A ROBUST COMPUTATIONAL APPROACH TO CONTROL SYSTEM ANALYSIS AND DESIGN

M. J. Denham, C. J. Benson and T. W. C. Williams

School of Electronic Engineering and Computer Science, Kingston Polytechnic, Kingston-upon-Thames, Surrey, UK

Abstract. For many years the theory of control system analysis and design has been developed with little consideration for the reliable numerical computation of a solution to the analysis or design problem. The growing industrial interest in the use of recently developed multivariable control system design methods on modern, complex systems has exposed and highlighted the computational deficiencies of these methods and the need for a reformulation of the problems in a way which will directly result in efficient and numerically sound computational algorithms for their solution. Many of the results of recent research in numerical linear algebra have direct relevance to this, and this paper demonstrates how the theory underlying modern control system design methods can be represented in a way which uses these results. The implementation of the resulting algorithms in software and the design of a subroutine library for control engineering (SLICE) to contain the algorithms is also described. Some computational results are given to demonstrate the performance of the algorithms.

Keywords: computer aided design; control system analysis; control system synthesis; multivariable control systems; numerical methods; computer software.

INTRODUCTION

For many years the theory underlying control system analysis and design methods has been developed and described almost entirely from a purely mathematical viewpoint and little consideration has been given to the practical need to translate the methods into reliable computational tools to solve real design problems. The growing interest from industrial control system designers in the use of recently developed multivariable system design methods has served to further expose and highlight their computational deficiencies and several users of these methods have found serious difficulties in applying them to the kind of large, complex system model which is typical of industrial controller design problems.

It is also apparent that some designers of control systems are totally unaware of the possible deterious effects of poorly scaled input data and of numerical rounding errors involved in the computation on the resultant solutions produced, on which their design is based. They appear satisfied with the computational performance of an algorithm as long as it produces some numbers, no matter how arbitrary these might be. This is clearly an unsatisfactory state of affairs and it is one aim of this paper to bring these matters to the general attention of people working in the field of CAD of control systems. Fortunately for control system designers, there has developed over recent years a considerable body of knowledge in the field of numerical linear algebra, much of which is directly applicable to analysis and design algorithms for control systems. Indeed, some numerical analysts have recently been paying considerable attention to algorithms arising from control problems, e.g. Laub (1980), Van Dooren (1981), and several new, reliable algorithms have been produced. We aim to introduce the concepts underlying these algorithms in this paper and to explore their relevance to control system design problems. As part of a programme of software development sponsored by the U.K. Science and Engineering Research Council, these algorithms have been collected together to form a subroutine library for control engineering (SLICE). This has enabled us to explore the efficiency, accuracy and robustness of the algorithms and the preliminary results of this investigation are presented here.

We have not presented the material here in a detailed, rigorous mathematical form. For this, we suggest a study of the given references. Instead, we have attempted to present the underlying numerical concepts in a simple non-mathematical form to enhance the likelihood of communicating these ideas to the control designer. We realise the deficiencies of this approach, but regard the prime task at present as that of increasing awareness of the advantages of using "good" numerical software tools in control system design. We also recognise that many control designers wish to be isolated from the mathematical details of their software tools yet require just sufficient knowledge of their properties to instil confidence in their use. Our approach is therefore very much from the control designers viewpoint rather than that of the numerical analyst.

NUMERICAL CONCEPTS

A preliminary study of numerical methods and consultation with numerical analysts immediately reveals one important fact: at the present state of knowledge in the field, substantially more is known about the performance of algorithms involving real or complex valued scalars, in terms of the action of rounding errors, etc., than about the corresponding performance of algorithms involving polynomials. The implication from this is that, at least at present, concentration on the use of system models which involve the manipulation of real matrices, etc. is more likely to yield numerically robust algorithms than is the case for models involving polynomial manipulation. Hence our concentration here is on the use of state-space models, although impulse response (weighting function) models would also fall into this category.

The major computational procedures which must be performed in analysis and design algorithms using such models are:

(i) rank determination

(ii) eigenstructure determination

(iii) solution of sets of simultaneous linear algebraic equations.

Fortunately for control designers, there is now a good understanding of these procedures in terms of the basic numerical properties of conditioning, stability, and reliability. The properties are summarised below.

Conditioning: a problem is "well-conditioned" if small changes in the data cause only small changes in the solution.

Stability: an algorithm is "stable" if the result obtained is "close" to the exact result obtained for a "nearby" problem.

Reliability: an algorithm is "reliable" if it provides full warning of results containing large errors.

Considering each of the computational procedures, we can make the following remarks.

Rank determination: due to the presence of rounding error, this is a non-trivial computational problem. Golub and co-workers (1976) have defined numerical rank in such a way as to show that if $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ are the non-zero singular values of a matrix A and ε is chosen so that $\sigma_k > \varepsilon > \sigma_{k+1}$, then all matrices B inside an ε - neighbourhood of A, i.e. $\|A - B\| \leq \varepsilon$, have rank $\geq$ k. Computationall the determination of the singular values of a matrix is expensive and similarly effective results can be obtained using the QR decomposition with column pivoting. In fact the QR decomposition can be used to form the basis of many algorithms for control problems, as will be demonstrated in later sections of this paper.

Stable algorithms exist for the solution of both the singular value decomposition (Golub and Reinsch, 1970) and for the QR decomposition and reliable implementations of these algorithms are available e.g. in LINPACK (Dongarra and others, 1979). Moreover the problem of computing the singular value or QR decompositions is a well-conditioned problem.

Eigenstructure determination: a major initial step in control design has always been to compute the poles of the system model, e.g. by determining the eigenvalues of the A matrix in the state space model, $\dot{x} = Ax + Bu$. More recently the significance of the eigenvectors in disturbance rejection and robustness has been recognised and methods for computing the full eigenvalue/eigenvector structure are of strong interest For the general case, a stable algorithm for computing this is available as the so-called double Francis QR algorithm, available as a reliable implementation, e.g. in EISPACK (Smith and others, 1976). However, it has been well known for some time that the eigenstructure problem is potentially ill-conditioned, in the sense that it is easy to construct ill-conditioned matrices for this problem. It is therefore important to the control designer to know if his problem is in such a category, for example, in case small changes in the system model within the region of modelling error could make a difference equivalent to the system model changing from stable to unstable or vice versa. It is rare however for the control designer to check the condition of his problem, even though this can be readily done by computing the condition number of the matrix (Wilkinson, 1965).

Solution of sets of simultaneous linear algebraic equations: this problem is clearly related to the previous two and again has been studied extensively. Many reliable algorithms exist, notably the LINPACK set of routines. Similar comments regarding ill-conditioning of the problem apply, in this case ill-conditioning with respect to inversion. An efficient condition number estimator is implemented in

LINPACK (Cline and others, 1979) Scaling of the input data can also affect the accuracy of the result. In the eigenstructure case (Parlett and Reinsch, 1969), scaling is simply a matter of reducing the norm of the A matrix by an exactly computed diagonal similarity transformation, which is equivalent to scaling the states of the system model. In problems involving the complete state equations, e.g. computation of controllability, zeros, frequency response, etc., as we shall see later, it is also necessary to apply scaling operations to the input and output variables of the system model. However, the best form of scaling to apply is still a topic for investigation.

In the above, we have emphasised the need for both stable algorithms and for assessing and if possible improving the conditioning of the problem in order to detect and reduce possible excessive errors. Practical modelling of industrial systems often result in poorly scaled data, as exemplified by the widely studied gas turbine engine model in Sain and others (1978), and action must be taken to improve this if accurate results are expected. Obtaining a numerically stable algorithm can be guaranteed if only unitary (orthogonal) transformations are used in computing the result (Wilkinson, 1963). Therefore we shall only consider those algorithms employing unitary transformations. Two examples of such algorithms are those quoted above for the QR and singular value decompositions.

COMPUTING BASIC CONTROL SYSTEM PROPERTIES

Subsequent to physical modelling of a system and both prior and subsequent to design of a controller for the system, it is customary to investigate various properties of the system model. Complexity of the model is one such property, in particular the dimension of the state, since for some proposed control synthesis methods, the complexity (dimension) of the controller is linked closely to that of the system model, e.g. observers, linear optimal regulators. An initial step in reducing complexity is to eliminate those states in the model which contribute little or not at all to the input-output performance of the model, namely the uncontrollable and unobservable (or closely so)states. It is clear that if we do not eliminate closely uncontrollable states then a controller designed to regulate these states will need to employ large control action. Similarly, closely unobservable states will require high gains in an observer for fast convergence to an accurate observation.

A full discussion of the "classical" theoretical approaches to computing controllability and their relative degrees of numerical deficiency is given in Paige (1981), together with a numerically stable algorithm for computing this property. A similar algorithm is implicit in the work of Konstantinov and his co-workers (1981). Perhaps the fullest treatment of this problem and other associated geometic properties of linear multivariable systems is given in Van Dooren (1981) where again the same basic algorithm is presented. Since this algorithm forms the basis of many control computions we will reproduce it here, using the notation of Van Dooren (1981).

Controllability algorithm

Let $[\lambda I - A\ B]$ be the $n \times (n+m)$ singular pencil corresponding to the usual state space model: $x = Ax + Bu$. The algorithm works directly on the composite matrix $[A\ B]$ using unitary transformations.

step 0: initialize $A_c = A$, $B_o = B$, $j = 1$,

$c = 0$, $T = I_n$

step j: compute a unitary transformation

U_j to compress the rows of B_{j-1},

i.e. $U_j^* B_{j-1} = \begin{bmatrix} 0 \\ Z_j \end{bmatrix}$, Z_j has full row rank.

Let ρ_j be the row dimension of Z_j

$\tau_j = n - c - \rho_j$

If $\rho_j = 0$, then go to exit 1.

If $\tau_j = 0$, then go to exit 2.

Apply U_j as a similarity transformation to A_{j-1} and partition as follows:

$$U_j^* A_{j-1} U_j = \begin{bmatrix} A_j & B_j \\ X_j & Y_j \end{bmatrix}$$

where τ_j, ρ_j are the row dimensions of A_j, X_j .

Update

$$T = T \begin{bmatrix} U_j & 0 \\ 0 & I_c \end{bmatrix}, \quad c = c+\rho_j, \quad j = j+1$$

Go to step j.

exit 1: the system model is uncontrollable with dimension of controllable part $c = n - \tau_{j-1}$

exit 2: the system model is controllable

The similarity transformation T reduces the pencil $[\lambda I - A\ B]$ in the case of an uncontrollable model to the form

$$\begin{bmatrix} \lambda I_{\bar{c}} - A_{\bar{c}} & 0 & 0 \\ * & \lambda I_c - A_c & B_c \end{bmatrix}$$

from which the uncontrollable part can be easily distinguished. As remarked in Van Dooren (1981), the above algorithm is numerically stable and thus the form of the pencil above is exact for a perturbed pencil $[\lambda I - \bar{A}\ \bar{B}]$ where

$$\| A - \bar{A} \| < \varepsilon_A , \| B - \bar{B} \| < \varepsilon_B$$

and ε_A, ε_B are small values approximately equal to the machine precision times the norm of the respective matrix (A or B) times a low order polynomial in n. Experience with this algorithm has shown that n^2 appears to work well.

It is evident that the major step in this algorithm is the reduction of B_{j-1} to a full row rank matrix Z_j and remaining rows zero. This numerical rank determination step can be performed either by singular value decomposition or QR decomposition of B_{j-1}, the latter being computational faster and only less reliable in pathological cases of the input data.

This algorithm can be used in a number of analysis algorithms which explore the geometric state space structural properties of the system model, e.g. presence of finite and infinite zeros, (A,B) - invariant subspaces, controllability subspaces, etc. These applications are described in Van Dooren (1981). The link to a polynomial matrix formulation of these ideas has also been explored (Williams, 1982) using an "orthogonalised" version of Wolovich's structure theorem (Wolovich, 1974), and has resulted in efficient, reliable algorithms for translating system models between state space and polynomial matrix fraction form.

USE IN CONTROL SYSTEM DESIGN

We are only just beginning to explore the ways in which the new classes of control analysis algorithms based on sound numerical methods can be employed in control system design. Recent examples of the use of stable methods in controller synthesis include algorithms for pole assignment in the scalar input case (Miminis and Paige, 1981; Pettkov, 1981), for the linear quadratic regulator (Laub, 1979; Van Dooren 1981).

Of great interest also is an understanding of the effect in the design solution of making certain numerical decisions in the operation of the algorithms involved, i.e. as mentioned above, the need exists to decide whether certain matrices which are generally of full rank are close to a matrix of lower rank. Such a decision will affect the form of the controller which results and its performance in the closed loop system. It is not possible due to limitations on space to give in this paper an account of the mechanism involved which produces this effect and work is still in progress to fully investigate this. We will give however a small and incomplete example of the way in which the choice of numerical parameters in a linear quadratic regulator design affects the resulting controller.

Our example is the F100 gas turbine engine model of Sain and others (1978). This is, in its simpler form, a 16 state, 5 input, 5 output system, of the form

$$x = Ax + Bu$$
$$y = Cx + Du$$

The data for the matrices A, B, C, D is extremely badly scaled, with values in A and B varying between 0.8898E+05 and 0.6654E-04. Application of the basic controllability algorithm given above, with a zero tolerance based on the machine precision and with no scaling, yields a controllable subspace of dimension equal to 15. After balancing of A and corresponding scaling of the columns of B, using exact arithmetic based on the machine radix, the controllable subspace dimension is recovered as 16, the full state dimension as modelled. The machine used for this and all computations described here is a VAX 11/780 on which the machine precision is 0.6E-07. To illustrate the effect of changes in machine precision on the computed dimension of the controllable subspace, the table below lists the value of zero tolerance used against the computed dimension and the corresponding controllability indices. These values are for a scaled version of the F100 data.

Zero tolerance	Controllable subspace dimension	Controllability indices	Norm F
0.60E-07	16	5,5,5,1	0.2126
0.10E-05	16	5,5,5,1	0.2126
0.60E-05	16	5,5,4,1,1	0.2099
0.30E-04	15	5,5,4,1	0.1839
0.60E-04	15	5,5,4,1	0.1839
0.10E-03	15	5,5,4,1	0.1839
0.15E-03	15	5,5,3,1,1	0.1879
0.21E-03	14	5,4,3,1,1	1072.0

It is clear from this that with normal machine precision, the generic case holds. An increase in the zero tolerance to 0.60E-05, which is still small in relation to the uncertainty in the system data, gives a non-generic result. A further increase to 0.30E-04 yields a controllable subspace dimension of 15.

The final column above gives a measure of the size of the feedback matrix F obtained from solving the linear quadratic regulator problem for the controllable part of the system model as determined by the controllability algorithm using the given zero tolerance. This shows a gradual decrease of up to 9% in the norm of F, followed by a large increase consequent on the loss of one more state in the model. This is believed to be due to the fact that the "weakly controllable" part of the system, which is not being taken account of in the computation of the optimal regulator can nevertheless affect the rest of the system, as is clear if one considers the block triangular structure of the A matrix obtained by the controllability algorithm. Whilst strong influence on this part of the system by the control is not possible without large feedback gains, it appears that their uncontrolled behaviour influences the system in such a strong manner as to require much larger gains in the controllable part to give the required optimal control than would be the case if the "weakly controllable" states were under some control.

The above explanation is largely intuitive but clearly illustrates the difficulty in interpreting the influence of the numerical precision used in the computational algorithms on the resulting design. Evidently more work is required to produce a rigorous, systematic approach to design which takes full account both of numerical error and data inaccuracies in the design algorithms.

SUBROUTINE LIBRARY IN CONTROL ENGINEERING

The need for reliable, stable computational algorithms in control system design packages has caused the U.K. Science Research Council to sponsor the development of a subroutine library in control engineering (SLICE), which is a collection of FORTRAN subroutines implementing the most recent developments in algorithm design by researchers worldwide. There is also a related research programme in progress to develop, test and analyse the performance of new and existing algorithms and to examine their role and performance in the control system design process.

All the algorithms in the SLICE library (currently 35) use exclusively stable orthogonal transformations to compute the required results and full information is provided wherever possible on the condition of the data being used and on reasons for failure of the algorithm should this occur. As far as possible, a "soft" failure is ensured, whereby control is returned to the calling program in an orderly fashion. The algorithms are documented and coded to defined standards (Denham and Benson, 1981). A subset of ANSI FORTRAN has been used to code the routines, corresponding to PFORT (Ryder, 1975) and each routine has been successfully passed through the PFORT verifier.

CONCLUSIONS

The need for a more rigorous approach to numerical computation in control system design was first described by the author in earlier papers (Denham, 1978, 1979) and to a large extent this present paper represents a progress report on our activities to try to achieve the aims set out there. With a strong and growing interest from numerical analysts in the field of algorithms for control design, we believe that the aim of having a comprehensive set of reliable algorithms is well on its way to being achieved. Much work remains however, and new algorithms for solving both the traditional and new problems in control design will continue to appear well into the future now that the initial stimulus has been applied.

The future requirements for effective tools for CAD of control systems is also a matter of concern. Principal amongst these is the need for a powerful design environment (Denham, 1982) of the kind now becoming familiar in the design and engineering of software, e.g. APSE, UNIX. Work on such an environment is shortly to begin, based on the Berkeley DELIGHT system (Polak and Mayne, 1982) which goes much of the way to fulfilling the requirements for such a system, e.g. high level of interaction, an incrementally compiled algorithmic design language, powerful editing and graphical input/output capabilities, ability to support any control design methodology. Together with the use of reliable computational tools such as those in the SLICE library, this will provide the kind of design tools which will be essential for control engineers in the future.

REFERENCES

Cline, A.K., Moler, C.B., Stewart, G.W., and J.H. Wilkinson (1979). An estimate for the condition number of a matrix. SIAM Journal on Numerical Analysis, 16, 368-375

Cline, A.K., Conn, A.R., and C. Van Loan (1981). Generalizing the LINPACK condition estimator. Report TR81-462, Computer Science Department, Cornell University.

Denham, M.J. (1978). A program package for computer aided design of control systems. Proceedings of the 7th IFAC World Congress, Helsinki.

Denham, M.J. (1979). Structural design of large systems using a geometric approach. Proceedings of the 1st IFAC Symposium on Computer Aided Design of Control Systems, Zurich

Denham, M.J., and C.J. Benson (1981). Implementation and documentation standards for the software library in control engineering (SLICE). Internal report, School of Electronic Engineering and Computer Science, Kingston Polytechnic.

Denham, M.J. (1982). Design of a software library and an interactive design environment for computer aided control system design. IEEE Workshop on Computer Aided Control System Design, University of California, Berkeley, (to appear in IEEE Control Systems Society Magazine, December 1982).

Dongarra, J., Moler, C.B., Bunch, J.R., and G.W. Stewart (1979) LINPACK Users' Guide, SIAM, Philadelphia.

Golub, G.H. and C. Reinsch (1970) Singular value decomposition and least squares solutions. Numerische Mathematik, 14, 403-420.

Golub, G.H., Klema, V.C., and G.W. Stewart (1976). Rank degeneracy and least squares problems. Report STAN-CS-76-559, Computer Science Department, Stanford University.

Konstantinov, M.M., Petkov, P.Hr., and N.D. Christov (1981) Invariants and canonical forms for linear multivariable systems under the action of orthogonal transformation groups. Kybernetika, 17, 413-424.

Laub, A.J. (1980). Survey of computational methods in control theory. Proceedings of the SIAM Conference on Electric Power Problems, Seattle.

Miminis, G.S. and C.C. Paige (1981). An algorithm for pole assignment of time invariant linear systems. Internal report, School of Computer Science, McGill University.

Paige, C.C. (1981). Properties of numerical algorithms related to computing controllability. IEEE Transactions on Automatic Control, AC-26, 130-138.

Parlett, B.N. and C. Reinsch (1969). Balancing a matrix for calculation of eigenvalues and eigenvectors. Numerische Mathematik, 13, 293-304

Petkov, P. Hr. (1981). A computational algorithm for pole assignment of linear single input systems. Internal report, School of Electronic Engineering and Computer Science, Kingston Polytechnic.

Polak, E. and D.Q. Mayne. DELIGHT-MIMO: An interactive system for optimization based design of multivariable control systems. IEEE Workshop on Computer Aided Control System Design, University of California, Berkeley. (to appear in IEEE Control Systems Society Magazine, December 1982).

Ryder, B.G. (1974). The PFORT verifier. Software Practices and Experience, 4, 359-377.

Sain, M.K., Peczkowski, J.L., and J.L. Melsa, (1978). Alternatives for Linear Multivariable Control. National Engineering Consortium, Chicago.

Smith, B.T., and others (1976). Matrix eigensystem routines - EISPACK guide. Lecture Notes in Computer Science, 51, Springer-Verlag, New York.

Van Dooren, P. (1981) The generalized eigenstructure problem in linear system theory. IEEE Transactions on Automatic Control, AC-26, 111-129.

Wilkinson, J.H. (1963). Rounding errors in algebraic processes. Prentice-Hall, New Jersey.

Wilkinson, J.H. (1965). The Algebraic Eigenvalue Problem Oxford University Press, London.

Williams, T.W.C. (1982). The orthogonal structure theorem. Internal report, School of Electronic Engineering and Computer Science, Kingston Polytechnic.

Wolovich, W.A. (1974). Linear Multivariable Systems. Springer-Verlag, New York.