3rd IFAC Symp. Advances in Control Education ACE'94, August 1-2, 1994, Tokyo

DEVELOPMENTS IN THE FIELD OF CONTROL LIBRARIES AND THEIR IMPACT ON CONTROL EDUCATION

AD VAN DEN BOOM[1], ALOYS GEURTS[2], GEORG GRüBEL[3] and ANDRAS VARGA[3]

[1]*Dept. of Electrical Engineering, Eindhoven University of Technology, NL-5600 MB, Eindhoven, The Netherlands*
[2]*Dept. of Mathematics and Computing Science, Eindhoven University of Technology, NL-5600 MB Eindhoven, The Netherlands*
[3]*DLR-Oberpfaffenhofen, Control Design Engineering, Institute for Robotics and System Dynamics, D-82230 Wessling, Germany*

Abstract. The state of the art of control libraries is presented. Attention is paid to the joining of efforts to developing a future mature control library based on the two existing libraries in active development RASP and SLICOT. The educational aspects of the use of control libraries are emphasized.

Key Words. Control libraries, computer aided system design, numerical control software, education, educational aids

1. INTRODUCTION

In the past decades much effort has been invested in the construction of Computer Aided Control System Design (CACSD) platforms. The initial designs were straightforward, with relatively little attention paid to aspects such as numerical reliability, flexibility, portability and user-friendliness. With the advent of increasingly powerful hardware and the developments in graphics, the CACSD platforms evolved rapidly into sophisticated, complex software structures. Nowadays some of these, such as MATLAB and MATRIXx (and their successors), have become generally accepted. Apart from these general platforms, several designs for specific educational CACSD platforms (often dealing with elementary control topics) have been proposed in the past, (Durfee, 1991).

When scanning the literature on CACSD platforms, one has to draw the conclusion that major emphasis is given to the architectural and performance issues of the platform. Also the papers on platforms for specific educational purposes mainly deal with architecture and performance. A relatively small part of the CACSD literature concerns the design and development of computational routines for control. In most of the platforms basic routines are used, which are locally developed and often designed on an ad hoc basis.

New developments in CACSD are to be expected along the principles of so-called "open systems", see Barker *et al.* (1992). An open system reflects a generally accepted and published system standard to which any vendor can build products. Examples of highly successful open systems are the IBM PC compatible bus, the UNIX operating system, the X-Window System, several published standards such as GKS, Fortran etc., or proprietary standards such as PostScript, Ethernet etc. The fact that all involved parties conformed to these open systems enabled major technical breakthroughs in the past, which in themselves were highly beneficial for all parties, in particular the user. As many vendors operate on the same market, there will be fierce competition and consequently a large variety of products offered. Then the user can choose those products that are best suited for his needs and that have the best performance and price. This is in contrast to the proprietary closed systems, which are fully controlled by the supplier, so that the user is completely dependent on him. The formulation and acceptance of an open system standard for CACSD would stimulate the further development of CACSD. Barker *et al.* (1992) present an open architecture for Computer Assisted Control Engineering (CACE) together with the necessary components. Their design contains tool slots for processing engines such as simulation tools, numerical analysis tools etc.

2. CONTROL LIBRARIES

In the architecture of several CACSD platforms one can explicitly identify a so-called control library representing the collection of all computational routines to solve specific mathematical and control problems (van den Boom *et al.*, 1992). Such a library can be developed independently of the development of the platform itself. This is, in general, advantageous as the library is not influenced by the specific requirements of the platform operation, by the languages used for its implementation, or by the data structures employed. Moreover, an independent control library can serve the development of more than one platform or can be used to complement the local library of an already existing platform. A true independence of the control library is generally only possible by using general purpose languages such as Fortran or C. Existing platforms such as MATLAB can benefit from these independent libraries via a gateway compiler, see Renes *et al.*, (1991).

CACSD platforms have had a relatively short life cycle in the past. However, a control library has a much longer life cycle, providing the necessary opportunity to create a mature library. For the development of a control library, a different expertise is required than that for the development of the architecture of the platform itself. It can be characterized by the joining of efforts in the domains of systems and control theory and engineering, numerical mathematics and uumerical software engineering.

Portable numerical software, written in Fortran for CACSD, is provided by several control libraries such as RASP (Grübel and Joos, 1990), SLICOT (van den Boom *et al.*, 1991), BIMAS (Varga and Sima, 1985) and BIMASC (Varga and Davidoviciu, 1986). These libraries share many common features, for instance rigorous implementation along well established programming and documentation standards, numerical robustness of software implementations, similarities in the covered topics, and use of high performance algorithms. On the other hand, they differ to some extent with respect to their main goals, organization, user interface, and size. Of these libraries, RASP and SLICOT are apparently, to date, the only ones in active further development.

The RASP routines cover a broad area of control engineering computations, supporting frequency- and time-domain analysis and synthesis techniques, multi-criteria parameter optimization, simulation, and graphics. RASP, together with the engineering-database and -operating system RSYST (Grübel and Joos, 1991), form the software infrastructure of the CACE environment ANDECS (Grübel *et al.*, 1993). The organization of the library and the standardization of user interface of routines reflect the orientation of RASP towards control engineering applications. Special emphasis in RASP is placed on the numerical reliability of the implemented algorithms. The development policy of RASP is oriented towards independence of commercial basic software. The numerical software supporting RASP includes the linear algebra libraries BLAS (Level 1), LINPACK, EISPACK, and recently LAPACK (Anderson *et al.*, 1992) as well as some libraries providing facilities for simulation and optimization of nonlinear systems (ODEPACK, MINPACK). The graphics in RASP is now implemented to use X-Windows calls directly. The current version RASP'92 consists of about 320 user callable routines (not counting those from the above-mentioned standard libraries). A new version RASP'93 is in a final implementation stage. RASP is used in education and research at many universities and research sites in Germany.

SLICOT can be primarily viewed as a mathematical library for control theoretical computations. The library provides tools to perform basic system analysis and synthesis tasks. The main emphasis in SLICOT is on numerical reliability of implemented algorithms and the numerical robustness of routines. A special emphasis is on providing maximum algorithmic flexibility to users, and on the use of rigorous implementation and documentation standards, which have been published by WGS (1990), and can thus be regarded as open in the sense mentioned in the previous section. SLICOT is a product of WGS, realized in cooperation with NAG. The supporting numerical libraries for SLICOT are BLAS (Level 1,2,3) and similar additional NAG routines, as well as some LINPACK and EISPACK equivalent routines from the NAG library. The latest Release 2 of SLICOT contains about 90 user callable routines and Release 3 is under development. SLICOT is distributed worldwide by NAG.

The effort to develop both the SLICOT and the RASP libraries was very intensive and substantial. Taking into account that both libraries will continue to evolve, there are serious concerns about rationalizing future development efforts. In particular, duplicating software pieces for which robust implementations are already available should be avoided. This has led to the idea of a coordinated future development of the RASP and the SLICOT libraries in order to optimize the implementation efforts. Therefore recently the RASP/SLICOT mutual compatibility concept has been introduced (Grübel *et al.*, 1994), which permits a coordinated future development. A basis for this concept is the exclusive use of BLAS and LAPACK libraries as supporting linear algebra libraries for future RASP and SLICOT implementations. Now LAPACK is a de-facto standard for linear algebra computations and will replace LINPACK and EISPACK completely in the future. First result of this RASP/SLICOT collaboration are the model reduction library RASP-MODRED (Varga, 1994) and RASP-DESCRIPT (Varga, 1992) for the analysis and modelling of descriptor systems.

The planned future coordination of the two libraries RASP and SLICOT has important consequences. It will be obvious that the combination of the two libraries cover many and large chapters of the discipline of systems and control theory and engineering. Therefore the combined library is potentially capable of approaching a "mature" status, with respect to size, completeness and numerical quality. A mature library represents the state of the art in the discipline, i.e. it contains (almost) all known best methods and their best implementation with known performances. It is an interesting platform for researchers to publish good implementations of powerful new methods. Due to its open character, such a library is suited to serve foreseen developments such as open CACSD systems.

3. EDUCATIONAL ASPECTS

Developers of educational interactive CACSD software have frequently relied on self-made control routines and basic numerical software, which were conceived for a limited set of control assignments of a specific course. Although these routines were indeed usable for their intended local use in courses, often they could not be used to solve almost similar (still relatively simple) control problems in other educational environments. Reasons were the low flexibility of the design and the lack of portability of the software, the often poor numerical quality of the algorithms, and the limited set of methods covered.

The developments in control libraries, as sketched above, permit the use of reliable, high performance and well-tested basic control routines in a variety of environments and has several important educational implications.

1. Algorithmic and functional flexibility. The developers of the control libraries RASP and SLICOT have concentrated their efforts on maximally attainable algorithmic flexibility and efficiency of the library routines. Examples of flexible and efficient implementations of good algorithms in SLICOT are various routines for transformation between different system representations, routines for solving matrix equations (Lyapunov, Sylvester, Riccati), or routines for system analysis. Examples of good implementations of up-to-date algorithms in RASP are the already mentioned collections of routines for analysis and modelling of descriptor systems and for model reduction. The development of these libraries was only possible because of the joint effort of experts from different fields, as mentioned before. A flexible library routine allows the use of a wide spectrum of educatioual examples, assignments or test cases, ranging from basic to postgraduate level, and is therefore adequate for use in all kinds of courses. These examples go far beyond the common set of simple, straightforward and low order examples which are presented too often. Well implemented and stable algorithms can be applied for a large class of even "hard cases", reflecting the maximal capabilities of the method in combination with the best possible (or best known) implementation.

2. Reference software. The subroutines of a control library can serve as a standard reference in textbooks, embodying the best possible or best known implementation of a method and demonstrating the performance and limitations of the methods. This is a direct consequence of the previous remark. The presentation of powerful control methods together with - and based on - their best possible or best known implementation is instructive and, from an educational point of view, highly desirable.

3. Benchmark problems. The previous statements lead to the recognition of the need for more and adequate benchmarks for control methods and their implementation, see Frederick (1988). Carefully chosen benchmarks give insight into the performance of methods in the language of the control system analyst or control engineer. The need for this kind of insight is rapidly increasing due to today's widespread availability of a wealth of methods and implementations. Therefore WGS has recently started an initiative to develop an adequate set of benchmarks for control methods; see Mehrmann (1994).

From this line of reasoning, it can be concluded that the combination and interrelation of the three entities *method*, *implementation* and *benchmarks* deserves more emphasis in the literature, both from a research as well as an educational point of view. In particular a coherent presentation of these three entities is a major educational objective.

4. Collection of standard (classroom) examples. Based on the present day's availability of control methods and their implementation, an *"open" standard collection* of carefully chosen appealing worked examples (preferably based on real data) and case studies, which are typically for the respective methods, can be composed. This standard collection can serve worldwide as a basis for several courses at various levels dealing with analysis and design of dynamical systems and for courses which are devoted to control applications in practice. Apart from university students, such a set of examples is of particular interest for industrial employees at R&D positions as these examples show directly the usefulness and the capabilities of methods and their implementations. It could be an initiative of conferences like this to stimulate and coordinate these activities.

5. Professional attitude towards professional tools. A mature and complete control library provides the student with a rich, up-to-date and complete set of good implementations of good methods with known performances and limitations, which are applicable in various kinds of (future) CACSD shells, so the student becomes aequainted with the state of the art of available system analysis and design tools. This cultivates and encourages a professional attitude towards the use of professional tools (for instance the student learns that programming control algorithms is more than producing a straightforward calculation scheme).

4. CONCLUSION

An overview of the state of the art of control libraries has been given in relation to developments in Computer Aided Control Systems Design. It is reported that coordination of efforts with respect to future realization of control libraries has been planned. This enables the development of a mature library. Much attention has been given to the educational implications of control libraries. These can be characterized as a further step towards the professional use of professional up-to-date tools in various levels of educational environments, enabling access to the best control methods and their best implementations. It is argued that there is a need for benchmark problems and standard (classroom) examples.

5. REFERENCES

Anderson, E., Z. Bai, J. Bishop, J. Demmel, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. (1992). *LAPACK User's Guide*, SIAM, Philadelphia, USA.

Barker, H.A., M. Chen, P.W. Grant, C.P. Jobling and P. Townsend. (1992). An open architecture for computer-assisted control engineering. 1992 IEEE Symp. on CACSD, Napa, USA.

Boom, A. van den, A. Brown, A. Geurts, S. Ham-marling, R. Kool, M. Vanbegin, P. Van Dooren, and S. Van Huffel. (1991). SLICOT, a subroutine library in control and systems theory. Prepr. 5th IFAC/IMACS Symp. CADCS'91, Swansea, UK, pp. 89-94.

Boom, A. van den, S. Hammarling, W. Renes, P. Van Dooren and A. Varga. (1992). Integrating CACSD packages and control libraries; towards more flexible and versatile CACSD. Proc. IEEE Symp. on CACSD, Napa, USA, pp. 62-67.

Durfee, W.K. (1991) VISDYCON: VISual DYnamics and Control. Prepr. IFAC Symp. ACE'91, Boston, USA, pp. 128-131.

Frederick, D.K. (1988). Benchmark problems for computer aided control system design. Prepr. 4th IFAC Symp. on CADCS, Beijing, pp. 1-6.

Grübel, G., and H.-D. Joos. (1990). *The control systems engineering numerical subroutine library RASP*. Technical Report TR R14-90, DLR - German Aerospace Research Establishment, D-82230 Wessling.

Grübel, G., and H.-D. Joos. (1991). RASP and RSYST - two complementary program libraries for concurrent control engineering. Prepr. 5th IFAC/IMACS Symp. CADCS'91, Swansea, UK, pp. 101-106.

Grübel, G., H.-D. Joos, M. Otter, and R. Finsterwalder. (1993). The ANDECS design environment for control engineering. Prepr. 12th IFAC World Congress, Sydney, Australia.

Grübel, G., A. Varga, A. van den Boom and A.J. Geurts. (1994). Towards a coordinated development of numerical CACSD software: the RASP/SLICOT compatibility concept. Prepr. IEEE/IFAC Symp. CACSD'94, Tucson, USA.

Mehrmann, V. (1994). Second call for benchmark examples. E-LETTER on Systems, Control, and Signal Processing, Issue 66, Febr. 1994.

NAG. (1990). *The NAG Fortran Library Manual.* Technical report, NAG Ltd., Oxford, UK.

Renes, W., M. Vanbegin, P. Van Dooren and J. W. Beckers. (1991). The MATLAB Gateway Compiler, a tool for automatic linking of Fortran Routines to MATLAB. Prepr. 5th IFAC/IMACS Symp. CADCS'91, Swansea, UK, pp. 95-100.

Varga, A., and V. Sima. (1985). BIMAS - A basic mathematical package for computer aided systems analysis and design. Proc. 9th IFAC World Congress, Budapest, Hungary.

Varga, A., and A. Davidoviciu. (1986). BIMASC A package of Fortran subprograms for analysis, modelling, design and simulation of control systems. Proc. 3rd IFAC/IFIP Symp. CADCE'85, Copenhagen, Denmark.

Varga, A. (1992). Numerical algorithms and software tools for analysis and modelling of descriptor systems. Prepr. 2nd. IFAC Workshop on System Structure and Control, Prague, Czechoslowakia, pp. 392-395.

Varga, A. (1994). Numerical methods and software tools for model reduction. Proc. 1st. MATHMOD Conf., Vienna, Austria, vol 2., pp. 226-230.

WGS. (1990). *Implementation and Documentation Standards*. Report 90-1, Working Group on Software. Editor R. Kool, Eindhoven University of Technology, the Netherlands.