what a long way we really have to go before such claims have more than a shred of reality. We urge our readers simply to maintain a healthy skepticism for anything and everything in this young and rapidly evolving field. We must all learn to become intelligent and discriminating users of software. And some of us will also learn how to set standards whereby future CACSD packages will be developed which can truly be used easily, efficiently, and confidently by a "non-expert". To that end, we firmly believe that the only thing that will elevate us from an era of piecemeal, patchwork, sometimes half baked attempts will be an opportunity for long term funding at an appropriate level. A disciplined, broadly based, well coordinated team can produce tools for CACSD of lasting quality. But exceptionally high quality software is exceptionally expensive—at least in the short run—in both time and money. However, an economic argument for a major CACSD commitment can certainly be made, and we hope that a major effort can be undertaken which will involve both coordinated funding and coordinated research.

CLADP: The Cambridge Linear Analysis and Design Programs

J. M. Maciejowski and A. G. J. MacFarlane

Control and Management Systems Division, Cambridge University Engineering Department, Mill Lane, Cambridge CB2 1RX, England

Abstract

An interactive software facility for designing multivariable control systems is described. The paper discusses the desirable characteristics of such a facility, the particular capabilities of CLADP and the numerical algorithms which lie behind them, and the probable course of future development.

1. Introduction

The problem of creating a feedback controller for a plant described in terms of a given dynamical model has three aspects, conventionally called *analysis*, *synthesis* and *design*. In developing a synthesis technique, the aim is to formulate a desired objective as a sharply-defined mathematical problem having a well-founded solution which is expressible in terms of a workable, efficient and robust computer algorithm. In principle then, one loads the synthesis problem description into the computer and the answer duly emerges. The disadvantages of a purely synthetic approach to design are obvious in an engineering context since the role of the designer, particularly the exercise of his intuitive judgment and skill, is severely reduced. An even greater drawback is that, at the beginning of his investigations, the designer simply may be unable to specify what he wants because he lacks information on what he will have to pay, in engineering terms, for the various aspects of desired final system performance.

In developing a design technique, one seeks to give a practicing and experienced design engineer a set of manipulative and interpretative tools which will enable him to build up, modify and assess a design put together on the basis of the physical reasoning within the guidelines laid down by his engineering experience. Thus, design inevitably involves both analysis and synthesis and hence, in the development of design techniques, consideration of the way in which a designer interacts with the computer is vitally important. It is imperative to share the burden of work between computer and designer in such a way that each makes an appropriate contribution to the overall solution.

In developing the Cambridge Linear Analysis and Design Programs (CLADP) the aims have been to:

(i) allow the designer to fully deploy his intuition, skill and experience while still making an effective use of powerful theoretical tools; and

(ii) to harness the manipulative power of the computer to minimize the level of detail with which the designer has to contend.

The designer communicates with the computer through an *interface*. This allows him to *interpret* what the computer has done and to *specify* what he wishes it to do next. In general terms we will call anything which is presented to the designer by the computer, and which is relevant to the design process, an *indicator*. The designer must operate within an appropriate *conceptual framework*, and any powerful interactive design package must present the designer with the full set of indicators required to specify his needs and interpret his results in the context of his conceptual framework.

The computer is used for calculation, manipulation and optimization. In any fully-developed interactive design package the "tuning" of controller parameters is best done by a systematic use of appropriate optimization techniques. Generally speaking, in the design process the designer will be doing analysis and the computer will be doing synthesis. That is to say, the computer will be used to solve a series of changing and restrictively-specified synthesis problems put to it by the designer as he works his way through a range of alternatives, among which he chooses on the grounds of engineering judgment, as he travels towards his final design.

Since the designer will usually want to think in the most physical way possible about the complex issues facing him, a high premium is placed on developing a conceptual framework which makes the maximum use of his *spatial* intuition, and which is formulated as much as possible in geometric and topological terms. For this reason, heavy emphasis is placed in CLADP on generalized frequency-response methods. Generalized Nyquist diagrams and multivariable root-locus diagrams are used as indicators of stability. These are derived from frequency-dependent characteristic decompositions

0272-1708/82/1200-0003$00.75 © 1982 IEEE

of transfer function matrices. While such a decomposition gives accurate stability information, singular-value decompositions (or Nyquist or Bode arrays) are needed for an accurate assessment of performance and robustness.

Bode plots of principal gains (derived from frequency-dependent singular value decompositions) are used as indicators for performance and in investigations of robustness. These indicators enable a natural extension of the classical gain/phase approach to feedback system design to be made to the multivariable case. For complex plants they can be used to derive a *realistic* closed-loop specification, which can then be achieved using appropriate parameter optimization techniques.

2. Summary of CLADP Capabilities

Analysis Facilities

Some analysis is required both before and after attempting the design of a multivariable feedback system. Before starting the design, it is necessary to see whether the system is stable, and, if not, how many unstable poles it has. It is useful to check for any right-half plane zeros, which will impose an upper bound on the attainable closed-loop bandwidth. The Nyquist array, in Bode form, will disclose whether there are any sharp resonances which can be expected to give trouble, and whether the system transmission paths are strongly cross-coupled; it will also reveal the open-loop bandwidth of each transmision path. CLADP provides facilities for performing this analysis easily, as well as the post-design analysis described below.

After proposing a feedback design, the most important property to be checked is closed-loop stability. This is most conveniently done by computing and displaying the characteristic loci of the compensated system, and applying the Generalized Nyquist theorem [1]. It is also possible to display a root-locus diagram, but this is usually much less useful for multivariable systems than it is for single-loop systems, since it gives virtually no guidance on how the design may be improved (for example, to attain greater stability margins).

It is usually simple enough to achieve closed-loop stability. To achieve acceptable closed-loop performance as well is much more difficult. For single-loop systems the Nyquist locus, in any of its usual forms, gives reliable information about stability, performance, and robustness of the design in the face of large parameter variations. However, in the case of multivariable systems, the characteristic loci carry all this information only for the special case of so-called "normal" systems, i.e. those for which the return-ratio matrix Q(s) satisfies

$$Q(s)Q^*(s) = Q^*(s)Q(s).$$

(Where * denotes "complex-conjugate transposed"). In general, the characteristic loci can give misleading information about performance and robustness, and it is necessary to compute and display other indicators in order to assess these.

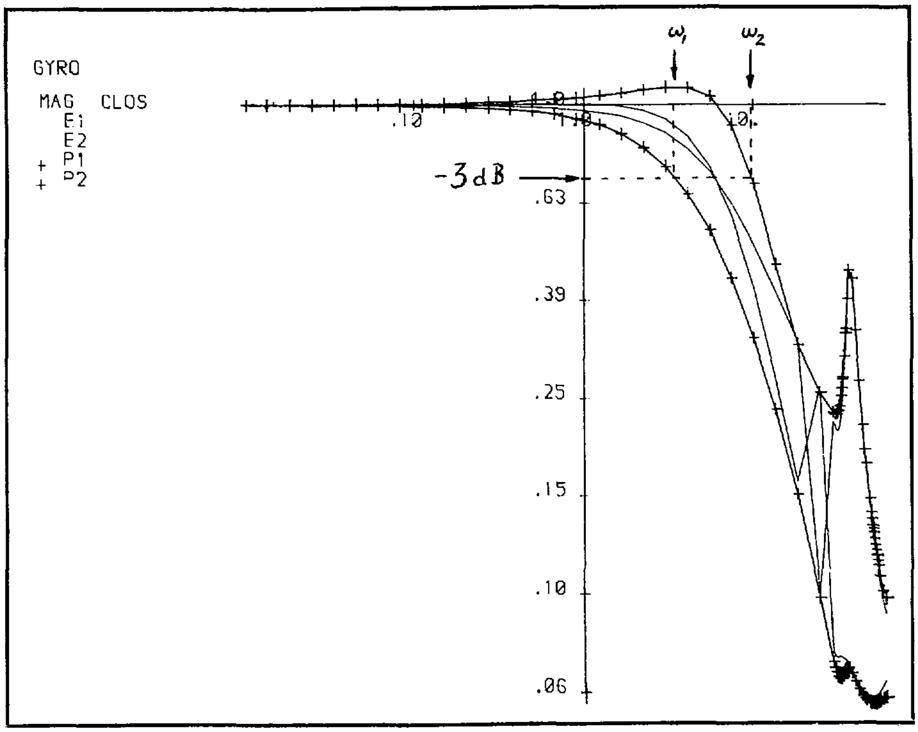


Fig. 1. Closed-loop characteristic and principal gains.

CLADP allows the designer to display the "principal gains" of the open-loop return-ratio $Q(s)$ (i.e., the singular values of $Q(j\omega)$, evaluated over a range of frequencies) [2] [3], as well as the principal gains of the closed-loop transfer function $(I+Q)^{-1}Q$, and of the sensitivity function $(I+Q)^{-1}$. These displays allow aspects of performance, such as "velocity constant", "output-disturbance rejection bandwidth", etc., to be accurately quantified. For example, Fig. 1 shows the two principal gains, as well as the gains of the characteristic loci, of $(I+Q)^{-1}Q$ for a 2-input, 2-output system. By adopting an arbitrary definition of bandwidth, namely the "-3dB frequency", we can define the "tracking bandwidth", ω_1, up to which tracking of reference signals can be guaranteed to be good, and the "noise-transmission bandwidth", ω_2, beyond which transmission of sensor noise can be guaranteed to be small. Note that the smallness of $\omega_2 - \omega_1$ gives one measure of the efficiency of the design, which would be greatly overestimated by looking at the characteristic gains alone. Principal gains are also useful for assessing robustness [3] [4].

A complementary means of assessing performance and robustness is to display the Nyquist arrays of $(I+Q)^{-1}Q$ and of $(I+Q)^{-1}$ [5]. In Bode magnitude form these give performance assessments which are particularly useful if the designer is faced with "structured uncertainty"—for example, if he knows that disturbances acting on certain outputs are "high frequency" while those acting on others are "low frequency". Interaction in the closed-loop design, as well as robustness in the face of partial or complete loop failures, can also be assessed from suitable Nyquist arrays.

The return-ratio of a multivariable system depends on the point of the loop at which it is calculated. Thus, if the plant transfer function is $G(s)$ and the compensator transfer function is $K(s)$, the return-ratios $Q_1(s) = G(s)K(s)$ and $Q_2(s) = K(s)G(s)$ are not the same. Information about robustness in the face of sensor failures is carried in $Q_1(s)$, whereas if actuator failures are of concern, then $Q_2(s)$ must be looked at. For this reason CLADP allows all the computations men-

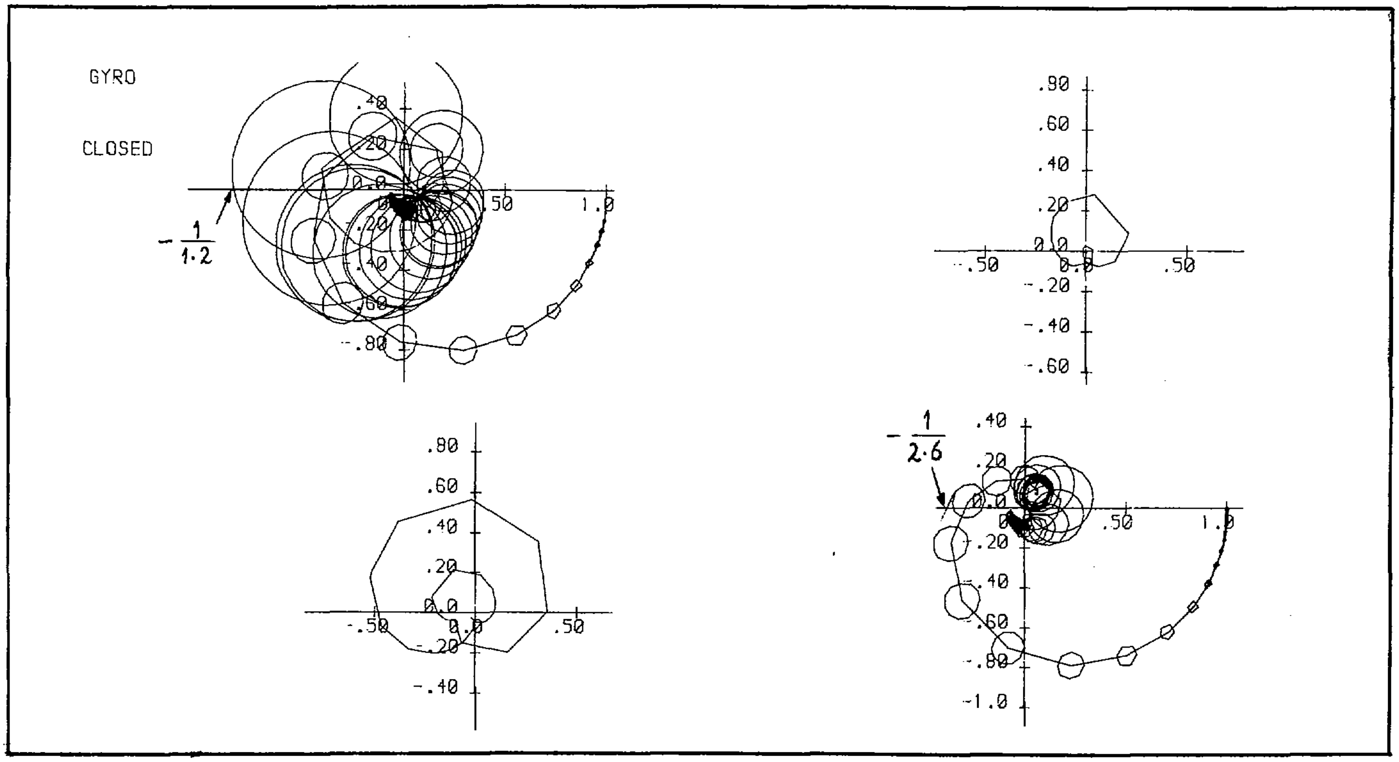


Fig. 2. Closed-loop Nyquist array at actuators, with generalized Gershgorin circles.

tioned above to be performed for any point of the loop. Fig. 2 shows the Nyquist array of $(I+Q_2)^{-1}Q_2$ for a 2-input, 2-output design, with "generalized Gershgorin circles" [6] superimposed. From these it can be deduced that the designed closed loop will be stable for the gain of the actuator on input 1 lying anywhere between 0 and 2.2 (nominal value = 1), and the gain of the actuator on input 2 simultaneously lying anywhere between 0 and 3.6.

In addition to these frequency-domain tools for analysis, CLADP provides some basic simulation facilities for checking step responses, etc.

Synthesis Facilities

The synthesis facilities available in CLADP range from very simple manipulative aids to quite sophisticated algorithms. At the lowest level, there are facilities for entering and modifying compensators from the keyboard, and for arranging these compensators into simple networks around the plant. These facilities alone are of significant assistance when using design techniques such as Inverse Nyquist Array [7] or Sequential Return Difference [8], which put almost all the load of synthesis onto the designer.

At the next level of sophistication comes the 'ALIGN' algorithm, which computes a real approximation to the inverse of a complex matrix. This is useful for decoupling the forward path in the region of the desired cross-over frequency, and for achieving diagonal dominance if one is using the INA design technique. It is also an essential tool if one is designing an Approximate Commutative Controller [9], which is a technique for manipulating characteristic loci. Here the designer is still left with the problem of choosing compensators for each characteristic locus (which he does using classical frequency-domain methods), but the tedious and complicated task of computing real approximations to complex eigenvector frames is taken over entirely by the program.

Much closer to a complete synthesis facility is an algorithm for tuning compensator parameters [5]. This requires the designer to specify the closed-loop transfer function matrix which he would like to achieve. The designer must also specify the structure of the compensator, namely the poles of each element, the order of the numerator of each element, and whether any elements are constrained to be zero. The numerator coefficients are tuned by obtaining a least-squares fit to the desired closed-loop transfer function at a number (typically 50) of specified frequencies. This algorithm is very flexible, particularly when combined with some of the other facilities of CLADP. For example, by augmenting the plant model suitably, one can impose constraints on plant input variations. On the other hand, the algorithm is not very robust if the designer demands too much: he must use the analysis facilities carefully to deduce achievable closed-loop performance—in some cases this will involve him in performing a preliminary design using some other technique.

A key feature of CLADP is a very powerful matrix manipulation facility. More will be said about this later, but here we note that this facility can be used to perform steady-state LQG design. Plant models can be augmented with disturbance dynamics, and advanced procedures such as the 'asymptotic recovery' advocated by Doyle and Stein [3] can be implemented very easily. Controllers designed in this manner tend to have a high dynamic order. This is often not a problem nowadays, in view of the advanced technology available for implementing controllers. But if it is a problem, a very effective order reduction algorithm, based on the theory of balanced realizations [10], is available.

'Utilities'

A control system designer spends much of his time performing routine tasks such as factorizing polynomials, inverting matrices, combining connected sets of system equations into single system descriptions, converting state-space descriptions to transfer-function descriptions and vice-versa, and so on. Facilities for all these and other similar tasks are provided by CLADP.

The most powerful of these 'utilities' is the matrix manipulation facility, which allows the manipulation of algebraic and other expressions involving matrix names. Its capabilities are best demonstrated by an example. Suppose the steady-state Kalman filter gain is to be calculated for a system with disturbance covariance Q and measurement noise covariance R, and the system equations are

$$\dot{x} = Ax + \omega$$

$$y = Cx + v.$$

One way of computing the Kalman filter involves finding the eigenvalues and eigenvectors of the matrix:

$$\begin{bmatrix} -A^T & C^T R^{-1} C \\ Q & A \end{bmatrix}.$$

In CLADP this can be done by the simple statement:

W = EIG((−TR(A)! (TR(C)*

· INV(R)*C))'(Q!A), VAL).

Here operators ! and ' are used to assemble partitioned matrices, so that $X!Y = [X\ Y]$, and $X'Y = \begin{bmatrix} X \\ Y \end{bmatrix}$. TR(·) is the transposition operator, while INV(·) denotes inversion. The eigenvalues of the assembled matrix are stored in the (complex) vector VAL, and its eigenvectors are stored in the (complex) matrix W.

Sequences of statements of this kind can be run from "batch files" which provide a kind of macro facility, with some conditional control of flow. Indeed, batch files containing *any* CLADP statements can be run non-interactively. This is useful, for example, for repeating the same computations for a number of different models.

Another vital utility is the provision of appropriate prompts for the user who is not sure of the options which are available to him at any point. For such a "help" facility to be effective, a balance has to be struck between restricting the user's options excessively, and swamping him with so many options that he becomes confused. The latter possibility poses a serious problem, since most users are either beginners or occasional users. In CLADP there is a tendency to swamp rather than restrict the user, but this is currently being ameliorated by the provision of hierarchically organized "menus" of options, so that the designer can choose to examine only the "display manipulation" options, or only the "computation" options, and so on.

Discrete-Time

The analysis and design of discrete-time systems in the form of either state-space descriptions or z-transform transfer function matrices is fully supported in CLADP. All facilities exist in parallel for continuous and discrete-time sytems. There are also utilities for converting from one form to the other.

Since CLADP has been developed in a university research group, it contains some facilities which aid theoretical investigations, but are not directly useful in the design process—or at least, no direct use has yet been found for them. An example of these is the capability of displaying individual sheets of the Riemann surface which is the domain of a characteristic gain or frequency function.

Of more practical use, but still unproven in design, are facilities for analyzing systems described by irrational transfer function matrices.

3. Algorithms

CLADP generally uses reliable numerical algorithms, although it does not make use of some of the latest advances, since it has been under development for the last six years. Some of the algorithms are outlined below.

The key decision which was taken in CLADP was to avoid analytical evaluation of the resolvent matrix $(sI-A)^{-1}$, but to use pointwise evaluation instead. Considerable use is also made of curve fitting. This approach has resulted in the ability of CLADP to handle the complex models which usually arise in real design studies. Successful designs have been performed for a 40-state, 2-input, 3-output continuous-time model, and for a 17-state, 5-input, 5-output discrete-time model, both of these models being given to CLADP in state-space form. The pointwise evaluation of the frequency response allows the user to spot immediately any dubious results, since these are usually revealed by discontinuities in the displayed loci. The user can also change the set of frequencies at which the evaluation is performed, and can therefore make these frequencies more dense in the region of a resonance or other important feature, and less dense elsewhere.

The option is provided of transforming the 'A' matrix of a state-space model to either Hessenberg or tridiagonal form before computing the frequency response, which can save substantial amounts of computing time if the model has many states. Computation from the tridiagonal form is faster than from the Hessenberg form, but this is offset by the possible numerical instability of the transformation to tridiagonal form.

Eigenvalue-eigenvector computations are performed frequently in CLADP, particularly for finding the characteristic loci. These are again evaluated pointwise, and then sorted so that a continuous set of loci is displayed. The algorithm used for finding eigenvalues (and eigenvectors when required) is that used in EISPACK, namely reduction to upper Hessenberg form, followed by a 'modified LR' algorithm [11].

Principal gains are the singular values of the frequency response matrix, also evaluated pointwise. The Golub and Reinsch algorithm for finding the singular value decomposition is used [11].

The transmission zeros of a model given in state-space form are the eigenvalues of the matrix $(A\text{-}BD^{-1}C)$. However, the matrix D is often singular, and even if it is regular, computation by this route can be numerically unstable. Nevertheless, if the model is square (i.e., D is square), then a bilinear transformation of the frequency variable usually leads to a transformed state-space representation in which the 'D' matrix is regular. The eigenvalue calculation can then be performed, and the zeros obtained by inverting the bilinear transformation [12]. By repeating this procedure with a different transformation, the results can be checked. However, this check can on occasions be misleading, and an alternative approach is available. This is to search iteratively in a region of the complex plane for a point at which an eigenvalue of the frequency response matrix is zero.

A third way of calculating zeros in CLADP is to use the matrix manipulation routine, which includes algorithms for solving generalized eigenvalue problems for pairs of matrices [13], using 'QZ' techniques [14].

If a transfer-function description of a model is needed, it is obtained from a state-space description by pointwise evaluation of both $det(sI\text{-}A)$ and $(sI\text{-}A)^{-1}$, followed by fitting polynomials to the values of $det(sI\text{-}A)$ and the elements of $C\ adj(sI\text{-}A)B$, where the adjoint matrix is obtained by multiplying values of $(sI$-

$A)^{-1}$ by values of the polynomial which approximates the determinant. (Of course, CLADP also allows models to be specified directly by their transfer function matrices.)

Conversion from continuous-time to discrete-time state-space models is performed by first using 'scaling and squaring' to obtain an approximation of $\exp(AT)$ (where T is the inter-sample interval). This is the least unsatisfactory of the methods reviewed in [15]. The input matrix $(\exp(AT)-I)A^{-1}B$ is then obtained by solving the linear equation $AX = (\exp(AT)-I)B$ for X.

When a potentially precarious computation is performed in CLADP, some check is usually made on the correctness of the result, and a message is output to the user if there are indications that the result is unreliable. However, there are a few cases, such as the computation of zeros, in which the user is not alerted to possible problems. The range of facilities available in CLADP is so wide that almost any computation can be checked by some means, given sufficient ingenuity on the part of the user. This makes CLADP not only powerful, but also reliable if used intelligently, but it undeniably falls short of the ideal situation, in which the designer could proceed confidently at every step without needing to know the details of the algorithms he is executing. It remains to be seen how closely such an ideal can be approached.

4. Structure and Portability

CLADP is written entirely in FORTRAN, and contains its own libraries of linear algebra and graphics subroutines. It consists of a main program and about 30 major subroutines, one of which is the 'supervisor.' The main program is little more than a multi-position switch, which calls one or other of the major routines. Initially the user enters a one-word command which is interpreted by the supervisor: the supervisor sets up a sequence of up to 10 major routines which are to be called, in the appropriate order. For example, the command 'NYQUIST' causes the 'Nyquist Calculation' routine to be run first, followed by the 'Nyquist Display' routine. Each of these routines writes messages to the terminal screen and accepts input from the keyboard (via input subroutines). Extended command lines with arguments are not used.

Altogether there are about 400 subroutines, and the source code runs to about 10^5 lines of FORTRAN. Although the user interacts directly with subroutines at various levels, input from the keyboard is processed by a limited number of machine-dependent routines. Consequently CLADP is reasonably portable between machines. To date it has been installed on GEC 4070, GEC 4090, Prime 550 and Vax 11/780 computers. Earlier versions have been installed on a PDP 10, PDP 11/45, and a Honeywell 6000. On the GEC machines the compiled and linked object code occupies about 800 kbytes, and a further 200 kbytes are taken up by data arrays.

5. Applications

A number of applications of CLADP for control system design has been described in the literature. Foss *et al* [17] report its use for the control of a two-bed catalytic reactor, while Grimble and Fotakis [18] use CLADP for the design of shape control systems for a Sendzimir steel rolling mill. Kouvaritakis and Edmunds [19] and Foss [20] describe the use of CLADP for the design of gas-turbine control systems. Limebeer and Maciejowski [21] use CLADP to design controllers for a large turboalternator, and for a two-gimbal gyroscope.

6. Commercial Availability

Marketing of CLADP is being undertaken by Compeda Ltd., of Stevenage, U.K., to whom all commercial enquiries should be addressed.

7. Future Development

Our aim is to develop a package which is as useful as possible to an industrial designer. To achieve this requires a careful balancing of its synthesis and design aspects. Ideally, the designer's role would consist mainly of specifying his requirements, having the computer automatically synthesize a controller, and then evaluating the apparent price to be paid for the requirements stipulated (in terms of controller complexity, gains, actuator drive levels, etc). In order to enable the designer to make an accurate initial estimate of a feasible closed-loop specification, he should first carry out a careful initial analysis of the plant he is dealing with. Hence the main emphases in future developments will be on:

(i) increased flexibility of use, and a wider range of analysis tools in the pre-synthesis phase;
(ii) a high degree of automation in the synthesis phase, given an accurate specification of the required closed-loop behavior;
(iii) a wide and flexible range of analysis tools for the post-synthesis phase, including facilities for the investigation of nonlinear aspects of system behavior.

One particular design approach, which we have called the Quasi-classical approach, has been extensively investigated and will shortly be incorporated in a new version of CLADP. This is based on singular-value and generalized polar decompositions of transfer-function matrices, and enables one to aim at a simultaneous satisfaction of specifications on stability, performance and robustness. In this approach, particular emphasis is given to the robustness aspects of closed-loop behavior. After a singular-value decomposition of the transfer-function matrix, phase information is transferred from the singular-direction frames to the singular values to generate a set of what have been called Quasi-Nyquist diagrams. A careful analysis of robustness behavior then gives a structure for the controller which uses the singular-direction frames of the plant (in reversed order) but with appropriately-specified Quasi-Nyquist diagrams. The usefulness of this approach stems from the fact that it enables one to specify the compensating controller in a way which handles all three key aspects of behavior: stability, performance and robustness. A further advantage of this quasi-classical approach is that it is well suited to the computer-synthesis phase of design. The controller is handled in the form of a general matrix-fraction decomposition whose parameters are optimized using a double (i.e. two-nested loops) weighted least-squares procedure. Plants with different numbers of inputs and outputs can be handled in this controller synthesis approach which is described in detail in [16].

8. References

[1] A. G. J. MacFarlane, and I. Postlethwaite, "The generalized Nyquist stability criterion and multivariable root loci," *Int. J. Cont.*, vol. 25, pp. 81–127, 1977.

[2] A. G. J. MacFarlane, and D. F. A. Scott-Jones, "Vector gain," *Int. J. Cont.*, vol. 29, pp. 65–91, 1979.

[3] J. C. Doyle and G. Stein, "Multivariable feedback design: concepts for a classical/modern synthesis," *IEEE Trans. Auto. Contr.*, vol. AC-26, pp. 4–16, 1981.

[4] I. Postlethwaite, J. M. Edmunds, and A. G. J. MacFarlane, "Principal gains and principal phases in the analysis of linear multivariable feedback systems," *IEEE Trans. Auto. Contr.*, vol. AC-26, pp. 32–46, 1981.

[5] J. M. Edmunds, "Control system design and analysis using closed-loop Nyquist and Bode arrays," *Int. J. Contr.*, vol. 30, pp. 773–802, 1979.

[6] D. J. N. Limebeer, "*The application of generalized diagonal dominance to linear system stability theory.*, Cambridge University Engineering Dept. Technical Report, CUED/F-CAMS/TR-222, 1982.

[7] H. H. Rosenbrock, "*Computer-aided control system design*," London, Academic Press, 1974.

[8] D. Q. Mayne, "The design of linear multivariable systems," *Automatica*, vol. 9, pp. 201–207, 1973.

[9] A. G. J. MacFarlane and B. Kouvaritakis, "A design technique for linear multivariable feedback systems," *Int. J. Contr.*, vol. 25, pp. 837–879, 1977.

[10] B. C. Moore, "Principal component analysis in linear systems, controllability, observability, and model reduction," *IEEE Trans. Auto. Contr.*, vol. AC-26, pp. 17–32, 1981.

[11] J. H. Wilkinson, and C. Reinsch, "*Handbook for automatic computation, vol. II: linear algebra*," New York: Springer, 1971.

[12] B. Kouvaritakis, and J. M. Edmunds, "Multivariable root loci: a unified approach to finite and infinite zeros," *Int. J. Contr.*, vol. 29, pp. 393–428, 1979.

[13] A. J. Laub, and B. C. Moore, "Calculation of transmission zeros using QZ techniques," *Automatica*, vol. 14, pp. 557–566, 1978.

[14] C. Moler, and G. W. Stewart, "An algorithm for generalized matrix eigenvalue problems," *SIAM J. Num. Anal.*, vol. 10, pp. 241–256, 1973.

[15] C. Moler, and C. van Loan, "Nineteen dubious ways to compute the exponential of a matrix," *SIAM Review*, vol. 20, pp. 801–836, 1978.

[16] Y. S. Hung, and A. G. J. MacFarlane, "A quasi-classical approach to multivariable feedback systems," to be published in Lecture Notes in *Control and Information Sciences*, Springer-Verlag, 1982.

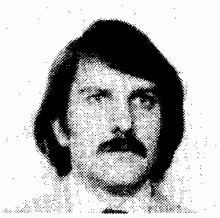
[17] A. S. Foss, J. M. Edmunds, and B. Kouvaritakis, "Multivariable Control System for Two-Bed Reactors by the Characteristic Locus Method," *I & EC Fundamentals*, vol. 19, pp. 109–117, 1980.

[18] M. J. Grimble, and J. Fotakis, "The design of strip shape control systems for Sendzimir mills," *IEEE Trans. Auto. Contr.*, vol. AC-27, pp. 656–666, 1982.

[19] B. Kouvaritakis, and J. M. Edmunds, "The characteristic frequency and characteristic gain design method for multivariable feedback systems," in: M. K. Sain, J. L. Peczkowski, and J. L. Melsa, (eds), *Alternatives for linear multivariable control*, (Chicago: National Engineering Consortium), pp. 229–246, 1978.

[20] A. M. Foss, "A practical approach to the design of multivariable control strategies for gas turbines," *ASME Paper No. 82-GT-150*, presented at the 27th International Gas Turbine Conference, Wembley, 1982.

[21] D. J. N. Limebeer, and J. M. Maciejowski, "*Two tutorial examples of multivariable control system design,*" Cambridge Univ. Eng. Dept. Technical Report, CUED/F-CAMS/TR.-229, 1982.



Jan Macieiowski received his B.Sc. degree in Automatic Control from the University of Sussex in 1971, and his Ph.D. in Control Engineering from the University of Cambridge in 1978. From 1971 to 1974 he was a systems engineer with Marconi Space and Defence Systems Ltd. He has held lecturing appointments at Imperial College, London, and at the University of Warwick. He is currently a Research Associate with Cambridge University Engineering Department, and a Fellow of Pembroke College, Cambridge. He is a member of the Science and Engineering Research Council's Special Interest Group in Control Engineering. His current research interests are CAD of control systems, and the design of macroeconomic regulation policies. His thesis, *The modelling of systems with small observation sets*, has been published by Springer Verlag in their series *Lecture notes in control and information sciences*.

Alistair G. J. MacFarlane was born in Edinburgh, Scotland in 1931. He received the B.Sc. and D.Sc. degrees from the University of Glasgow in 1953 and 1968; the Ph.D. degree from the University of London in 1964; the M.Sc. degree from the University of Manchester in 1973; and the M.A. and Sc.D. degrees from the University of Cambridge in 1974 and 1979 respectively. After graduating in electrical engineering from the University of Glasgow, he joined the Metropolitan-Vickers Electrical Company in Manchester in 1953 to complete his post-graduate training in the Radar and Servomechanisms Division. After some years working on a variety of feedback and signal-processing problems he became group leader of the Moving Target Indication and Receiver Laboratories of that company in 1956-58. He was then appointed Lecturer in Electrical Engineering in Queen Mary College, University of London, and was promoted to Reader in Electrical Engineering in 1965. In 1966 he moved to the University of Manchester Institute of Science and Technology as Reader in Control Engineering in the newly established Control Systems Centre, becoming Professor of Control Engineering in 1969. Following this he was elected to a Chair of Engineering in the University of Cambridge in 1974 and became head of the Control and Management Systems Division of the Engineering Department at Cambridge. His research interests are in the fields of feedback theory, dynamical theory, and automatic control. For papers on these topics he was awarded the Institution Premium of the IEE in 1966, the IEE Control and Automation Division Premium in 1970, the ICI prize of the Institute of Measurement and Control in 1975, and an ASME Centennial Medal in 1980. Dr. MacFarlane is Vice-Master of Selwyn College, Cambridge, a Fellow of the IEE, of the Institute of Measurement and Control, and of the IEEE; and Editor of the International Journal of Control. He is a member of the Council of the Science and Engineering Research Council, and of the Executive Council of the International Federation of Automatic Control. He was elected to the Fellowship of Engineering in 1981.