiliary Routines
DB : Scaling, Interpolation
DC : Statistical Properties
DD : Trend Removal
DE : Covariances
DF : Spectra
DG : Discrete Fourier Transforms
DH : Z-Transforms
DJ : Prediction
DK : Windowing
DL : Filter Design

Chapter F: Filtering

FA : Auxiliary Routines
FB : Kalman Filters
FC : LPC Filters
FD : Fast Recursive Least Squares Filters

Chapter I: Identification

IA : Auxiliary Routines
IB : Non-parametric Methods
IC : Parametric Methods
ID : General Methods

Chapter M: Mathematical Routines

MA : Auxiliary Routines
MB : Linear Algebra
MC : Polynomial and Rational Function Manipulations
MD : Optimization
ME : Zeros and Nonlinear Equations
MF : Differential Equations

Chapter N: Nonlinear Systems

NA : Auxiliary Routines
NB : Volterra Series
NC : Bilinear Systems
ND : Describing Functions
NE : Stability Tests

Chapter S: Synthesis Routines

SA : Auxiliary Routines
SB : State-space Synthesis
SC : Polynomial Matrix Synthesis
SD : Rational Matrix Synthesis
SE : Frequency Response Synthesis
SF : Time Response Synthesis

Chapter T: Transformation Routines

TA : Auxiliary Routines
TB : State-space
TC : Polynomial Matrix
TD : Rational Matrix
TE : Frequency Response
TF : Time Response

Chapter U: Utility Routines

UA : Auxiliary Routines
UB : File Handling
UC : Graphics Handling
UD : Numerical Data Handling
UE : Machine Parameters
UF : Text Handling

5. First release of SLICOT

5.1. Contents

Following the Library Index, the contents of SLICOT's first release (see NAG, 1990) is as follows:

Chapter - A: Analysis Routines
This chapter includes 11 routines, dealing with reduction of state-space models, continuous to discrete time transformation, state space representation of (cascaded, parallel or feedback) interconnection of systems, minimal state-space representation, dual forms, invariant zeros of a given state-space form, and deadbeat control.

Chapter - D: Data analysis
This chapter includes 4 routines, dealing with topics such as convolution and deconvolution of signals, forward and inverse Fourier transform and anti-aliasing windowing.

Chapter - F: Filtering
This chapter includes 4 routines, dealing with topics such as Kalman filtering using square root covariance/information implementation.

Chapter - M: Mathematical Routines
This chapter includes 20 routines, dealing with elementary basic linear algebra operations, linear equations, least squares including total least squares, manipulations for eigenvalue and eigenvector calculation, interchanging and reordering of blocks in special matrices, decompositions and transformations such as balancing, reduction to special forms such as bidiagonal or upper Hessenberg or column echelon, diagonalisation, orthogonal transformation to bring the transformed pencil in upper block triangular form, Cholesky decomposition, computation of a basis for singular

subspace and matrix functions such as matrix exponentials.

Chapter - S: Synthesis Routines
This chapter includes 9 routines, dealing with topics such as pole placement, Riccati equations (continuous time ARE using Laub's Schur method, continuous or discrete ARE using method of deflating subspaces), Lyapunov equations for continuous and discrete time using Barraud's method or Bartels/Stewart method, continuous time Sylvester equations for general matrices and special forms, and realisation methods.

Chapter - T: Transformation Routines
This chapter includes 18 routines, dealing with several transformations from state-space form to canonical state-space forms (such as observer or controller Hessenberg) and to transfer function or to left and right polynomial matrix representation, from polynomial matrix form to transfer matrix and vice versa, computation of output sequences, F[illegible]l and Toeplitz expansion of multivariable sequences, Markov parametrisation from state-space form and transfer function.

Chapter - U: Utility Routines
This chapter includes 2 routines, dealing with printing of matrices and evaluation of machine dependent parameters.

5.2. Availability

The SLICOT Library is available as standard FORTRAN 77 code in either single or double precision and currently contains 68 user-callable routines, as mentioned in section 5.1. Each user-callable subroutine is accompanied by an example program which illustrates the use of the subroutine and can act as a template for the user's own routines. The SLICOT Library Manual provides an introduction to the Library, gives some background information to the numerical algorithms used and provides a description of each user-callable subroutine, together with a listing of the example f[illegible]ms and their output. The necessary routines from the BLAS and the NAG Library are also supplied, although users of the NAG Library are encouraged to use their implemented version.
SLICOT is a stand-alone library and should be compiled under a standard conforming compiler with no changes necessary to the source code.

6. Conclusions and future plans

Past initiatives to realising control libraries have been reviewed. Control libraries contribute nowadays highly to the flexibility and capability of existing interactive CACSD packages when using an appropriate interface between library and CACSD package such as the MATLAB Gateway Compiler.
The Benelux WGS and NAG have started a new initiative to realising a control library of basic numerical subroutines SLICOT. In this initiative several control groups at universities and industries are collaborating. SLICOT is partially based on its predecessors SLICE and SYCOT. The design of SLICOT has been introduced and the first release of SLICOT, which contains 68 routines, has been presented. SLICOT is an evolving library where new computational algorithms will be built in when they appear to outperform earlier versions (either in reliability, speed or accuracy). It is also the designer's policy to extend the Library as future routines become available. Such an ongoing effort of course relies on outside contributors willing to submit their (prototype) software to the Library and to collaborate in improving their codes. For the next releases WGS and NAG are already collaborating with contributors on topics such as identification, polynomial matrix description and H-infinity control. SLICOT offers its external contributors a means to improve and a medium to publish their software.

7. References

Boom, A. van den (1988). CADCS Developments in Europe, Invited paper, Fourth IFAC Symp. CADCS '88, Beijing, 65-73.

Cellier, F.E., P.O. Grepper, D.F. Rufer and J. Toedli (1977). Educational aspects of development and application of a subprogram package for control. Prepr. IFAC Symp. on Trends in Aut. Control Education, Barcelona, 151-159.

Denham, M.J. and C.J. Benson (1981). Implementation and documentation standards for the software library in control engineering (SLICE). Kingston Polytechnic, Control Systems Research Group, Internal Report 81/3.

Denham, M.J. (1985). A software library and interactive design environment for computer aided control systems design. In: M. Jamshidi and C. Herget (Eds) Computer Aided Control Systems Engineering, North Holland, Amsterdam.

Elmqvist, H., A. Tysso and J. Wieslander (1976). Scandinavian Control Library, Programming. Rep. Dept. of Aut. Control, Lund Inst. of Techn., Lund, S.

Grübel, G. (1983). Die Regelungstechnische Programmbibliothek RASP. Regelungstechnik, 31, 75-81.

Grübel, G. and H.D. Joos (1990). The control systems engineering numerical library RASP. Rep. TR R14-90, DLR-Institut für Dynamik der Flugsysteme.

NAG (1990). SLICOT Library Manual, Control System Analysis and Design for FORTRAN 77 Programmers, The Numerical Algorithms Group, Oxford, UK.

Pollicini, A. A. (Editor, 1989). Using Toolpack Software Tools. Kluwer Academic.

Renes, W.A. (1990). MGC - A Tool for Automatic Linking of FORTRAN Routines to MATLAB, WGS-Report 90-2.

Renes, W.A., M. Vanbegin and P. Van Dooren (1991). The MATLAB Gateway Compiler, A Tool for Automatic

Linking of FORTRAN Routines to MATLAB, IFAC CADCS '91, Swansea, UK.

Rimvall, M., D.K. Frederick, C.J. Herget and R. Kool (Eds) (1985). ELCS-Extended List of Control Software, Newsletter. Correspondence address: R. Kool, Dept. of Math. and Comp. Sc., Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, NL.

Van Dooren, P. (1989). Numerical Aspects of System and Control Algorithms, Survey article Journal A, 30, 25-32.

Varga, A. and A. Sima (1982). BIMAS-General description. Report ICI, TR-03.82, Central Inst. for Management and Informatics, Bucharest.

Varga, A. (1983). BIMASC-General description. Report ICI, TR-10.83, Central Inst. for Management and Informatics, Bucharest.

Varga, A. and A. Davidoviciu (1985). BIMASC - A package for Fortran subprograms for analysis, modelling, design and simulation of control systems. Prepr. Third IFAC/IFIP Int. Symp. CADCE '85, Copenhagen, 151-156.

WGS (1983). Implementation and Documentation Standards for the Basic Subroutine Library SYCOT. WGS Report.

WGS (1985). An Inventory of Basic Software for Computer Aided Control System Design, WGS-Report 85-1.

WGS (1990). SLICOT, Implementation and Documentation Standards, WGS-Report 90-1; also NAG report NP2032.

Wieslander, J. (1977). Scandinavian Control Library, A Subroutine Library in the field of Automatic Control. Rep. Dept. of Aut. Control, Lund Inst. of Techn., Lund, S.

Williams, T.W.C. (1986). Numerically reliable software for control: the SLICE library. IEE Proc., vol 133, Part D, 73-82.