Workshop on numerical methods in automatic control, Sept 23-25, 1980

Hagander, Per

1980

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Workshop on Numerical Methods in Automatic Control

Lund Institute of Technology
Sweden

Sept 23-25, 1980

Sponsored by The Swedish Institute for Applied Mathematics (ITM)

Tuesday Sept 23   9.00–11.45
Session I: Surveys
The Annex   Room   MA:3

L E C T U R E S

9–9.30	Registration etc
9.30–10.00	KJÅ GD CEF Introduction
10.00–10.15	Coffee
10.15–11.05	A Laub, University of Southern California, USA Survey of Computational Methods in Control Theory
11.05–11.25	V Klema, MIT, USA Mathematical Software for Control and Estimation Theory (tentative title)
11.25–11.45	M Denham, Kingston Polytechnic, UK UK SRC Control algorithm library
11.45–13.30	Lunch Helsingkrona nation

Tuesday Sept 23,   13.30–17.00
Session II: Linear Algebra Methods
The Annex   Room   MA:3

L E C T U R E S

13.30–14.00	D Boley and G H Golub, Stanford University, USA Block Lanczos and Control Theory
14.00–14.40	M Denham, Kingston Polytechnic, UK The Determination of Large Scale Systems Structures using Principal Component Analysis
14.40–15.00	Coffee
15.00–15.40	P Van Dooren, Philips research laboratory, Brussels, Belgium A Generalized Eigenvalue approach for solving Riccati Equations
15.40–16.20	V Strejc, Czechoslovak Academy of Science, Prag Square Root Filtering and UD-decomposition
16.20–17.00	C B Moler, Univ of New Mexico, USA MATLAB -- An interactiv matrix laboratory
18.30 18.45	Bus leaves hotel Lundia bus leaves M-building, LTH for dinner at the castle Svaneholm

Wednesday Sept 24, 14-17
Session IV: Optimization and Identification
The Annex Room MA:3

L E C T U R E S

14.00-14.20	R Gorez, University of Louvain, Belgium Some features of Chandrasekhar equations and Orthogonal Transformation Techniques in Optimal Control
14.20-14.40	A Wierzbicki, IIASA, Austria Trajectory Optimization- Theoretical Background and Practical Motivations for Infinite-Dimensional Objectives in Dynamical Optimization and Control
14.40-15.00	B Asselmeyer, Technische Hochschule Darmstadt, West Germany Comparison of several optimization methods not requiring exact one-dimensional searches - Numerical experience.
15.00-15.10	Coffee
15.10-16.00	E Polak, University of California Berkley, USA Algorithms for Control Systems Design Involving Singular Value Inequalities
16.00-16.20	E Spedicato, University of Bergamo, Italy Use of Slack Variables in Nonlinear Programming
16.20-16.40	P Zencke, University of Bonn, West Germany A self starting algorithm for the identification of positive exponential decay components by minimization of the Chebyshev error
16.40-17.00	A Ruhe, University of Umeå, Sweden Fitting Empirical Data by Positive Sums of Exponentials
18.00-	Possibility to use the program packages developed at the Department of Automatic Control, Lund

Wednesday Sept 24, 9.00-12.15
Session III: Differential equations and Optimal Control
M-building Room M:B

L E C T U R E S

9.00-9.40	F Cellier, ETH, Switzerland On the Numerical Integration of Discontinuous Ordinary Differential Equations - Simulations and Real-time Applications
9.40-10.20	C W Gear, University of Illinois, USA Oscillating Problems and Limit Cycles (tentative title)
10.20-10.35	Coffee
10.35-11.15	M Sorine, INRIA, France Numerical Solution of Riccati and Chandrasekhar Equations arising in Optimal Control of Distributed Parameter Systems
11.15-11.45	N Nichols Application of Numerical Boundary Value Problems in Optimal Control
11.45-12.15	S A Gustafsson, The Royal Institute of Technology, Stockholm, Sweden Numerical Treatment of Parabolic Boundary-Value Control Problem by means of semi-infinite Programming
12.15-14.00	Lunch Finn Inn

Thursday Sept 25, 9-12
Session V: Polynomial and frequency reponse methods
The Annex, Room MA:3

L E C T U R E S

9.00-9.40	V Kucera, Czechoslovak Academy of Science, Prag Control Systems Design using polynomial Equations. Numerical Aspects
9.40-10.00	L Pernebo, Lund Institute of Technology, Sweden Numerical Problems in Multivariable Synthesis using the Polynomial Approach
10.00-10.15	Coffee
10.15-10.55	J Edmunds, Cambridge University, UK A frequency responce algorithm to calculate precompensators which optimise the closed loop response of a multivariable system
10.55-11.15	K Zeiske, Ruhr-Universität Bochum, West Germany Routh-criterion for obtaining the roots of polynomials
11.15-12.00	Informal presentations and Discussion
12.00-13.30	Lunch Finn Inn

Thursday Sept 25, 13.30-17.00
Session VI: Software Packages
The Annex Room MA:3

L E C T U R E S

13.30-14.00	J Wieslander, Stora Kopparberg AB, Falun, Sweden A Sequence of Programs for Control Systems Analysis and Design
14.00-14.30	A Barraud, Laboratoire D'Automatique de Grenoble, France Autopac a Robust Software for Automatic Control Purposes
14.30-15.00	H Elmqvist, Lund Institute of Technology, Sweden A Language for Dynamical Models Based on General Equations
15.00-15.15	Coffee
15.15-15.35	C Suleyman and A Barraud, Laboratoire D'Automatique de Grenoble, France ISER-CSD an Interactive System for Education and Research in Control
15.35-15.55	R Gorez, University of Louvain, Belgium PAAS and SIMUL, Conversational Programs for Educational Purposes in Control Systems Analysis
15.55-16.15	H C Lammers and A Kraak, Delft University of Technology, The Netherlands STUPID - an in-line computer-aided control/design package
16.15-17.00	Concluding Remarks by KJA, GD, C-E F

WORKSHOP ON NUMERICAL METHODS IN AUTOMATIC CONTROL
Sept 23-25 1980

List of Participants

ASSELMEYER, Bernhard	Technische Hochschule Darmstadt Darmstadt, Germany
BARRAUD, A.Y.	Laboratoire d'Automatique de Grenoble Saint-Martin d'Heres, France
BJÖRCK, Åke	Linköping University Linköping, Sweden
BOLEY, D	Stanford University Stanford, USA
BRAUN, Rolf	Lund Institute of Technology Lund, Sweden
CELLIER, F	ETH-Zentrum Zürich, Switzerland
DAHLQUIST, Germund	The Royal Institute of Technology Stockholm, Sweden
DENHAM, M	Kingston Polytechnic Kingston-upon-Thames, UK
van DOREN, P	Tienen, Belgium
EDBERG, Eva	FOA Linköping, Sweden
EDMUNDS, J	Univ. Eng, Dept. Cambridge, UK
EINARSSON, Bo	FOA Umeå, Sweden
EKBLOM, Håkan	Luleå University Luleå, Sweden
ELFVING, Tommy	FOA Linköping, Sweden
ELMQVIST, Hilding	Lund Institute of Technology Lund, Sweden
ERIKSSON, Göran	Lund Institute of Technology Lund, Sweden
FROCK, Beatrice	The Royal Institute of Technology Stockholm, Sweden
FRÖBERG, Carl-Erik	Lund University Lund, Sweden
GEAR, C.W.	University of Illinois Urbana, USA
GLAD, Torkel	Linköping University Linköping, Sweden

GOLUB, G.H.	Stanford University Stanford, USA
GOREZ, Raymond	Automatique Louvain-la-Neuve, Belgium
GUSTAFSSON, S-Å	The Royal Institute of Technology Stockholm, Sweden
GUTMAN, Per-Olof	Lund Institute of Technology Lund, Sweden
HAGANDER, Per	Lund Institute of Technology Lund, Sweden
HAMMARLING, S.J.	National Physical Laboratory Teddington, UK
HEROLF, Magnus	FOA Stockholm, Sweden
HOLMSTRÖM, Kenneth	Umeå University Umeå, Sweden
HOLST, Jan	IMSOR, DTH Lyngby, Denmark
HOUBAK, Niels	DTH Lyngby, Denmark
HÄGGLUND, Tore	Lund Institute of Technology Lund, Sweden
JACQUELIN, Jean-Claude	Norsk Data Frankrike France
JOHANSSON, Rolf	Lund Institute of Technology Lund, Sweden
JONSSON, Henrik	Linköping University Linköping, Sweden
JØRGENSEN, Sten B	DTH Lyngby, Denmark
KARNY, M	Czechoslovak Academy of Sciences Prague, Czechoslovakia
KLEMA, Virginia	Massachusetts Institute of Technolocy Cambridge, USA
KOMAN, Pierre	Twente University of Technology Enschede, The Netherlands
KUCERA, V	Czechoslovak Academy of Sciences Prague, Czechoslovakia
KÅGSTRÖM, Bo	Umeå University Umeå, Sweden
LAMMERS, H.C.	Electrical Engineering Delft, The Netherlands
LANGE, Ove	Norsk Data Frankrike France

LAUB, A	University of Southern California Los Angeles, USA
LENELLS, Matz	Lund Institute of Technology Lund, Sweden
LENNARTSSON, Bengt	Gothenburg, Sweden
LINDAHL, Sture	Lund Institute of Technology Lund, Sweden
LJUNG, Stefan	Linköping University Linköping, Sweden
MANNERFELT, Carl Fredrik	Lund Institute of Technology Lund, Sweden
MATTSSON, Sven Erik	Lund Institute of Technology Lund, Sweden
MOLER, C.B.	University of New Mexico Albaquerque, USA
NICHOLS, N	University of Reading Reading, UK
NILESEN, Lars	Lund Institute of Technology Lund, Sweden
NYBRANT, Thomas	Institute of Technology Uppsala, Sweden
OLBJER, Lennart	Lund Institute of Technology Lund, Sweden
PERNEBO, Lars	Lund Institute of Technology Lund, Sweden
POLAK, L	University of California at Berkley Berkley, USA
REICHEL, Lothar	The Royal Institute of Technology Stockholm, Sweden
RUHE, Axel	Umeå University Umeå, Sweden
RÜSTER, Ennu	Tallinn Technical University Tallinn, USSR
SORINNE, M	INRIA Le Chesnay, France
SPEDICATO, E	University of Bergamo Bergamo, Italy
STREJC, V	Czechoslovak Academy of Sciences Prague , Czechoslovakia
STERNBY, Jan	Kockumation AB Malmö, Sweden
SULEYMAN, C	Laboratoire d´Automatique Saint-Martin-d´Heres, France
SÖDERLIND, Gustaf	The Royal Institute of Technology Stockholm, Sweden

SÖDERSTRÖM, Torsten	Institute of Technology Uppsala, Sweden
THUNHEM, Aage J	SINTEF-NTH Trondheim, Norway
TRIGIANTE, D	IBM Scientific Center Roma, Italy
TULLSSON, Bert-Eric	FOA Stockholm, Sweden
WEDIN, Per-Åke	Umeå University Umeå, Sweden
WIERZBICKI, Andrzej P	International Institute for Applied Systems Analysis Laxenberg, Austria
WIESLANDER, Johan	Stora Kopparberg AB Falun, Sweden
WITTENMARK, Björn	Lund Institute of Technology Lund, Sweden
WITTMEYER-KOCH, Linde	Linköping University Linköping, Sweden
ZEISKE, Karl	Ruhr-Universität Bochum, Germany
ZENCKE, P	Institut für Angewandte Matematik der Universität Bonn Bonn, Germany
ÅSTRÖM, Karl Johan	Lund Institute of Technology Lund, Sweden

STUPID - an in-line computer-aided control/design package

H.C. Lammers
A. Kraak

Laboratory for Control Engineering
Dept. of Electrical Engineering
Delft University of Technology
Delft, The Netherlands.

Many recent publications show the possible implementations of various digital control algorithms in small microcomputers.
Realisations ranging from simple P.I.D.-controllers up to more advanced self-tuning and adaptive control algorithms (1) are known.
The process of designing digital controllers can be separated into two steps: the development and experimental stage and the final implementation stage.
Consequently, in designing a single controller as such, one is faced with a few demands

- the microcomputer implemented controller needs to be cheap, robust, simple, safe and to have only those intervention and supervention facilities considered necessary. So, to reach an optimal result in the last stage one has to find out which control algorithm and previous tuning will combine the qualities mentioned with a desired control loop behaviour.

In most practical industrial situations, however, the final resulting controller will form a part of large-scale control-systems, either concentrated in a computer main frame or in a control system with some kind of hierarchical structure (2,3). As available control equipment offered by industry is based on long experience with mostly P.I.D.-like algorithms, it will become necessary to study the behaviour of tuning and adaptive controllers in similar environments. To benefit from the use of self-tuning controllers in large scale systems a few more problems, related to the use of this type of controller among others, have to be considered:

- How to confront control-system operators with the different types of self-tuning regulators together with the familiar P.I.D.-controllers. Some ideas on this topic can be found in (4).
- How to guarantee safe operation and sufficient alarm facilities.
- How to make use of a "scanning tuner": several controllers tuned by one tuning device.

The computer program "STUPID" (Self-Tuners User's Package for Interactive Design), that will be presented, is a block-oriented control/simulation software package specially developed for these design purposes (5). With this program package it is possible to compose simulation/control structures to test and compare various control algorithms, whether or not combined with optimalisation and/or estimation facilities in off- and in-line environments. Figure 1 shows a possible control structure.
Blocks to simulate proces behaviour are concentrated within the curved line.
In-line control is made possible by replacing blocks by AD- or DA-convertors.
Supervision can be immediately performed using extended graphic display facilities showing current signals, parameters and/or data characteristics to the performance of the control loop.

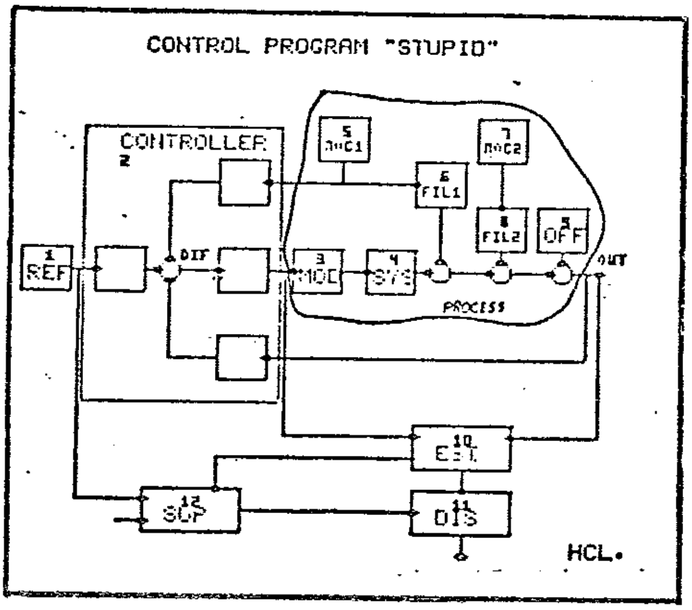


Fig.1 General control/simulation structure

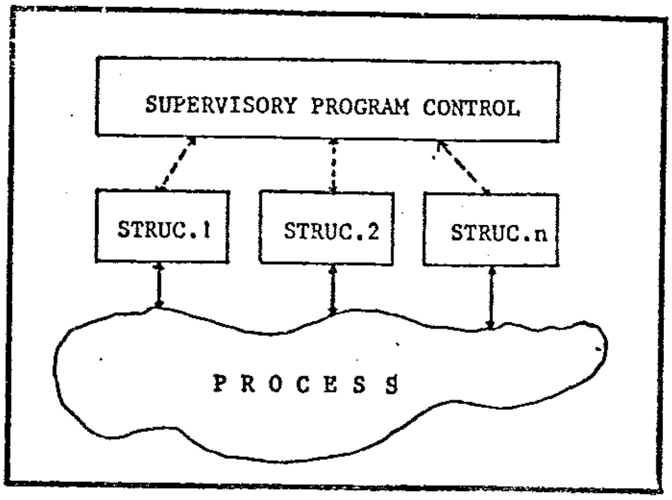


Fig. 2 Possible hierarchical structure

Changes in structure data can be made in real time or whole blocks can be replaced. Run and structure data can be stored in-line, to reconstruct and study the experiments afterwards.
To study controllers in hierarchical or decentralised networks it is possible to run several structures of figure 1 at the same time all governed by one supervisory program (fig.2).
In this case the different structures could run in separate microcomputers. Besides the design and testing of optimal control algorithms the program can, because of its blockwise programmable structure, be used by researchers to try out new algorithms in off- and in-line environments, or by students to get acquainted with the properties and behaviour of different modern control techniques. Special precautions have been taken into account to meet this educational demand.

(1) Clarke, D.W. — "Implementation and application of microprocessor-based Self-Tuners". Proc. of the Fifth IFAC Symp. on Identification and system parameter estimation, 1979.

(2) Williams, T.J. — "A hierarchical control for large scale systems- A survey", Proc. of the 7th IFAC World Congress, 1978.

(3) Eelderink, G.H.B., Bruijn, P.M., Verbruggen, H.B. — "A block-diagram language to implement controllers in a distributed computer control network", The 2nd IFAC/IFIP Symp. on Software for Computer Control June 1979.

(4) Wittenmark, B., Åström, K.J. — Simple Self-Tuning Controllers International Symp. on adaptive systems, Bochum, March 1980.

(5) Lammers, H.C. — STUPID-manual, Laboratory for Control Engineering, Delft University of Technology, March 1980.

Survey of Computational Methods
in Control Theory

Alan J. Laub*

[illegible]

Abstract. This paper focuses on the interplay between problems arising in linear control and systems theory and their solution by recent results and methodologies in numerical linear algebra. The following are among the topics addressed: controllability, special matrix equations such as Lyapunov, Sylvester, and Riccati equations, numerical calculation of various objects arising in the geometric theory of multivariable control, and efficient frequency response calculations. Applications of some of the topics are mentioned and embodiment of the theory and algorithms as reliable mathematical software is also discussed. The format is kept deliberately expository and many open problems are featured throughout.

*Department of Electrical Engineering - Systems, University of Southern California, Los Angeles, California 90007.
This research was supported by the U.S. Army Research Office and the Office of Naval Research under Contract No. DAAG29-79-C-0031.

REFERENCES

[1] W. M. WONHAM, Linear multivariable control: a geometric approach, Second Edition, Springer-Verlag, New York, 1979.

[2] R. W. BROCKETT, Finite dimensional linear systems, Wiley, New York, 1970.

[3] H. KWAKERNAAK and R. SIVAN, Linear optimal control systems, Wiley, New York, 1972.

[4] J. H. WILKINSON, Rounding errors in algebraic processes, Prentice-Hall, Englewood Cliffs, NJ, 1963.

[5] J. H. WILKINSON, The algebraic eigenvalue problem, Oxford University Press, London, 1965.

[6] B. T. SMITH, et al., Matrix eigensystem routines -- EISPACK guide, Second Edition, Lect. Notes in Comp. Sci., Vol. 6, Springer-Verlag, New York, 1976.

[7] B. S. GARBOW, et al., Matrix eigensystem routines -- EISPACK guide extension, Lect. Notes in Comp. Sci., Vol. 51, Springer-Verlag, New York, 1977.

[8] G. H. GOLUB and J. H. WILKINSON, Ill-conditioned eigensystems and the computation of the Jordan canonical form, SIAM Review, 18(1976), pp. 578-619.

[9] W. GIVENS, Numerical computation of the characteristic values of a real symmetric matrix, Oak Ridge National Lab., ORNL-1574, March, 1954.

[10] G. STRANG, Linear algebra and its applications, Second Edition, Academic Press, New York, 1980.

[11] G. W. STEWART, Introduction to matrix computations, Academic Press, New York, 1973.

[12] G. E. FORSYTHE, M. A. MALCOLM, and C. B. MOLER, Computer methods for mathematical computations, Prentice-Hall, Englewood Cliffs, NJ, 1977.

[13] G. DAHLQUIST and Å. BJÖRCK, Numerical methods, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[14] G. H. GOLUB and W. KAHAN, Calculating the singular values and pseudo-inverse of a matrix, SIAM J. Numer. Anal., 2(1965), pp. 205-224.

[15] G. H. GOLUB and C. REINSCH, Singular value decomposition and least squares solutions, Numer. Math., 14(1970), pp. 403-420.

[16] J. DONGARRA, et al., LINPACK user's guide, SIAM, Philadelphia, 1979.

[17] V. C. KLEMA and A. J. LAUB, The singular value decomposition: its computation and some applications, IEEE Trans. Aut. Contr., Vol. AC-25, April, 1980.

[18] G. H. GOLUB, V. C. KLEMA, and G. W. STEWART, Rank degeneracy and least squares problems, Tech. Rep't. STAN-CS-76-559, Computer Science Dept., Stanford University, Aug. 1976.

[19] A. M. OSTROWSKI, On the spectrum of a one-parametric family of matrices, Journal für die reine und angewandte Mathematik, Band 193, Heft 3/4(1954), pp. 143-160.

[20] A. K. CLINE, C. B. MOLER, G. W. STEWART, and J. H. WILKINSON, An estimate for the condition number of a matrix, SIAM J. Numer. Anal., 16(1979), pp. 368-375.

[21] C. B. MOLER and C. F. VAN LOAN, Nineteen dubious ways to compute the exponential of a matrix, SIAM Rev., 20 (1978), pp. 801-836.

[22] W. ENRIGHT, On the efficient and reliable numerical solution of large linear systems of ODE's, IEEE Trans. Aut. Contr., AC-24(1979), pp. 905-908.

[23] B. C. MOORE and A. J. LAUB, Computation of supremal (A, B)-invariant and controllability subspaces, IEEE Trans. Aut. Contr., AC-23(1978), pp. 783-792.

[24] P. VAN DOOREN, The generalized eigenstructure problem. Applications in linear system theory, Ph.D. Thesis, Katholieke Universiteit Leuven, Belgium, May, 1979.

[25] F. R. GANTMACHER, The theory of matrices, Vols. I, II, Chelsea, New York, 1959.

[26] A. J. LAUB and B. C. MOORE, Calculation of transmission zeros using QZ techniques, Automatica, 14(1978), pp. 557-566.

[27] C. B. MOLER and G. W. STEWART, An algorithm for generalized matrix eigenvalue problems, SIAM J. Numer. Anal., 10(1973), pp. 241-256.

[28] J. C. WILLEMS, Almost invariant subspaces: an approach to high gain feedback design, submitted to IEEE Trans. Aut. Contr.

[29] R. H. BARTELS and G. W. STEWART, Solution of the matrix equation AX + XB = C, Comm. ACM, 15(1972), pp. 820-826.

[30] G. H. GOLUB, S. NASH, and C. F. VAN LOAN, A Hessenberg-Schur method for the problem AX + XB = C, IEEE Trans. Aut. Contr., AC-24(1979), pp. 909-913.

[31] W. GIVENS, Elementary divisors and some properties of the Lyapunov mapping $X \to AX + XA^*$, Argonne Na. Lab. Rep't. ANL-6456, Nov., 1961.

[32] A. J. LAUB, A Schur method for solving algebraic Riccati equations, IEEE Trans. Aut. Contr., AC-24 (1979), pp. 913-921.

[33] T. PAPPAS, A. J. LAUB, and N. R. SANDELL, On the numerical solution of the discrete time algebraic Riccati equation, IEEE Trans. Aut. Contr., Vol. AC-25, June, 1980.

[34] C. PAIGE, Properties of numerical algorithms related to computing controllability, to appear in IEEE Trans. Aut. Contr.

[35] B. C. MOORE, Principal component analysis in linear systems: controllability, observability, and model reduction, to appear in IEEE Trans. Aut. Contr.

[36] A. J. LAUB, On computing "balancing" transformations, Dec. 1979, submitted for 1980 JACC, San Francisco, August, 1980.

[37] A. J. LAUB, Robust stability of linear systems - some computational considerations, in C. Wilde, et al. (Eds.), Information Linkages Between Applied Mathematics and Industry, Academic Press, New York, 1980.

[38] M. G. SAFONOV, A. J. LAUB, and G. L. HARTMAN, Feedback properties of multivariable systems: the role and use of the return difference matrix, February, 1980, submitted to IEEE Trans. Aut. Contr.

[39] J. H. WILKINSON, Error analysis of direct methods of matrix inversion, J. ACM., 8(1961), pp. 281-330.

[40] J. E. DENNIS, JR., Private communication, (1978).

[41] J. E. DENNIS, JR., D. M. GAY, and R. E. WELSCH, An adaptive nonlinear least squares algorithm, NBER Working Paper 196, 1977, submitted to ACM Transactions on Mathematical Software.

[42] J. E. DENNIS, JR., Nonlinear least squares and equations, in D. Jacobs (Ed.), The State of the Art of Numerical Analysis, Academic Press, London, 1977.

[43] W. COWELL (Editor), Portability of numerical software, Oak Brook, 1976, Lect. Notes in Comp. Sci., Vol. 57, Springer-Verlag, New York, 1977.

[44] B. G. RYDER, The PFORT verifier: user's guide, CS Tech. Rep't. 12, Bell Labs., 1975.

[45] T. J. AIRD, The FORTRAN converter user's guide, IMSL, 1975.

[46] D. COLEMAN, P. HOLLAND, N. KADEN, V. KLEMA, and S. C. PETERS, A system of subroutines for iteratively reweighted least squares computations, NBER Working Paper 189, 1977, submitted to ACM Transactions on Mathematical Software.

Block Lanczos and Control Theory

D. L. Boley and G. H. Golub

A common problem in Control Theory involves the computation of the controllable subspace of a linear dynamic system. The Block Lanczos Algorithm can be used to construct an algorithm to compute this subspace. Such an algorithm, though frequently described in the literature, can be explained very simply in terms of the Lanczos procedure. In fact, for large sparse problems, it is appropriate to actually implement the computation of the subspace using a variant of the Block Lanczos Algorithm.

We will discuss only the case of a linear time-invariant dynamic system

$$\mathrm{dx}/\mathrm{d}t = A\mathrm{x} + B\mathrm{u}$$

where A is a $n \times n$ matrix, B is an $n \times p$ matrix, x is an n-vector of states, and u is a p-vector of inputs. The problem is to construct a basis for the controllable subspace for this system. In terms of linear algebra, the problem is to find the smallest invariant subspace of A that contains the columns of B treated as n-vectors. This is equivalent to determining the range space of the matrix

$$C = (B \;\; AB \;\; A^2B \;\; . \; . \; .),$$

a so-called *Krylov* sequence. Without somehow orthogonalizing the blocks, the problem of computing the rank of such a system can often be extremely ill-conditioned. Hidden rank degeneracies may exist among the blocks of C.

Assume without loss of generality that B has orthonormal columns. Then, in brief, the Block Lanczos algorithm will construct two matrices

$$Q = (Q_1 \;\; Q_2 \;\; Q_3 \;\; . \; . \; .)$$

and

$$H = (H_{ij}),$$

where H_{ij} denotes the i-j^{th} block of H, a block upper Hessenberg matrix. The method can be summarized by observing that, at each step, we are essentially constructing the blocks $AB\ A^2B \ldots$ of C, but also orthogonalizing each block against the preceding blocks. Because such orthogonalization may leave each block with a smaller rank, we must modify the Lanczos Algorithm to allow the block size to change with the rank dynamically.

We sketch the algorithm:

Start with $Q_1 = B$. Then at the i^{th} stage, $i=1, 2, \ldots,$ we compute as follows:

(a) Let $H_{ji} = Q_j^T A Q_i$, for $j=1, \ldots, i$.

(b) Let $Z = AQ_i - Q_iH_{ii} - Q_{i-1}H_{i-1,i} - \ldots - Q_1H_{1i}$. Then Z will be orthogonal to Q_j, $j=1, \ldots, i$.

(c) Decompose Z into $Z=Q_{i+1}H_{i+1,i}$, where Q_{i+1} is a set of orthonormal columns whose span is the same as that of Z, and $H_{i+1,i}$ is a matrix of full rank. In contrast to the classical Block Lanczos Algorithm, the number of columns in Q_{i+1} is the same as the rank of Z. We do not extend Q_{i+1} with extra columns to give it the same size as Q_1.

We repeat the above iteration for $i=1, \ldots, k$, stopping when $\mathrm{rank}(Z)=0$.

It can easily be shown that at the i^{th} stage,

$$\mathrm{span}(Q_1, \ldots, Q_i) = \mathrm{span}(B\ AB \ldots A^{i-1}B).$$

At the last stage k, we will have the relationship

$$Q_{i+1}H_{i+1,i} = AQ_i - Q_iH_{ii} - Q_{i-1}H_{i-1,i} - \ldots - Q_1H_{1i},$$

for $i=1, \ldots, k-1$, and

$$0 = AQ_k - Q_kH_{kk} - Q_{k-1}H_{k-1,k} - \ldots - Q_1H_{1k},$$

from the last stage. Collecting terms, we get:

$$QH = AQ,$$

i.e., Q is a basis for an invariant subspace of A containing the vectors B. It can be easily demonstrated from the nature of this algorithm that Q represents the smallest such invariant subspace.

If the matrix A is too large to work with as a unit, then the Block Lanczos Algorithm may be the only practical approach for computing the invariant subspace, since all we need for this method is to be able to form the matrix–vector product Aq.

Research supported in part by U.S. Air Force Office of Scientific Research Grant No. AFOSR-79-0094 and National Science Foundation Grant No. MCS78-11985.

STRUCTURAL DETERMINATION IN LARGE SCALE SYSTEMS USING PRINCIPAL COMPONENT ANALYSIS.

M.J. DENHAM

Kingston Polytechnic, U.K.

ABSTRACT

One of the most important problems at present in the field of large scale systems is that of determining a structural decomposition of the system such that controller design can be carried out on subsystems of relatively low order and with a relatively small number of control inputs and controlled outputs. In this way, not only can well established methods of control system design be used but, perhaps more importantly, the complexity of the resultant controller is reduced and its overall operation more easily understood for the purposes of commissioning and subsequent tuning.

Any structural determination and decomposition technique must be closely linked to a design method or at least offer quantitative results with regard to stability of the overall system in relation to subsystem interactions. Of equal importance is the need for a reliable and robust method for computing the structural information, i.e. such that variations of the parameters of the system model within the degree of uncertainty which generally exists in any practically derived data do not result in large variations in the computed structural information.

The technique proposed in this paper defines system structure in terms of the relative influence of control input signals on the state space of the system, i.e. in terms of the subspaces reachable from each input, and the influence of state signals within these subspaces on the system outputs, i.e. in terms of the subspaces unobservable from each output. This definition of structure is somewhat arbitary; the word "structure" means many things to many people. However it does enable us to use well known geometric concepts from system theory and well known (but not by the control community) techniques from numerical linear algebra.

The ideas of principal component analysis were recently applied to systems in [1]. In this paper we adapt these ideas to the problem of structur determination and make use of the related singular value decomposition algorithm, as implemented in [2], to provide (hopefully) robust computational results.

The approach to structure determination developed in the paper is based on a visual image of the regions of the system state space reachable from each input with unit norm input signal as ellipsoids with semi-axes determined by the principal component magnitudes and vectors of the matrix of piecewise continuous signals $\exp(At)b_i$, where A and b_i are defined by the usual linear system model:

$$x(t) = Ax(t) + Bu(t) \; ; \quad y(t) = Cx(t)$$

(b_i denotes the i th. column of B). Similarly, the regions of the state space which are observable from each output with unit norm zero-input output response are characterised by ellipsoids with semi-axes determined by the principal component magnitudes and vectors of $\exp(A^T)c_i^T$, where c_i denotes the i th. row of C.

We use component magnitudes to determine approximate subspaces for the reachable and observable subspaces corresponding to each input/output pair. The smallest angle between these approximating subspaces is then determined using the method described in [3]. This angle is then taken as a measure of the influence of each input on each output via the coinciding regions of the relevant reachable and observable subspaces. Unfortunately, this approach is only valid apparently when the model states have specific physical meaning and regions of the state space can be deleted with the additional knowledge provided by engineering significance. In the case where only the inputs and outputs are of physical significance, it is necessary to transform the internal coordinate system of the model to, for example, the input-normal form proposed in [1]. We then consider only the component magnitudes relating to the subspaces observable from each output. If, for a specific input/output pair, these magnitudes are "small", we consider the subspace reachable from that input to be approximately unobservable from that output and hence the input has little influence on the output in that case.

The paper concludes by presenting some computational results using the approach described in the paper. These are based on the F100 turbofan jet engine model described and extensively studied in [4]. The results obtained appear to coincide with the input/output pairings chosen by others using methods largely based on physical reasoning. There is hope therefore that the proposed approach will provide a useful algorithmic guide in those cases, which are common in large scale systems, where physical intuition is lacking.

References

1. Moore, B.C., "Principal component analysis in linear systems: controllability, observability,and model reduction", IEEE Trans. Automatic Control, to appear.
2. Garbow, B.S. et al., "Matrix eigensystem routines-EISPACK guide extension", Lecture Notes in Computer Science, 51, Springer-Verlag, NY.
3. Bjorck, A. and Golub, G.H., "Numerical methods for computing angles between linear subspaces", Math. Comput., 27, 579-594, 1973.
4. Sain, M.K. et al., (eds), "Alternatives for linear multivariable control", National Engineering Consortium Inc., Chicago, 1978.

A GENERALIZED EIGENVALUE APPROACH FOR SOLVING RICCATI EQUATIONS*

P. Van Dooren#

Abstract

A numerically stable algorithm is derived to compute orthonormal bases for any deflating subspace of a reguler pencil $\lambda B-A$. The method is based on an update of the QZ-algorithm, in order to obtain any desired ordering of eigenvalues in the quasi-triangular forms constructed by this algorithm.

As application we discuss a new approach to solve Riccati equations arising in linear system theory, where the computation of deflating subspaces with specified spectrum plays a crucial role. The standard eigenvalue approach for solving Riccati equations is by now well established but involves the inversion of matrices that are not necessarily well conditioned. E.g. for the optimal control problem of the plant $\dot{x} = Ax + Bu$, one wants the positive definite solution of the Riccati equation $-PBR^{-1}B'P + A'P + PA + Q = 0$ (where Q and R define the cost function). This can be obtained by computing the stable eigenspace χ_s of the Hamiltonian

$$H = \begin{bmatrix} A & BR^{-1}B' \\ -Q & -A' \end{bmatrix}$$

but R may be nearly singular. We show how to construct a pencil $\lambda\tilde{E}-\tilde{A}$ with the above space χ_s as stable deflating subspace, and such that only stable transformations are performed on the data $\{A,B,Q,R\}$. A similar approach is also given for the optimal control problem in discrete time and for the spectral factorization problem in continuous time and discrete time.

* This research was supported by the National Science Foundation under Grant ENG78-10003 and by the U.S. Air Force under Grant AFOSR-79-0094

Dept. El. Eng. and Dept. Comp. Sc., Stanford University, Stanford, CA 94305
Present address : Philips Research Lab., Av. Van Becelaere 2, Box 8, B-1170 Brussels, Belgium.

References

[1] B. Anderson, J. Moore, "Optimal filtering", Prentice Hall, New Jersey, 1979.

[2] B. Anderson, S. Vongpanitlerd, "Network analysis and synthesis. A modern systems approach", Prentice Hall, New Jersey, 1972.

[3] A. Emami-Naeini, G. Franklin, "Design of steady state quadratic loss optimal digital controls for systems with a singular system matrix", in Proceedings 13th Asilomar Conf. Circ. Syst. & Comp., pp. 370-374, Nov. 1979.

[4] G. Franklin, J. Powell, "Digital control of dynamic systems", Addison-Wesley, New York, 1979.

[5] G. Golub, J. Wilkinson, "Ill-conditioned eigensystems and the computation of the Jordan canonical form", SIAM Rev., vol. 18, pp. 578-619, Oct. 1976.

[6] T. Kailath, "Linear systems", Prentice Hall, New Jersey, 1980.

[7] H. Kwakernaak, R. Sivan, "Linear optimal control systems", Wiley Interscience, 1972.

[8] A. Laub, "A Shur method for solving algebraic Riccati equations", IEEE Trans. Aut. Contr., Vol. AC-24, pp. 913-921, Dec.1979.

[9] C. Moler , G. Stewart, "An algorithm for generalized matrix eigenvalue problem", SIAM J. Num. Anal., Vol. 10, pp. 241-256, April 1973.

[10] T. Pappas, A. Laub, N. Sandell Jr., "On the numerical solution of the discrete time algebraic Riccati equation", to appear in IEEE Trans. Aut. Contr.

[11] A. Ruhe, "An algorithm for numerical determination of the structure of a general matrix", BIT, Vol. 10, pp. 196-216, 1970.

[12] G. Stewart, "On the sensitivity of the eigenvalue problem $Ax=\lambda Bx$", SIAM J. Num. Anal., Vol. 9, pp. 669-686, Dec. 1972.

[13] G. Stewart, "Algorithm 506 : HQR3 and EXCHNG. Fortran subroutines for calculating and ordering the eigenvalues of a real upper Hessenberg matrix", ACM TOMS, Vol. 2, pp. 275-280, Sept. 1976.

[14] P. Van Dooren, "A generalized eigenvalue approach for solving Riccati equations, Rept. NA-80-02, Dept. Comp. Sc., Stanford Univ., Stanford, Jul. 1980.

[15] J. Wilkinson, "The algebraic eigenvalue problem", Oxford University Press, London, 1965.

[16] M. Wonham, "On a matrix Riccati equation of stochastic control", SIAM J. Contr., Vol. 6, pp. 681-697, Nov. 1968.

V. Strejc

Square root filtering and UD-decomposition

In many practical problems there arises the problem of solving an overdetermined ill-conditioned set of algebraic equations. In control field such a solution concerns for example the parameter estimation evaluating the measured input and output controlled system variables or controlled optimization according to quadratic cost functions. To circumvent the difficulty of the solution it is possible to apply the method for propagating the error covariance matrix in a square root form. It provides twice the effective precision of the conventional filter implementation and it is distinguished by outstanding numerical characteristics and relative simplicity [1], [2].

The straightforward calculation using the orthogonal matrices of elementary rotations transforms the rectangular data matrix Z into the triangular information matrix F which enables to calculate the desired parameters by the so called "back run" substituting thus the matrix inversion.

A significant simplification [3] can be reached when the same square root approach is applied for triangulization of the regression matrix G, where $G\ G^T = (Z\ Z^T)^{-1}$.

In both mentioned cases one square root extraction is needed when one element of the data matrix Z is to be zeroed. Alternatively the Cholesky algorithm [2] may be used for the triangulization of the data matrix. In this case, the square root extraction is needed only for the calculation of the diagonal elements of the information matrix F.

If UD-decomposition [4] is properly applied, no root extraction is needed at all. For single-input/single-output systems the required parameters are obtained directly as elements of one block of the regression matrix G.

[1] Kaminski, P.G., A.E. Bryson, and S.F. Schmidt (1971). Discrete square root filtering. A survey of current techniques. IEEE Trans. Autom. Control, AC-16, 727-735.

[2] Strejc, V. (1980). Least squares parameter estimation. Automatica, Special Section of the 1980 September issue.

[3] Peterka, V. (1975). A square root filter for real time multivariate regression. Kybernetika, 11, 53-67.

[4] Clarke, D.W., and P.J. Gawthrop (1979). Implementation and application of microprocessor-based self-tuners. IFAC Symp. on Identification and Syst. Param. Estimat., Darmstadt, Vol. 1, 197-204.

MATLAB -- An Interactive Matrix Laboratory

MATLAB is an interactive computer program that serves as a convenient "laboratory" for computations involving matrices. It provides easy access to matrix software developed by the LINPACK and EISPACK projects. It is expected that one of MATLAB's primary uses will be in the classroom. The program is written in Fortran and is designed to be readily installed under any operating system which permits interactive execution of Fortran programs.

Prof. Cleve Moler
Department of Mathematics
University of New Mexico
Albuquerque, NM 87131
505-277-4110

NUMERICAL INTEGRATION OF DIFFERENTIAL EQUATIONS OVER DISCONTINUITIES -- SIMULATION AND REAL-TIME APPLICATIONS

by: François E. Cellier
Institute for Automatic Control
The Swiss Federal Institute of Technology Zurich
ETH - Zentrum
CH-8092 Zurich
Switzerland

The simulation of discontinuous models is a very common task which finds many applications in engineering disciplines. By use of the currently available continuous simulation languages (like CSMP-III, ACSL, DARE-P), the step size control algorithm is "misused" to detect and locate discontinuities. In this presentation, it will be shown that this solution is extremely inefficient (ineconomic) and even dangerous.

A better way of handling discontinuities is proposed. Discontinuous models may be separated into continuous submodels (timewise) which are accompanied by conditions when to switch to another submodel. It is shown that the most natural way to look at such systems is a combined continuous and discrete event approach.

Efficient methods to interpolate back to the unknown event times (at which the discontinuities take place) are discussed including the inverse Hermite' interpolation and the use of the Nordsieck vector. It is shown what conditions the models have to fulfil to be tackled by these algorithms (canonical form) and what transformations exist to bring models into this desired form.

The limitations of the discussed methodology are discussed. These have mostly to do with inseparable discontinuities (like indifferentiable functions, e.g. noise).

Up to this point, we have discussed particular numerical techniques to handle difficult simulation problems. Bridging the gap to control theory, it is now illustrated how the previously described methods can be profitably applied to control problems. Typical applications include air-craft simulators, and the economic dispatch problem in power systems control.

At the example of state feedback control of an electro-mechanical system (DC-motor driving a heavy rotating mass through a spring) it is shown that for time critical applications the previously discussed general-purpose algorithms are no longer applicable. Another special-purpose algorithm which may be applied to some of these problems is shortly described. When reasonably inexpensive 32 bit microprocessors shall become available in a few years from now, it is expected that many more applications can be handled by the robust general-purpose algorithms advertised in this presentation.

Zurich, September 11, 1980

François Cellier

Dr. François E. Cellier
Lecturer of Simulation Techniques

Workshop on Numerical Methods in Automatic Control, *Sept. 23-25, 1980, Lund, SWEDEN.*

NUMERICAL SOLUTION OF RICCATI AND CHANDRASEKHAR EQUATIONS ARISING IN OPTIMAL CONTROL OF DISTRIBUTED PARAMETER SYSTEMS

Michel SORINE
INRIA, B.P.105
78150 Le Chesnay, France

1. INTRODUCTION.

Consider the quadratic optimization problem for the system $\dot{y} + Ay = Bv$, $y(t) = h$ in a general Hilbert space setting. Under some natural assumptions ([1], [2]) we know that there exists a feedback law $u(t) = -K(t)y(t)$ minimizing the cost

$$J(v) = \int_t^T \{|Cy|^2 + |v|^2\}\, dt$$

The gain K is given by $K = B^*P$ where P is a solution, in some restricted sense, of the Riccati equation :

$$\text{(1)} \qquad -\dot{P} + PA + A^*P + PBB^*P = C^*C, \quad P(T) = 0$$

K is also a formal solution of the Chandrasekhar equations which are for the time invariant control problem :

$$\text{(2)} \qquad \begin{cases} \dot{K} = -L^*L & K(T) = 0 \\ -\dot{L} + L(A + BK) = 0 & L(T) = C \end{cases}$$

We mention in this abstract and shall present with more details some attempts to make precise the setting of (1) and (2) and to compute their solutions.

2. THE RICCATI EQUATION APPROACH VIA THE CONTROL PROBLEM.

In studying (1) or (2) difficulties arise because A and in some cases B and C are unbounded operators. To avoid some of these difficulties Lions [1] has studied the state-costate equations associated with the control problem and has shown regularity results for the linear mapping $y(t) \to p(t)$ (p being the costate). This allowed him to define P(t) and to show that this operator is in some sense the unique solution of (1).

In the same spirit Nedelec ([3], [4], see also [2]) has approximated the control problem, has defined the related Riccati operator and then was able to show that this finite dimensional operator was a stable convergent approximation of the original one.

Along the same line, for generalizations or particular problems (control of evolution or hyperbolic equations, steady state Riccati equation ...) see [2], [5], [6], [7], [8], [9], [10] and the references therein.

3. THE RICCATI EQUATION DIRECT APPROACH.

Direct studies of generalized Riccati equations :

$$\dot{P} + AP + PB + \Phi(P) = F \quad , \quad P(0) = P_o \tag{3}$$

have been done in various Hilbert space settings. Temam [11] was looking for Hilbert-Schmidt solutions when $\Phi(P) = P^2$ and F and P_o were Hilbert-Schmidt. He found a variational formulation for (3) leading to numerical schemes. Da Prato [12], using the semi-group theory, proved the existence and uniqueness of the solution of (3) with more general Φ.

Tartar [13] (see also [14]) has extended these results to still more general A, B and Φ, using various techniques (in particular existence of upper and lower solutions jointly with the use of monotone methods).

4. CHANDRASEKHAR EQUATIONS.

They were stated for distributed parameter systems in [15] (time invariant case) and [16] (time varying case) without proof. A precise framework and proofs are available, for some parabolic time-invariant operators in [17], [18] along with numerical schemes.

The approach chosen is similar to that of Lions : K and L are first defined as : $K(t) : y(t) \to -u(t)$, $L(t) : y(t) \to Cy(T)$ then (2) is shown to be verified in some sense and Nedelec's method provides with numerical schemes. Partial results will be presented for the time varying case.

5. REFERENCES.

[1] J.L. LIONS, Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles. Dunod 1968.

[2] A. BENSOUSSAN, M. DELFOUR, S.K. MITTER, to appear.

[3] A. BENSOUSSAN, A. BOSSAVIT, J.C. NEDELEC, Approximations des problèmes de contrôle, Cahier de l'IRIA n°2, 1970.

[4] J.C. NEDELEC, Thèse d'Etat, Paris, 1970.

[5] R. CURTAIN, A.J. PRITCHARD, The infinite dimensional Riccati equation for systems defined by evolution operators, SIAM J. Control, Vol.14, n°5, 1976.

[6] J.S. GIBSON, The Riccati integral equations for optimal control problems on Hilbert spaces, SIAM J. Control, Vol. 17, n°4, 1979.

[7] D.L. RUSSEL, Quadratic performance criteria in boundary control of linear symmetric hyperbolic systems, SIAM J. Control, Vol. 11, n°3, 1973.

[8] A. MORREEUW, Contrôle optimal et approximation numérique des systèmes de Friedrichs, Centre de Mathématiques Appliquées, Ecole Polytechnique, n°12, 1976.

[9] J. ZABCZYK, Remark on the algebraic Riccati equation in Hilbert space, J. Applied Math. and Optimization 2, 1976.

[10] M. SORINE, Sur l'équation de Riccati stationnaire associée au problème de contrôle d'un système parabolique, C.R.A.S, Paris, t.287, Série A-445, 1978.

[11] R. TEMAM, Sur l'équation de Riccati associée à des opérateurs non bornés en dimension infinie, J. Functional Analysis 7, 1971.

[12] G. DA PRATO, Quelques résultats d'existence, unicité et régularité pour un problème de la théorie du contrôle, J. Math. Pures et Appl. 52, 1973.

[13] L. TARTAR, Sur l'étude directe d'équations non linéaires intervenant en théorie du contrôle optimal, J. Functional Analysis, 6, 1974.

[14] H.J. KUIPER, S.M. SHEW, Strong solutions for infinite dimensional Riccati equations arising in transport theory, SIAM J. Math. Anal, Vol. 11, n°2, 1980.

[15] J. CASTI, L.LJUNG, Some new analytic and computational Results for Operator Riccati equations, SIAM J. Control, Vol. 13, n°4, 1975.

[16] J.S. BARAS, D.G. LAINOTIS, Chandrasekhar algorithms for linear time varying distributed systems, Conf. Information Sciences and Systems, Mar.1976.

[17] M. SORINE, Schéma d'approximation pour des équations de Chandrasekhar intervenant en contrôle optimal, C.R.A.S., Paris, t. 284, Série A-61, 1977.
Sur les équations de Chandrasekhar associées au problème de contrôle d'un système parabolique, C.R.A.S., Paris, t. 285, série A-863 and série A-911, 1977.

[18] M. SORINE, Rapport IRIA-LABORIA n°267, 1977 and report to appear.

for Per Hagander

title: application of numerical boundary value methods to problems in optimal control

abstract: Necessary conditions for the solution of optimal control problems lead naturally to boundary value problems in differential equations. The application of various procedures for the numerical solution of such problems will be discussed. Difficulties arise where the boundary of the domain is also to be determined, or where the controls are discontinuous. First, ordinary differential problems with continuous controls will be considered for both fixed and unfixed end-time cases. Extensions (i) to a class of unknown boundary problems in partial differential equations, and (ii) to some ordinary differential problems with discontinuous controls will be described. Techniques for singularly perturbed systems may also be examined briefly.

DR. N.K. NICHOLS
DEPT. OF MATHEMATICS
UNIVERSITY OF READING
READING, BERKS, U.K.

Numerical treatment of a parabolic boundary-value control problem by means of semi-infinite programming.

Sven-Åke Gustafson
Dept of Numerical Analysis and
Computing Science
Royal Institute of Technology
S-100 44 STOCKHOLM 70, Sweden

Abstract We consider a class of heat - diffusion processes. Our goal is to determine the boundary condisions in such a manner that a specified temperature is approximated as closely as possible at a given time. This task is recast into the form of a linear approximation problem with linear constraints on the coefficients. Semiinfinite programming algorithms have proved efficient for the computational treatment of such problems.

References

A.G. Butkovskiy, Distributed Control Systems, American Elsevier, New York 1969

K. Glashoff and S.-Å. Gustafson, Numerical treatment of a parabolic boundary-value control problem, JOTA 19 (1976), 645-663

K. Glashoff and S.-Å. Gustafson, Einführung in die lineare Optimierung, Wissenschaftliche Buchgesellschaft, Darmstadt, 1978

S.-Å. Gustafson, On the numerical treatment of a multidimensional parabolic boundary-value control problem, in R. Bulirsch, W. Oettli and J. Stoer (Eds) Optimization and Optimal Control, Springer Lecture Notes in Mathematics, No 477 (1975), 121-126.

SOME FEATURES OF CHANDRASEKHAR EQUATIONS AND ORTHOGONAL TRANSFORMATIONS TECHNIQUES IN OPTIMAL CONTROL

by

Raymond R. GOREZ (*)

Chandrasekhar equations or algorithms based on orthogonal transformations can be used for solving linear-quadratic optimal control problems of continuous or discrete time-invariant systems.

These techniques have intrisic separation properties and result in a roughly linear relationship between the system dimension and the computational burden. Both are most valuable when designing model-following control systems and/or disturbances accomodating regulators.

Moreover, in dead-beat control problems, where the weigting factor of the control variable in the performance criterion is set equal to zero, orthogonal transformations are straightforward and can be obtained by inspection, resulting in paper and pencil calculation of the optimal control law, even for multivariable systems.

These properties are illustrated by several examples. The first one refers to the computer control of a two inputs - two outputs paper making machine; orthogonal transformations techniques are used for determining a dead-beat controller. The second example is concerned with the temperature regulation of a nuclear reactor in a 400 MW PWR power plant; continuous and discrete optimal controllers have been obtained either by Chandrasekhar equations or by orthogonal transformations techniques

References

1 P.G. KAMINSKY, A.E. BRYSON and S.F. SCHMIDT, "Discrete Square Root Filtering : A Survey of Current Techniques", *IEEE Trans. on Automatic Control*, vol. AC-16, pp. 727-736, 1971.

2 A. LINDQUIST, "A New Algorithm for Optimal Filtering of Discrete-Time Stationary Processes", *SIAM J. on Control*, vol. 12, pp. 736-746, 1974.

(*) University of Louvain
Lab. of Automatic Control and Systems Analysis
Bâtiment Maxwell, Place du Levant 3
B-1348 LOUVAIN LA NEUVE Belgique.

3 T. KAILATH, "Some Chandrasekhar-Type Algorithms for Quadratic Regulators", *Proc. 1972 IEEE Conf. on Decision and Control*, New-Orleans, pp. 219-223.

4 J.L. CASTI, "Dynamical Systems and Their Applications : Linear Theory", *Academic Press*, New-York, 1977.

5 H.J. PAYNE and L.M. SILVERMAN, "Matrix Riccati Equations and Systems Structure", *Proc. 1973 IEEE Conf. on Decision and Control*, San Diego, pp. 558-563.

6 L.M. SILVERMAN, "Discrete Riccati Equations : Alternative Algorithms, Asymptotic Properties, and Systems Theory Interpretations" in : Control and Dynamic Systems, ed. by G.T. Leondes, vol. 12, *Academic Press*, New-York, 1976.

7 C. FOULARD, S. GENTIL et J.P. SANDAZ, "Commande et Régulation par Calculateur Numérique", *Eyrolles*, Paris, 1977.

9 R.R. GOREZ "Orthogonal Transformations and Square-root Algorithms in Linear-Quadratic Optimal Control of Discrete-Time Systems", to appear in *Proc. of the Int. Conf. on Information Sciences and Systems*, Patras, 1979.

Andrzej P. Wierzbicki
System and Decision Sciences Area
International Institute for Applied Systems Analysis

TRAJECTORY OPTIMIZATION - THEORETICAL BACKGROUND AND PRACTICAL MOTIVATIONS FOR INFINITE-DIMENSIONAL OBJECTIVES IN DYNAMIC OPTIMIZATION AND CONTROL.

The typical objective of an optimal control or dynamic optimization problem is to optimize a scalar performance functional; less frequently, also vectors of performance functionals are considered in multiobjective optimization. However, there are practical problems--mostly related to the use of dynamic control models in economic planning--where the objectives are stated in terms of desirable trajectories. If the goal would be to approximate the desired trajectory from both sides, then the problem could be equivalently stated as a typical approximation problem. However, in many cases the desired trajectories have the meaning of aspiration levels: if possible, they should be exceeded.

The paper presents a mathematical formulation of such a trajectory optimization problem, various theoretical approaches to this problem, conditions of optimality, remarks about possible computational approaches and examples of actual computations.

Comparison of several optimization methods not requiring exact one-dimensional searches - Numerical experiences

Dipl.-Ing. B. Asselmeyer
Fachgebiet Regelsystemtheorie
Technische Hochschule Darmstadt
Schloßgraben 1
D-6100 Darmstadt

Darmstadt, den 27. Juni 1980

Abstract

Optimization methods using a gradient-procedure with super-linear convergence properties without using second derivatives (for instance conjugate-gradient or variable metric methods), require mostly one-dimensional searches for the determination of the stepsize. But in these one-dimensional searches many function evaluations are necessary for the exact determination of the minimum. Especially for optimal control problems, where each function evaluation is the integration of a set of differential equations, this is a very time-consuming part of the program.

However, some algorithms have been proposed (see for instance (1) - (8)), which do not require exact searches, and yet do not loose the property of super-linear convergence, thus allowing to reduce the number of function evaluations in this part of the program drastically.

In this work, some of these algorithms have been compared with well-known conventional ones (like Fletcher-Reeves and Davidon-Fletcher Powell algorithm).
The results show, that indeed the algorithms yield a reduction of the necessary number of function-evaluations for obtaining the same accuracy in the cost function. In addition conventional algorithms have been used with inexact searches, and it was found, that they show advantages too: Typically for the complete optimization less function evaluations are required, than with one dimensional searches of high accuracy, even if the number of optimization steps may be slightly higher.

References

[1] Huang, Chambliss
Quadratic Convergent Algorithms and One-dimensional Search Schemes
JOTA, 1973, p 175

[2] Ostrowski, Bereshinskii, Belyaera
Methods of minimizing functions of many variables that do not employ precise line searches
Avt. Vychislitel'naya Tekhnica, 1976, p 43

[3] Kawamura, Volz
On the rate of convergence of the conjugate gradient reset method with inaccurate linear minimizations
IEEE, Trans. on AC, 1973, p 360

[5] Nazareth
A conjugate direction algorithm without line searches
JOTA, 1977, p 373

[6] Shanno
Conjugate gradient methods with inexact searches
Math. of OR, 1976, p 244

[7] Davidon
Optimally conditioned optimization methods without line searches
Math. Prog., 1975, p1

[8] Oren
Self-scaling variable metric algorithm without line searches for unconstrained minimisation
Math. Comput., 1973, p 873

[9] *Hoffmann,*
Untersuchung verschiedener Möglichkeiten zur Optimierung ohne exakte Suche
Studienarbeit at the "Institut für Regelungstechnik" of the "Technische Hochschule Darmstadt",
supervised by B. Asselmeyer

Institut für Angewandte Mathematik ~~und Informatik~~
der Universität Bonn

53 BONN, den 14. July 1980
Wegelerstraße 6
Telefon 733424 ~ 733174

P. Zencke

Abstract:

A Self Starting Algorithm for the Identification of Positive Exponential Decay Components by Minimization of the Chebyshev Error

The Chebyshev error criterion has some advantages in Identifying exponential decay components from data series. This makes the Chebyshev solution easier to handle numerically than the common least square fit.
For instance, the uniqueness of the best positive exponential approximation is garanteed [1] , the linearization method is known to converge locally quadratic [2] and starting parameters can be produced by the embedding method [3] .
Nevertheless, known implementations of the linearization method [4] suffer from very poor global convergence properties. Also, the simplex method for solving the linearized problems is expensive and may fail because of the ill conditioning of the problem.
By parametrizing exponential sums as solutions of linear differential equations with constant coefficients, these difficulties can be overcome.
Some properties of the numerical realization of the differential equation approach [5] , [6] , [7] are discussed, in particular

- the setup of the linearized systems and how to solve them
- the use of polynomial equations to avoid the explicit integration of differential equations.

In many practical examples, the given algorithm has proved to be a fast and robust method for the identification of exponential decay components.

[1] Braess, D.: Über die Approximation mit Exponentialsummen, Computing 2 (1967), 309 - 321.

[2] Hettich, R., P. Zencke: Superlinear konvergente Verfahren für semi-infinite Optimierungsprobleme im stark eindeutigen Fall, Universität Bonn, Preprint SFB 72 No 354, 1980.

[3] Braess, D.: Die Konstruktion der Tschebyscheff-Approximierenden bei der Anpassung mit Exponentialsummen, J. Appr. Th. 3(1970), 261 - 273.

[4] Cromme, L.: Eine Klasse von Verfahren zur Ermittlung bester nichtlinearer Tschebyscheff-Approximationen, Num. Math. 25 (1976), 447 - 459.

[5] Kammler, D.W.: Characterization of best approximations by sums of exponentials, J. Appr. Th. 9 (1973), 173 - 191.

[6] Zencke, P.: Zur Parametrisierung reeller Exponentialsummen durch lineare Differentialgleichungen mit Parameterrestriktionen, in: Numerische Methoden der Approximationstheori Bd. 4 (edit. L.Collatz, G.Meinardus, H.Werner), Birkhäuserverlag, Basel-Stauttgart (1978), pp 320 - 344.

[7] Zencke, P.: Theorie und Numerik der Tschebyscheff-Approximation mit reell-erweiterten Exponentialsummen, Dissertation Bonn 1980.

April 12, 1980

Fitting empirical data by positive sums of exponentials

by

Axel Ruhe
Dept of Information Processing
University of Umeå, 901 87 Umeå, Sweden

Summary Least squares and maximum likelihood fitting of a positive sum of exponentials, to an empirical data series is discussed. A characterization in terms of a convex moment cone is used, to develop a globally convergent self starting algorithm. The numerical algorithm is described. Tests on empirical data series are reported, and different criteria for determining the number of terms are compared. The convex cone characterization gives an upper bound on the number of terms, while the cross validation method is used to test the statistical relevance of the sums that are computed.

AMS classification: Primary: 65D10
Secondary: 62F10, 90C25, 93B30

Key words: Exponential sums, compartment models, nonlinear least squares, cross validation, Gauss Newton method.

References

1. Cantor D.G. and Evans J.W. (1970), On approximation by positive sums of powers, SIAM J.Appl.Math., 18, 380-388.
2. Efron B. (1979), Computers and the theory of statistics: thinking the unthinkable, SIAM Rev., 21, 460-480.
3. Golub G.H., Heath M. and Wahba G. (1979), Generalized cross-validation as a method for choosing a good ridge parameter, Technometrics, 21, 215-223.
4. Jennrich R.I. and Bright P.B. (1976), Fitting systems of linear differential equations using computer generated exact derivatives, Technometrics, 18, 385-392.
5. Ruhe A. (1978), Fitting empirical data by positive sums of exponentials, Tech.Rep. UMINF-70.78, submitted for publication.

Workshop on Numerical Methods in Automatic Control
Lund 1980

CONTROL SYSTEM DESIGN USING POLYNOMIAL EQUATIONS
Numerical Aspects

Vladimír Kučera
Institute of Information Theory and Automation
Czechoslovak Academy of Sciences
182 08 Prague, Czechoslovakia

Abstract - Polynomial methods in control theory are receiving still growing attention. They gradually developed from frequency-domain concepts on recognizing their algebraic nature. The use of polynomial equations in control system design was pioneered by Volgin, Åström and Kučera.

In particular, promissing results were obtained in the design of linear, constant, finite dimensional, constrol systems. The principal idea is to reduce the design procedure to the solution of a Diophantine equation in polynomials or polynomial matrices. Many design problems can be solved in a uniform, simple and direct way using this approach.

The basic problem which can naturally be solved by polynomial equation is that of pole placement. Many problems including stabilization, model matching, decoupling, deadbeat control etc. can directly be reduced to this basic problem. The least squares control problems - both deterministic and stochastic - can also be treated as a pole placement when complemented with spectral factorization, which serves to specify the optimal pole locations.

The topic of this paper are the algorithmic and numerical aspects of the polynomial equation approach. Typically, in the SISO design problems, we are to find a minimum-degree solution X, Y of the equation

$$A\,X + B\,Y = C$$

in polynomials. Blankinship algorithm is recommended to calculate a particular solution; the minimum-degree solution is then obtained simply by the division algorithm. The discussion includes notes on complexity of this algorithm as well as numerical experience.

Spectral factorization consists in the following. Given a polynomial Q satisfying $Q = Q_*$ where the asterisk denotes the conjugate polynomial (defined for continuous-time and discrete--time systems in a different way) calculate a stable polynomial F such that

$$F F_* = Q .$$

There are efficient iterative schemes to perform this factorization. The discussion includes the algorithms of Rissanen, Bauer and Vostrý, the latter being recommended for its accuracy and quadratic convergence.

In the MIMO systems, we work with the polynomial matrices in place of polynomials. The Blankinship algorithm can easily be generalized to solve (one-sided) equations in polynomial matrices, and matrix versions of the spectral factorization algorithms are also available. There are other problems to solve, however. We have to calculate coprime matrix fractions, either left or right, and we often need them in reduced form. In general, matrix operations are much more complex than the polynomial ones and first numerical experience will be reported in this area.

References

Blankinship, W.A.: A new version of the Euclidean algorithm. Amer. Math. Monthly 70 (1963), 742-745.

Kučera, V.: Discrete Linear Control - The Polynomial Equation Approach. Wiley, Chichester 1979.

Vostrý, Z.: New algorithm for polynomial spectral factorization with quadratic convergence. Kybernetika 11 (1975), 415-422 and 12 (1976), 248-259.

Numerical Problems in Multivariable Synthesis using the Polynomial Matrix Approach.

Lars Pernebo
Lund Institute of Technology, Sweden

If the polynomial matrix approach is used in multivariable synthesis the following problem occurs frequently. Given a polynomial matrix $P(s)$. Find a unimodular polynomial matrix $N(s)$ and a polynomial matrix $L(s)$, with linearly independent columns, such that

$$P(s)N(s) = (L(s) \quad 0)$$

holds. The problem can be solved by performing a series of elementary column operations on $P(s)$. These operations are not unique and it is difficult to see how to make the choices from a numerical point of view.

Another approach will be presented here. The rational matrix $\frac{1}{\lambda}P(\frac{1}{\lambda})$ will be represented in state space form (A,B,C). An equivalent problem is then the following. Find a state feedback F, such that

i) $A+BF$ has all its eigenvalues at the origin.

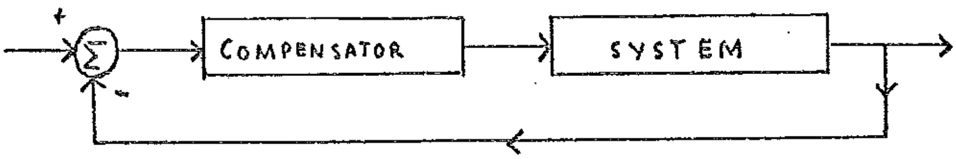
ii) There is an $A+BF$ invariant subspace in $\ker C$, which contains $\mathcal{R}^*$, the maximal (A,B)-controllability subspace in $\ker C$.

Furthermore, find a nonsingular input transformation G, such that $BG = (B_1 \quad B_2)$, where $\mathcal{B}_2 = \mathcal{R}^* \cap \mathcal{B}$. Here script letters represent range spaces of the corresponding matrices.

A frequency response algorithm to calculate dynamic compensators to optimise the closed-loop response of a multivariable system. Sept. 25 10.15

by J.M.Edmunds

In this paper a method is described for designing linear multivariable control schemes which have a closed-loop frequency response as close as possible, in a least squares sense, to a 'desired' response. Examples are then used to illustrate how the 'desired' response may be chosen.



There are 4 main steps in using this design method.

1) Specify the desired closed-loop response.
2) Choose the structure of the compensator.
3) Find the beszt coefficents for this compensator structure.
4) Analyse the resulting system and if it is not satisfactory change the compensator structure, or change the desired response.

Notation

$G(s)$	Plant frequency response	$H'(s)$	Desired closed-loop response
$G'(s)=(I-H')^{-1}H'$	Desired open loop	$H(s)$	Closed-loop response
$K(s)$	Compensator	$E_c(s)=H'(s)-H(s)$	Closed-loop error

Algorithm

Once the denominators of the controller have been specified, the problem is to choose the best parameter values for the numerators of the transfer function matrix K(s). Unfortunately the closed-loop problem cannot be reduced to a linear problem since the closed-loop error $E_c(s)$ to be minimised is given by

$$H' = H + E_c \tag{1}$$

$$= (I+GK)^{-1}\, GK + E_c \tag{2}$$

The occurence of K(s) in two places in equation 2 means that the equation cannot be rearranged to give the correct form for a linear least squares problem. However an approximation to the solution can be obtained in the following manner. It can be shown (Edmunds 1979) that equation 2 implies:-

$$E_c = (I-H)\ (G'-GK)\ (I-H') \tag{3}$$

rewriting this becomes

$$(I-H)\ G'\ (I-H') = (I-H)\ GK\ (I-H') + E_c \tag{4}$$

Assuming that the final closed-loop error is going to be small, H(s) in equation 4 can be approximated by the desired closed-loop response H'(s), reducing equation 4 to a linear least squares form. The most significant difference from usual is the term (1-H'(s)) post-multiplying the K(s) on the right hand side of the equation. When the (I-H'(s)) term is diagonal the columns $\tilde{k}_m(s)$ of the compensator can be calculated separately, and the (I-H'(s)) term is accounted for by multiplying each of the elements in G(s) by $(1-H'_{m,m}(s))$. The separate calculation of the columns of K(s) is particularly useful on systems with many inputs and outputs, since much less computation is required to find several small sets of parameters than one large one. In the unusual case of (I-H'(s)) not being diagonal the problem unfortunately does not simplify in such a nice manner, and all of the columns of the controller have to be calculated simultaneously, as follows. Equation 4 has the form

$$Y = AKB + E \tag{5}$$

which can turned to the usual least squares form

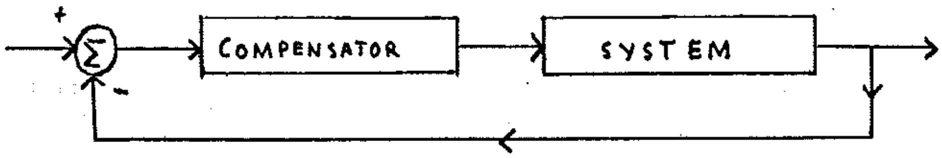
$$\tilde{y} = A'\tilde{k} + \tilde{e} \tag{6}$$

where $\tilde{k}$ is the vector of all the coefficients to be calculated and $\tilde{y}$ has n times as many rows as Y. The rows of the matrix A' are formed by taking a set of copies of the relevant row of A and multiplying each of these copies by an element from B (Glover, 1973).

A frequency response algorithm to calculate dynamic compensators to optimise the closed-loop response of a multivariable system. Sept. 25 10.15

by J.M.Edmunds

In this paper a method is described for designing linear multivariable control schemes which have a closed-loop frequency response as close as possible, in a least squares sense, to a 'desired' response. Examples are then used to illustrate how the 'desired' response may be chosen.



There are 4 main steps in using this design method.
1) Specify the desired closed-loop response.
2) Choose the structure of the compensator.
3) Find the beszt coefficents for this compensator structure.
4) Analyse the resulting system and if it is not satisfactory change the compensator structure, or change the desired response.

Notation

$G(s)$	Plant frequency response	$H'(s)$	Desired closed-loop response
$G'(s)=(I-H')^{-1}H'$	Desired open loop	$H(s)$	Closed-loop response
$K(s)$	Compensator	$E_c(s)=H'(s)-H(s)$	Closed-loop error

Algorithm

Once the denominators of the controller have been specified, the problem is to choose the best parameter values for the numerators of the transfer function matrix K(s). Unfortunately the closed-loop problem cannot be reduced to a linear problem since the closed-loop error $E_c(s)$ to be minimised is given by

$$H' = H + E_c \quad (1)$$
$$= (I+GK)^{-1}\,GK + E_c \quad (2)$$

The occurence of K(s) in two places in equation 2 means that the equation cannot be rearranged to give the correct form for a linear least squares problem. However an approximation to the solution can be obtained in the following manner. It can be shown (Edmunds 1979) that equation 2 implies:-

$$E_c = (I-H)\,(G'-GK)\,(I-H') \quad (3)$$

rewriting this becomes

$$(I-H)\,G'\,(I-H') = (I-H)\,GK\,(I-H') + E_c \quad (4)$$

Assuming that the final closed-loop error is going to be small, H(s) in equation 4 can be approximated by the desired closed-loop response H'(s), reducing equation 4 to a linear least squares form. The most significant difference from usual is the term (1-H'(s)) post-multiplying the K(s) on the right hand side of the equation. When the (I-H'(s)) term is diagonal the columns $\tilde{k}_m(s)$ of the compensator can be calculated separately, and the (I-H'(s)) term is accounted for by multiplying each of the elements in G(s) by $(1-H'_{m,m}(s))$. The separate calculation of the columns of K(s) is particularly useful on systems with many inputs and outputs, since much less computation is required to find several small sets of parameters than one large one. In the unusual case of (I-H'(s)) not being diagonal the problem unfortunately does not simplify in such a nice manner, and all of the columns of the controller have to be calculated simultaneously, as follows. Equation 4 has the form

$$Y = AKB + E \quad (5)$$

which can turned to the usual least squares form

$$\tilde{y} = A'\tilde{k} + \tilde{e} \quad (6)$$

where $\tilde{k}$ is the vector of all the coefficients to be calculated and $\tilde{y}$ has n times as many rows as Y. The rows of the matrix A' are formed by taking a set of copies of the relevant row of A and multiplying each of these copies by an element from B (Glover, 1973).

Routh-Criterion for obtaining the roots of polynomials

K. Zeiske

Department of Electrical Engineering

Ruhr-Universität Bochum

D-4630 Bochum, West-Germany

Abstract

It can be shown that the well-known Routh-Criterion most used for proofing the stability of control systems can be used to compute the roots of a polynomial f(s) which has real coefficients. For this purpose it will be proofed that the roots of the polynomial f(s) which are lying of the imaginary axis are also roots of all polynomials of lower order than f(s) which are constructed by applying the Routh-Criterion to the polynomial f(s). Thus the Routh-Criterion gives a polynomial in ω^2, if there are such roots, and this polynomial in ω^2 can be used for the determination of these roots of f(s) which are lying on the imaginary axis.

Combining the Routh-Criterion with a transformation $s \to s + a$, where a is real, yields the possibility to shift any root of f(s) to the imaginary axis, where the imaginary part of the root can be computed, whereas the parameter a which is determined iteratively gives the real part of the root considered. The criterion of the iteration is the answer of the Routh-Criterion "stable" or "unstable", if the most right root of f(s) is considered, or is the sign and magnitude of a appropriately chosen element of the first column of the Routh-array.

The comparison of computed examples with results given in references shows a good performance of the procedure presented. Especially roots of higher order can be computed with nearly the same exactness as roots of order one.

AUTOPACK - A robust software for automatic control

Alain BARRAUD

Laboratoire d'Automatique de Grenoble

B.P. 46

38402 - SAINT-MARTIN-D'HERES - FRANCE

=:=:=:=:=:=:=

Abstract.

Considerable efforts have been made for almost ten years in the development of high quality software. Needs and availability of general purpose libraries increasingly answering this criterion are now well understood.

AUTOPACK represents the joint effort of two researchers (1) since 1975, in order to produce a self-contained FORTRAN package in the specialized area of automatic control. This work began at the University of Nantes (France) on a medium size IBM 370 machine and is now continued at the University of Grenoble (France) on a 16 bit Norsk Data computer.

The first part of this communication concerns the general description of the package : its modular and hierarchical structure, the organization of its documentation and its use via specialized files of names and keywords and utility programs the purpose of which being to help users in selecting appropriate modules for a given problem (2). Some remarks are then presented around the coding aspect and portability enhancement through the use of some FORTRAN tools such as POLISH (3), PFORT verifier (4) and DECS (5).

AUTOPACK is a relatively large package (> 25.000 FORTRAN lines today) which has been completely developed by the authors of (1) except for general purpose routines (such as non linear optimization, zero/root finding, linear algebra, eigenvalue etc...) which comme from well known libraries or packages EISPACK (6), LINPACK (7), HARWELL (8). This kind of routines has been usually rewritten to match style and modular structure (9) adopted for AUTOPACK.

The second part of the paper is devoted to automatic control methods and algorithms included in AUTOPACK, mainly in identification and control. See

(23) for details. Four particular numerical problems are then discussed in more details, namely : matrix exponential and related computation, Lyapunov equations, Riccati equations and parametric structural estimation problems.

Matrix exponential is computed via Padé approximant and scaling. For large spectrum matrix, block diagonalization (10) enhances the basic procedure performance in solving a set of sub-problems whose matrix spectrum is usually very narrow. Finally, an option has been introduced in order to obtain the expected number of correct figures of the computed exponential matrix. The last result is obtained through the permutation-perturbation method of Vignes (11) which is a very robust and quite general approach, but multiplies by 3 the total cost of computations (12).

Lyapunov equations are solved by reduction to real Schur form : Bartel-Stewart's algorithm for the continuous case (13) and author's method in the discrete case (14). Scaling/balancing and iterative refinement are treated in a way similar to the classical linear systems.

Two closed form approaches are discussed for Riccati equations (both for discrete and continuous cases) via matrix sign computation and ordered real Schur form reduction of Hamiltonian matrix (15), (16), (18). It is pointed out that the problem relating to singular transition matrix can be avoided without solving any generalized eigenvalue reduction (17). However, in the discrete case it must be noticed that only iterative square root (fast or not) methods are able to guarantee numerical symmetry and non negative property of the computed solution. Lastly, some parameter/structural identification schemes are presented on the basis of extended least squares method. It is pointed out that on one hand it is necessary to give up the classical form :

$$" \left\{ \begin{array}{lll} \textit{new parameters} & = & \textit{old parameters + correction vector} \\ \textit{time } t & & \textit{time } t-1 \end{array} \right\} "$$

and, on the other hand, sequential information square root algorithms offer a natural way to take into account possible changes in the parameter vector dimension (modification of order or Kronecker invariants). More details about implementation and properties of the method can be found in (19).

In conclusion, it can be said that AUTOPACK has been developed as an evolutive and maintained FORTRAN library. One of its first systematic applications is its global introduction in the interactive system ISER-CSD (20) which will be the subject of a second communication.

New developments and results around numerical confidence intervals are expected for Lyapunov and Riccati equations.Study of ad-hoc algorithms, using interval arithmetic, implemented via the FORTRAN preprocessor AUGMENT (21), (22), is planned in the near future as an alternative to Vignes' method.

REFERENCES

(1) A. BARRAUD, J. HANEN — "Design and development of an advanced numerical package in automatic control" International Conf. Syst. Sc. IV. Wroclaw (Poland) - 6-9 Sept. 1977.

(2) J. HANEN — "Conception et développement d'un logiciel d'aide à l'automaticien" Th. Docteur-Ingénieur, Université de Nantes, France - 15 Déc. 1978.

(3) J. DORRENBACHER et al. — "Polish, a Fortran program to edit Fortran programs" Report CU-CS-050-76, May 1976, University of Colorado,USA.

(4) B.G. RYDER, A.D. HALL — "The PFORT verifier" Computing Science Technical Report 12, Bell Laboratories 1979, Murray Hill, USA.

(5) D. SAYERS — "DECS, a declarative program implementation document" NAG Central office, Oxford, U.K.

(6) B.T. SMITH et al. — "Matrix Eigensystems Routines - EISPACK guide 2nd.ed." Lecture Note in Computer Science, vol. 6, Springer-Verlag - 1976.

(7) J.J. DONGARRA, C.B. MOLER — "LINPACK working note ≠ 13, Implementation guide for LINPACK" NEA Data Bank - Gif sur Yvette, France.

(8) M.J. HOPPER — "Harwell subroutine library, a catalogue of subroutine" Harwell Report AERE-R-7477. 1977.

(9) J. HANEN, A. BARRAUD — "Accurate basic modules in linear algebra" Tex. Conf. Math. Software - Contributed paper, Session III Soft. 29-31 March 1978, Austin, U.S.A.

(10) C.A. BAVELY, G.W. STEWART — "An algorithm for computing reducing subspaces by block diagonalization" Comp. Sc. Techn. Rep., University of Maryland, USA, 1976.

(11) M. LA PORTE, J. VIGNES — "Algorithmes numériques - Analyse et mise en oeuvre" Tome 1 - Technip (1974).

(12) J. ROCHE — "Application des approximants de Padé au calcul de l'exponentielle d'une matrice" Thèse de 3° Cycle, Université de Grenoble, Juin 1980.

(13) R.H. BARTELS, G.W. STEWART — "Solution of the matrix equation $AX+XB = C$, " C.A.C.M., vol. 15, pp. 820-826, 1972.

(14) A. BARRAUD — "A numerical algorithm to solve $A^TXA-X = Q$" IEEE Tr. Aut. Contr. Vol. AC-22, n°5, oct. 1977, pp.883-885.

(15) A. BARRAUD — "Investigation autour de la fonction signe d'une matrice - Application à l'équation de Riccati" RAIRO Aut. vol. 13, n°4, 1979. pp. 335-368.

(16) A. BARRAUD — "An accelerated Newton process to solve Riccati equation via matrix sign function" IFAC Symposium C.A.D. of Control Systems. Aug. 29-31, 1979, Zürich, Switzerland.

(17) A. BARRAUD — "Produit étoile et fonction signe de matrice - Application à l'équation de Riccati dans le cas discret" RAIRO, Aut., vol. 14, n°1, 1980 - pp. 55-85.

(18) A.J. LAUB — "A Schur method for solving algebraic Riccati equation" Lab. for Inform. and Decision Syst., MIT, Cambridge, USA Rep. LIDS - R.859. Oct. 1978.

(19) A. BARRAUD — "Identification des invariants fondamentaux - approches numériques". RCP 567,"Outils et Modèle mathématiques pour l'Automatique" 29-30 Nov. 1979 - E.N.S.M., Nantes, France.

(20) C. SULEYMAN, A. BARRAUD — "ISER - CSD an interactive system for education and research in control" Workshop on Numerical methods in Automatic Control, Lund Inst. of Tech. - Sept. 23-25, 1980 - Sweden.

(21) F.D. CRARY — "The AUGMENT precompiler, User Information" Tech. rep. 1469, University of Wisconsin, 1976, USA.

(22) R.G. WARD — "Implementation of the AUGMENT preprocessor and interval arithmetic on the DEC system..." Report U.S. Army Corps of Engineers,Aug. 1976.

(23) G. BORNARD, J.P. GAUTHIER — "Commande dynamique multivariable des systèmes industriels de production" Techn. Report L.A.G. 77-29, Nov. 1977, Grenoble, France.

=:=:=:=:=:=:=:=:=:=:=:=

A Language for Dynamical Models Based on General Equations

Hilding Elmqvist

Department of Automatic Control
Lund Institute of Technology
Lund, Sweden

Abstract

Dynamical models are originally stated as a set of general equations such as mass- and energy balances. However, most integration algorithms and simulation languages are based on an explicit formulation where the derivatives have been solved for. A model language, called Dymola, based on general equations is presented. A method is also given which transforms the equations to an algorithm consisting of assignment statements for the derivatives.

Dymola permits large models to be decomposed using a submodel concept. There is a concept cut, which associates a set of variables with a connection mechanism. The cuts can be used in a connection statement in order to describe the connection structure of a model. The connection statements are equivalent to simple equations involving the variables in the cuts.

The structure of the equations are studied in order to decide which variable should be solved for in each equation and to sort the equations for sequential computation. Systems of equations that must be solved simultaneously are also detected. The equations are then if possible transformed to assignment statements using formula manipulation.

A preprocessor will soon be available in Pascal which converts a Dymola model into an equivalent model in CSSL or Simnon.

References

Elmqvist H.: A Structured Model Language for Large Continuous Systems. Ph.D. Thesis, TFRT-1015, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden (May 1978).

Elmqvist H.: DYMOLA - A Structured Model Language for Large Continuous Systems, Proc. Summer Computer Simulation Conference, Toronto, Canada (July 1979).

Elmqvist H.: Manipulation of Continuous Models Based on Equations to Assignment Statements, Proc. IMACS Congress 1979 / Simulation of Systems, Sorrento, Italy (Sept 1979).

INTERACTIVE SYSTEMS FOR EDUCATION AND RESEARCH

IN

CONTROL SYSTEMS DESIGN

(ISER - CSD)

C. SULEYMAN - A. BARRAUD

Laboratoire d'Automatique de Grenoble
Institut National Polytechnique de Grenoble
B.P. 46, 38402 - SAINT-MARTIN-D'HERES

=:=:=:=:=:=:=:=:=:=:=

A B S T R A C T

It would be almost impossible to over-emphasize the importance of computers, to the understanding of engineering concepts and to engineering design. The use of simulation in analysis and design has been vital to the development of the profession. Digital computers play an increasingly important role in system design as time-sharing terminals, which allow engineers rapid interaction with the computer, become more prevalent, and as more and better computer aided design programs are developed. One of the major keys to this continued growth of computer-aided design will be the rapid graphical interaction between engineer and the computer with the tedium of drawing graphs removed the engineer will be able to examine a much boarder range of design possibilities in a shorter time and obtain a much better understanding of the system with which he is working. With the important role that these tools play in engineering, if is extremely important that universities make their students aware of the potential of these tools and prepare them for their use.

..../...

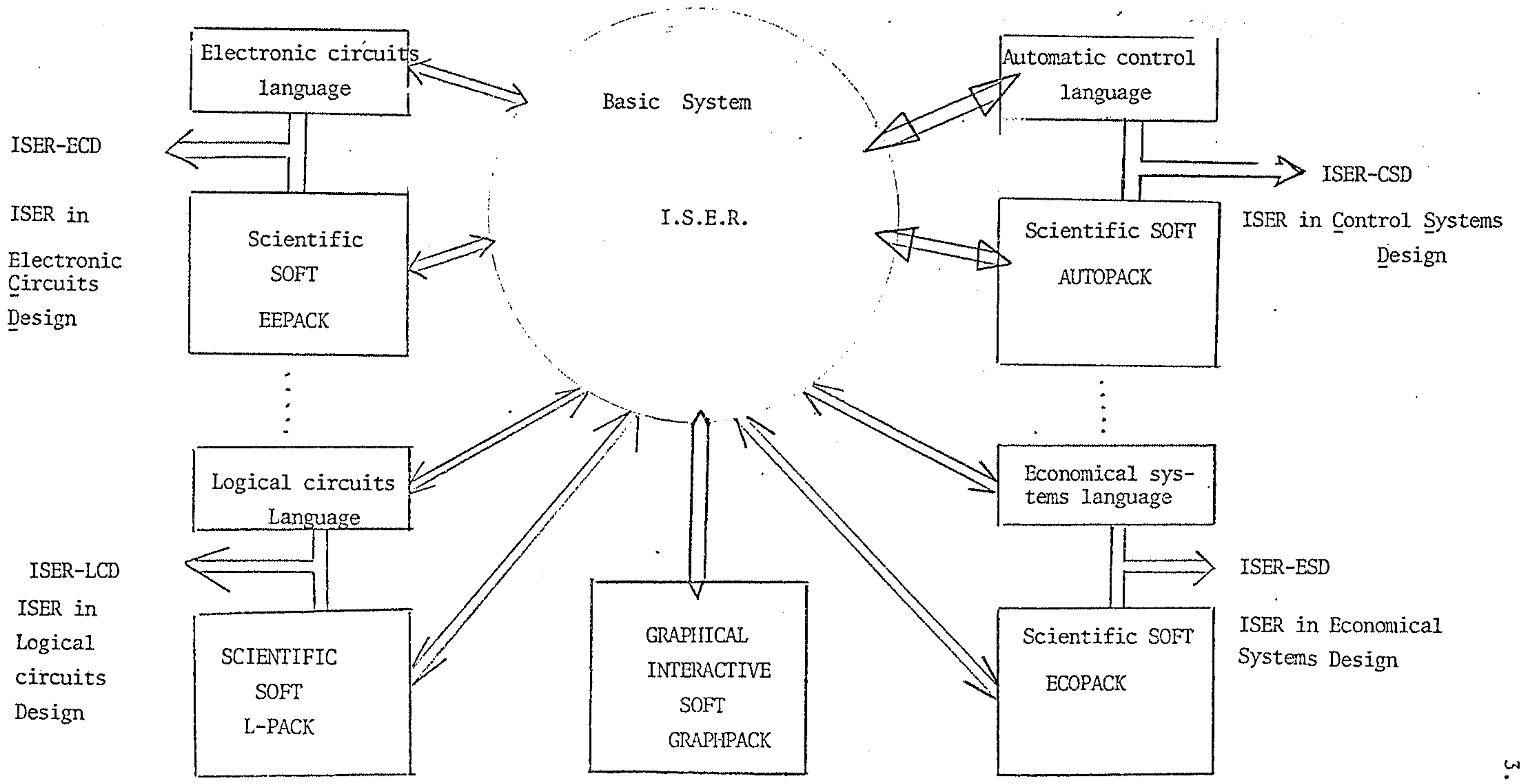
It is the objective of the project (ISER) to accomplish this goal in most engineering areas. Our immediate occupation is the particular area, automatic control theory.

The NORD-10 / S (NORSK DATA) digital mini-computer is utilized. ISER-CSD is operationnal under the Operating-System SINTRAN-III. In addition time shared graphic terminals such as Tektronix 4010 or 4014 are used.

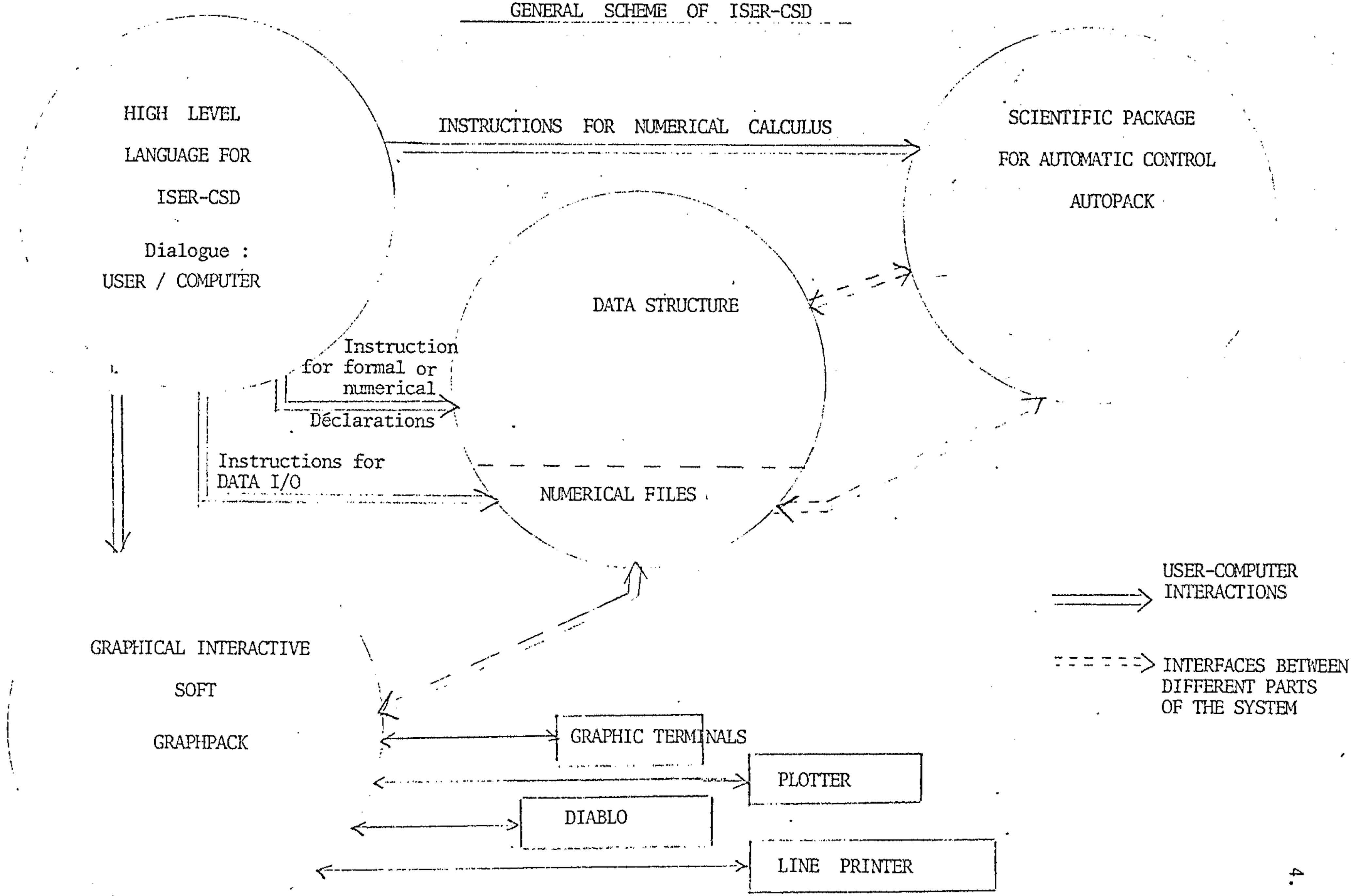
The major consequences of this project have been the computer aided design and simulation of linear continuous or discrete time simple or complex dynamical systems and design programs to identify and control simulated processes or real ones known as data input-output files.

These will be used in theoretical and laboratory courses in Automatic Control theory at the Automatic Control Laboratory in Grenoble - France.

ISER - PROJECT



GENERAL SCHEME OF ISER-CSD



* *CARACTERISTICS OF ISER-CSD.*

- Problem oriented, portable.
- User-oriented Input/output language, very easy to use (totally aided or free modes).
- No computer science knowledge is necessary.
- Supported by a DATA-STRUCTURE which memorize the problem in the same maner as patch block in analog computers with many possibilities :
  * Formal calculus and block diagram reduction.
  * Non linear simulation.
  * Systems association, deletion, modification etc...
- Performant file system for numerical or symbolic Input - Output.
- Interactive graphical soft package, performant and problem oriented, easy to use.
- Scientific soft package based on performant numerical methods in automatic control theory.
- Procedure HELP aids the users.

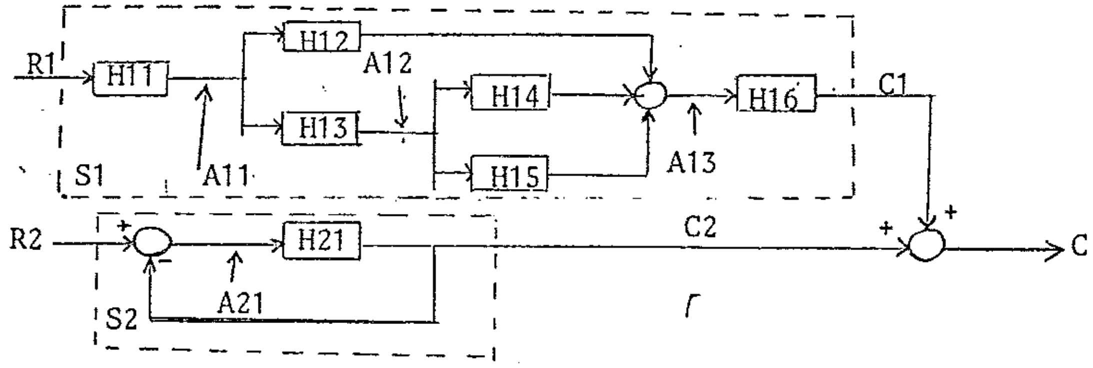
* *FEATURES OF ISER-CSD.*

- Simulation dynamical linear systems described in the transform plan (s or z) or in the state plan (t or kT).
- Simulation of non linear systems.
- Root locus, frequency and time responses.
- Graphical visualization of the system responses (frequency and time).
- Processes identification and control from performent algorithms.

are provided.

The next exemple shows the features provided by the high level language and how easy is to use the system.

EXAMPLE :



*	TYPE	LAPL	formal declaration of the system S1, Input-Output and auxiliary variables
**	SYST	S1	
S*	ENTR	R1	
S*	SORT	C1	
S*	AUXI	A11, A12, A13	
S*	C1=[H16] * A13		
S*	A13=[H12] * A11 + [H14+H15] *A12		
S*	A12=[H13] * A11		
S*	A11=[H11] * R1		
S*	END		
**	SYST	S2(R2 ; C2)	Formal declaration of the system S2
S*	TRMT	H21	
S*	AUX1	A21	
S*	C2=[H21] * A21		
S*	A21=R2-C2		
S*	END		

** MØDI S2, LAPL M* R2 = A13(S1) M* END	Modification of the system S2
** DATA S2,LAPL D* H21=1/S^2+3*S 1+2 D* END	Data for system S2
** SYST S(R=[R1(S1),R2(S2)] ; C) S* END	Association of S1 and S2 for obtain S .
** RELA S,LAPL R* C=? R* C/R1(S1)=? R* END	Formal computing for S
** TIME RESP S, LAPL RT* R1=STEP (2.4, 5) > Simulation time : 20 RT* C=? RT* VISU C > > > RT* SAVE C > Destination file : RT* END	Time response for S Computer asked questions
** END	Logout of the system

A Sequence of Programs for Control Systems Analysis and Design

J. Wieslander
Stora Kopparbergs Bergslags AB
Corporate Research Centre
Falun, Sweden

This presentation will give a brief outline of a suite of programs designed to be a tool to the control systems research student or engineer. The programs are the result of a development project 1972-1980, sponsored by STU, the Swedish Board for Technical Development. The work is reported in e.g. (1) and (2).

The Objectives

The objectives of the project were twofold.

a) Make modern, often computation intensive, methods available to researchers and engineers.

b) Make use of the possibilities offered by interactive graphics.

Note that a) implies that the programsare aimed at people with at least intuitive but not necessarily exact knowledge of the methods applied. We did not aim at persons whose interest is in trying out various aspects of the algorithms as such, nor did we aim at "automatic" analysis or design.

The importance of b) is wellknown: Interaction allows the combination of the best abilities of both computer and man. Fast computation and data retrieval is combined with intuition, experience, common sense and the ability to recognize patterns. The importance of graphics is simply: "A picture says more than 1000 words".

The Programs

The following set of programs is available with a general description in (3).

- IDPAC (4) — data analysis, spectral analysis, identification, simulation of models, analysis of models
- SYNPAC (5) — state space models, simulation, synthesis by linear quadraticand related methods, analysis
- MODPAC (6) — model transformations, model analysis
- POLPAC — MISO transfer functions, analysis and design by "polynomial methods" including classical design
- SIMNON (7) — simulation of ordinary 1 st order nonlinear differential or difference equations.

How they were implemented

The key to the implementation of the entire set of programs is the subroutine package Intrac (2). Intrac functions as a module between the interactive program and its action routines and the user at the terminal. Features of Intrac includes:

- Command oriented
  - Interaction is concise, hence easy to learn
  - Initiative stays with the user
- Macro facility
  - Local/global variables
  - Arguments
  - Control and I/O statements
  - Common sequencies, allowing –
  - Simplified dialogue

Note that the drawback of the command dialogue method, viz. its lack of quidance to the inexperienced user, is alleviated by the consequential ability to implement a macro facility. Hence it is possible to write procedures/subroutines in the "top level language" implemented by Intrac. One might say that Intrac together with the action routines of the interactive program defines a problem solving language for a specific class of topics.

Experiences

The programs mentioned above have been used in many projects, with a marked increase in engineer efficiency. Some of the programs have been sold, e.g. to STFI (Swedish Forest Products Research Laboratory) where they have established themselves as a standard tool in many investigations.

The reason for the increase in efficiency is evident when the set of data objects with their belonging operations offered by Idpac, Synpac etc is compared to those available in conventional languages such as FORTRAN, BASIC etc.

Objects	Operations
Scalars	
Matrices	Matrix operations
Signals (Time series)	Simulation
Frequency Responses	Frequency Analysis
Loci	Identification
Systems	Synthesis

The ideal solution

At the end of this project, an important question is: Is this the ideal solution? The answer is no. The interactive programs together with Intrac are implemented in FORTRAN, while the user of the programs uses the "Intrac language". Thus there is a language difference between implementation and use which cause restrictions and troubles in maintenance and extension of the programs. The ideal solution would be a modern well-structured programming language with an incremental interactive compiler. In waiting for such a tulip-rose, the ideas behind the programs presented here will serve well in many problem areas.

References

The references below are all from the Department of Automatic Control, Lund Institute of Technology, Sweden.

(1) J. Wieslander: Interaction in Computer Aided Analysis and Design of Control Systems. TFRT - 1019

(2) J. Wieslander, H. Elmqvist: INTRAC - A Communication Module for Interactive Programs - Language. TFRT - 3149

(3) J. Wieslander: Interactive Programs - General Guide. TFRT-315

(4) "- : Idpac Commands - User's Guide. TFRT - 3157

(5) "- : Synpac Commands - User's Guide. TFRT - 3159

(6) "- : Modpac Commands - User's Guide. TFRT - 3158

(7) H. Elmqvist: Simnon - An Interactive Simulation Program for Nonlinear Systems - User's Manual. TFRT - 3091

PAAS and SIMUL, conversational programs

for educational purposes in control systems analysis.

presented by

Raymond R. GOREZ (*)

PAAS is a conversational program which has been developped as an aid for systems analysis. Its main features are:

1. manipulation of transfer functions : the user can define or read a set of transfer functions, manipulate them (add or multiply them, two by two, or calculate a closed-loop transfer function $H_1 (1 + H_1 H_2)^{-1}$), investigate the behaviour of any of the previously defined subsystems (frequency response, Bode-Black or Nyquist plots, modified Nyquist plots as used with Popov stability criterion; pulse and step reponses; root locus analysis);

2. use in conversational mode : the user has only to answer Y or N, enter his data, specify scales for graphics (display or plot) or to select among various options on requests of the computer,

3. structured programming, allowing for further developments or modifications.

SIMUL is a simulation program and language for continuous-time or combined continuous- and discrete-time systems. Among the main features of the language are:

1. writing of the differential equations in the standard form of state equations,

2. partitionning of the program in 1 to 6 phases corresponding to the standard phases of an analog simulation (coefficients setting, initialization, run, terminal phase);

3. automatic sorting of the phases and of the equations inside the dynamic phase,

4. use of subscripted variable, allowing the simulation of discretized models of distributed parameter systems.

(*) University of Louvain
Lab. of Automatic Control and Systems Analysis
Bâtiment Maxwell, Place du Levant 3
B-1348 LOUVAIN LA NEUVE Belgique.

The program is used in conversational mode : after the SIMUL program has been entered by the user, it is automatically translated into a FORTRAN IV program, then compiled and linked to the standard libraries.

Afterwards, the user may be requested to specify some parameters, and the simulation run (or a sequence of runs) is started; it may be traced during execution, thanks to the display of one or two state variables on a video console, and possibly interrupted by the user. Further the complete time-histories of all the state variables are stored during the run, allowing the user to call any of them for display, print or plot, after the run.

References

1 D. LIENART & R. GOREZ "SIMUL, a New Continuous Systems Simulation Language for Mini Computers" to appear in *Proc. of the 3d Int. Symposium on Simulation*, Interlaken, 1980.