T.M. M.T. Borelli

RENSSELAER POLYTECHNIC INSTITUTE
KNOWLEDGE AND THOROUGHNESS
1824

Rensselaer Polytechnic Institute

Troy, New York

Rensselaer Polytechnic Institute
Troy, New York 12189

CR-119798

Final Report - Vol. II
Contract No. NAS8-21131
Covering the period Nov. 4, 1969-Apr. 4, 1971
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Compensator Design for Low-Sensitivity Linear Time-Invariant Systems

by

Lutz Willner

Submitted on behalf of

Rob J. Roy
Professor
Systems Engineering Division

COMPENSATOR DESIGN FOR
LOW-SENSITIVITY LINEAR
TIME-INVARIANT SYSTEMS

by

Lutz Willner

Rensselaer Polytechnic Institute
Troy, New York

June, 1971

COMPENSATOR DESIGN FOR LOW-SENSITIVITY LINEAR TIME-INVARIANT SYSTEMS

by

Lutz Willner

A Thesis Submitted to the Graduate
Faculty of Rensselaer Polytechnic Institute
in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: Control Systems

Approved by the
Examining Committee

Rob Roy

Rob J. Roy, Thesis Advisor

Paul M. DeRusso, Member

Howard Kaufman

Howard Kaufman, Member

Derek E. McBrinn

Derek E. McBrinn, Member

Philip M. Zaretzki

Philip M. Zaretzki, Member

Rensselaer Polytechnic Institute
Troy, New York

June, 1971

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

The author wants to thank his committee members for their help, concern and guidance. Special gratitude is due to his advisor, Dr. Roy, for good counsel and skillful help in outmaneuvering obstacles. The acknowledgement would be incomplete without thanking the author's friends, Dr. Derek E. McBrinn and Dr. Raj Ramachandran for many clarifying discussions and helpful suggestions.

Miss Kathy Reid's expert typing of the manuscript is promptly acknowledged.

This research was performed at Rensselaer Polytechnic Institute under the sponsorship of NASA Contract No. NAS8-21131. Two one-year fellowships from "Deutsche Forschungsgemeinschaft" and Xerox, respectively, helped to ease the financial burden of graduate studies.

ABSTRACT

This work is concerned with the design of feedback compensators to obtain linear time-invariant systems which are insensitive to parameter variations. A new concept in sensitivity design is introduced. A sensitivity function is derived, based on the condition number $\varkappa = \inf_M \|M\| \|M^{-1}\|$, where M is a matrix which transforms the system under consideration to diagonal form. The knowledge of $\varkappa$ permits the computation of a bound on permissible parameter variations for which the closed-loop system will still exhibit a specified minimum stability.

An algorithm to locally minimize the sensitivity function with respect to the compensator parameters is developed, programmed for the digital computer and applied to the design of three systems.

To solve the compensator design problem it was also necessary to develop a fast and efficient pole-placement algorithm and an algorithm to determine $\varkappa$. Both algorithms are potentially very useful and are described in detail in Part 4 of this dissertation. The pole-placement algorithm was combined with Kleinman's iterative method of solving the steady-state matrix Riccati equation and programmed for the digital computer. This resulted in a self-contained program package, which is able to compute all-state feedback gains for any specified set of realizable closed-loop eigenvalues. If stable eigenvalues are specified, the resulting feedback gains can be used to initialize the Riccati equation solver. The capability of this program package is demonstrated by 11 examples.

PART 1

1.1 Introduction

Physical processes can be analyzed with only a certain degree of accuracy. This is due to measurement errors and parameter variation because of aging, heat and other influences. Further inaccuracies are introduced by linearization in order to obtain a system model which is mathematically tractable. The modelling is usually done in the convenient state variable form[1,12], permitting the description of multi-input, multi-output systems. Once a linear model of the process is established, desired system responses can be obtained by applying the theories of optimal control[10,11,12,36], state estimation[2,13,24,26] and compensator design[4,18-22]. The question is, how valid are these results in light of the system parameter uncertainties?

Efforts to answer this question resulted in system sensitivity analysis. Already Bode[38] was concerned about system sensitivity to parameter variations and laid the ground work for this branch of control theory. Since then many extension and generalizations[42] of his work have been published.

The present dissertation will apply sensitivity analysis to the design of low order feedback compensators for linear time-invariant systems with large parameter uncertainties. In doing so a new measure of sensitivity is introduced. This new sensitivity measure takes into account the uncertainties of all parameters of the closed-loop system, consisting of the plant and the feedback compensator.

1.2 Historical Review

Compensation of control systems in order to achieve a pre-specified system behavior has been common since the early days of control system design. However, 'classical' results are applicable only to time-invariant, single-input, single-output systems and result in design procedures for feed-forward or minor-loop compensators[29]. The compensator design procedures relies mostly on Nyquist plots or Bode diagrams.

The introduction of 'state variables'[1,12] in the 1950's to describe the dynamic behavior of control systems gave new impetus for the development of methods to achieve desired system responses. Consider the linear dynamic system

$$\underline{\dot{x}}(t) = A(t)\, \underline{x}(t) + B(t)\, \underline{u}(t) \tag{1.2-1a}$$

with output

$$\underline{y}(t) = C(t)\, \underline{x} \tag{1.2-1b}$$

where $A(t)$, $B(t)$ and $C(t)$ are $n \times n$, $n \times m$ and $p \times n$ matrices, respectively. Whenever the system, given by equations (1) is time-invariant, i.e., $A(t) = A$, $B(t) = B$, $C(t) = C$ for all t, and controllable and observable[10], it is possible to place the closed-loop poles arbitrarily by feeding back all system states[3]. All-state feedback requires for the above system, that the matrix C be of rank n. Then system (1) can be transformed into an equivalent system with states $\underline{z}(t) = C\, \underline{x}(t)$ and output $\underline{y}(t) = \underline{z}(t)$.

Another problem connected with system (1) is that of optimizing the system with respect to some performance criterion, i.e., to find a control that minimizes the cost criterion. The most widely chosen performance index is a quadratic integral functional of the form

$$J(u) = \int_{t_0}^{t_f} \frac{1}{2}(\underline{x}^T(\tau)\ Q(\tau)\ \underline{x}(\tau) + u^T(\tau)\ R(\tau)\ \underline{u}(\tau))\ d\tau \qquad (1.2\text{-}2)$$

For the case where all states of system (1) are available, e.g., $C(t) = I$, the well known solution[3] to the optimization problem is given by the linear feedback law

$$\underline{u}(t) = -G_0(t)\ \underline{x}(t) \qquad (1.2\text{-}3)$$

where

$$G_0(t) = +\ R^{-1}(t)\ B^T(t)\ P(t) \qquad (1.2\text{-}4)$$

If $Q(t)$ and $R(t)$ are symmetric positive semi-definite and positive definite weighting matrices, respectively, and system (1) is controllable, then $P(t)$ is the unique, symmetric, positive definite solution of the matrix Riccati equation

$$-\dot{P}(t) = P(t)\ A(t) + A^T(t)P(t) - P(t)B(t)R^{-1}(t)B^T(t)P(t) + Q(t)$$

$$P(t_f) = [0] \qquad (1.2\text{-}5)$$

When system (1) is time-invariant and $C = I$, Letov[30] showed how to choose closed-loop poles yielding optimality of the system with respect to a quadratic performance criterion as given by equation (2). In this case, Q and R are constant, and the integration interval is $[0, \infty)$.

Generally it is not possible to achieve arbitrary pole assignment for time-invariant systems if not all states of system (1) are available for feedback. In this case only as many eigenvalues can be shifted arbitrarily as independent outputs y_i are available[31,32]; the remaining eigenvalues may move anywhere in the complex s-plane. The use of cost functional (2) to optimize systems with unavailable states does not lead to tractable results as in the all-state case. Cassidy[27] developed a modified quadratic performance criterion which yields optimal results for systems which can be stabilized by partial state (or output) feedback. The drawback of his method is that it has to be started with a stable system.

If no stabilizing feedback gains exist or can be determined, dynamic feedback compensators have to be implemented to achieve stability or optimality with respect to some performance index. The task of designing a dynamic feedback compensator is usually achieved in one of the following two ways:

(a) Determine an observer[2,13,26,33] which yields an estimate of the unavailable states. Use the estimates and available states and proceed as in the all-state case in determining appropriate feedback gains. The overall structure of observer and feedback gains will be termed dynamic compensator.

(b) Since only a fixed linear combination of the states is needed to achieve a specified system response, estimation and feedback will be immediately combined[4,18-20]. This method[4,18] may yield lower order compensators than the first method, but may also be computationally more difficult[25].

The major difference between the two approaches is the fact, that the second method does not obtain explicit estimates of the unavailable states[25]. Both methods allow arbitrary placement of all closed-loop poles of time-invariant systems and will yield asymptotic stability for time-varying systems if so desired.

The theory of observers and compensators is needed for the design of real systems, because many practical systems have not all states available as outputs. Another characteristic of practical systems is the fact, that they usually cannot be described as accurately as needed for the proper application of control laws. To deal with inaccuracies or slow variations due to aging of the system parameters the sensitivity of certain desired system properties with respect to possible parameter changes has been analyzed.

Numerous papers[34,35,37-43] deal with sensitivity analysis and design. To use the classification of Rohrer and Sobral[35], the sensitivity design methods can be divided into two categories, absolute and relative sensitivity designs. Absolute sensitivity is concerned with the change of some desired system quantity, e.g., the transfer function $T(s)$, due to the change of some parameter x. Thus, Bode[38], Cruz and Perkins[39], Morgan[41], etc. deal, in the scalar case, with sensitivity functions of the type

$$S_x^T = \frac{\Delta T}{T} \cdot \frac{x}{\Delta x} \tag{1.2-6}$$

Equivalent formulations for the vector case are available[39].

Relative sensitivity is applied when describing the deterioration of the performance index, e.g., equation (1.1-2), due to parameter variations. One of several possible expressions to describe relative sensitivity is the change of performance index $J(\underline{v}, \underline{u})$ due to a change in system parameters $\underline{v}$.

$$S\left[\underline{v}, \underline{u}(t)\right] = \frac{J(\underline{v}, \underline{u}^0(t)) - J(\underline{v}^0, \underline{u}^0(t))}{\left|J(\underline{v}^0, \underline{u}^0(t))\right|} \tag{1.2-7}$$

where $J(\underline{v}^0, \underline{u}^0(t))$ denotes the optimal cost, obtained for optimal control $\underline{u}^0(t)$ and nominal system parameters $\underline{v}^0$. Clearly, when $\underline{v}$ differs from $\underline{v}^0$, $\underline{u}^0$ is no longer the optimal control, but a control resulting in the cost $J(\underline{v}, \underline{u}^0(t))$. Reports of Rohrer and Sobral[35], McClamroch[43] et al., Cassidy[27], Tuel[37], Porter[34], etc., are just some of many publications which are devoted to the derivation and application of some type of relative sensitivity.

Further references for sensitivity designs and problems can be found in reference 42 .

The goal of both methods is to minimize S_x^T and $S\left[\underline{v}, \underline{u}(t)\right]$. To achieve this goal in a mathematically tractable way it is assumed that the parameter variations are 'small'. After the design is completed the effects of 'large' parameter variations are investigated[39,43]. None of the methods establishes an a priori bound on the parameter uncertainties, for which the system characteristic will vary within a permissible region only. Furthermore it is assumed that the feedback structure can be implemented accurately, and thus possible inaccuracies of this part of the closed-loop system are neglected.

Although absolute sensitivity was introduced and analyzed[38] earlier than relative sensitivity, the first approach did not really progress past a trial and error design procedure[39], especially when only bounds on the parameter variations were available. Tuel[37] and Cassidy[27] showed that, by making use of modern optimization techniques, the relative sensitivity approach yields meaningful results with comparative ease, as long as only few parameters are subject to variations.

It is the intention of this dissertation to eliminate some of the short-comings of both sensitivity approaches. To do so a completely different route of investigation is chosen.

1.3 Scope and Contribution of this Work

This dissertation deals with the sensitivity of eigenvalues of a closed-loop system consisting of a linear time-invariant plant and a feedback compensator. The feedback compensator is of sufficient order to permit arbitrary pole assignment for and thus stabilization of the closed-loop system. It is assumed, that it is only desired to obtain closed-loop poles within a certain region of the complex s-plane, and not to fix the pole locations a priori.

Since the compensator permits arbitrary pole-assignment it is assumed that the closed-loop system will have distinct eigenvalues only. Then there exists a non-singular matrix, M, which transforms the closed-loop system to diagonal form. A measure of the sensitivity of the system matrix is given by

$$\varkappa = \inf_{M} \| M \| \, \| M^{-1} \| \tag{1.3-1}$$

Thus, by shifting the pole-location within the specified region, it is possible to obtain a new $\varkappa_1$, having a lower (or equal) value than $\varkappa$ corresponding to the original set of poles; i.e., $\varkappa$ can be locally minimized.

Usually stability is a main design criterion. It is desired to keep the closed-loop system stable even under the influence of large parameter variations. A very conservative bound on the maximum permissible parameter variation, for which the system will still be stable, is given by

$$\frac{-\max \operatorname{Re}(\lambda_i)}{\varkappa} \tag{1.3-2}$$

where λ_i is an eigenvalue of the closed-loop system. To increase this bound, it no longer suffices purely to minimize $\varkappa$ as defined by equation (1), but to maximize expression (2), at least locally.

Part 2 will give the theory behind the choice of this measure of sensitivity. Also contained in Part 2 will be the numerical algorithm for designing the compensator and computing expressions (1) and (2) in a slightly modified form.

Part 3 will present three numerical examples to illustrate the theory of Part 2.

The computation of equation (1) and iteration on the value of $\varkappa$ required some efficient numerical algorithmsfor pole-assignment and determination of $\inf_M \|M\| \, \|M^{-1}\|$. Since these algorithms are of general application, they are described in detail in Part 4. The pole-placement algorithm is an especially useful tool in determining a set of stable gains to initialize the Kleinman[9] iterative technique for the solution of the Riccati equation. Numerical examples for the pole-placement algorithm will also be given in Part 4.

Thus the contributions of this dissertation are:

(a) a new sensitivity measure which takes into account the variation of all closed-loop parameters;

(b) a bound (still very conservative) given on the parameter uncertainty for which the closed-loop system is guaranteed to exhibit a specified minimum stability;

(c) a very efficient algorithm for arbitrary pole assignment and initialization of the iterative Riccati matrix equation; and

(d) an algorithm to compute $\varkappa = \inf_{M} \|M\| \|M^{-1}\|$, where the matrix norm $\|\cdot\|$ is induced by either the 'one' or 'infinity' absolute vector norm.

PART 2

COMPENSATOR DESIGN FOR LOW SENSITIVITY LINEAR SYSTEMS

2.1 Introduction

This part deals with linear time-invariant plants which are inaccurately known and have fewer independent outputs than states. The aim is to design a low-order feedback compensator such that the closed-loop system, consisting of the plant and the feedback compensator, has poles in some desired region of the complex s-plane. Furthermore the poles should be insensitive towards variations of the system parameters.

The goal will be achieved by choosing an estimator of minimum order as developed by Luenberger[13] and by using the estimates together with the plant outputs to determine a set of feedback gains, which will shift the poles of the closed-loop system to locations within a desired region of the s-plane. A sensitivity measure, which will be defined in the following section, is evaluated for this set of eigenvalues. Then the eigenvalues are moved within the desired region in order to decrease the sensitivity measure. The method will terminate if a local sensitivity minimum is obtained.

A conservative bound on the maximum permissible parameter variation for which the eigenvalues of the closed-loop system will still be stable is derived from an extension of Gersgorin's[14] theorem.

2.2 Theory

Consider the linear time-invariant, controllable and observable plant

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t) \qquad (2.2\text{-}1a)$$

$$\underline{y}(t) = C\,\underline{x}(t) \qquad (2.2.\text{-}b)$$

where A, B and C are matrices of order $(n \times n)$, $(n \times m)$ and $(p \times n)$ respectively. Let B and C be of maximum ranks m and p, respectively. Then (A,B) represents a controllable system[13], if

$$\operatorname{rank}\left[B, AB, A^2B, \ldots, A^{\nu-1}B\right] = n \qquad \text{for} \quad \nu \leq n \tag{2.2-2}$$

Similarly (A^T, C^T) is observable if

$$\operatorname{rank}\left[C^T, A^TC^T, (A^T)^2C^T, \ldots, (A^T)^{\mu-1}C^T\right] = n \qquad \text{for} \quad \mu \leq n \tag{2.2-3}$$

If C is of maximum rank p and $p < n$ then it is not possible to achieve arbitrary pole assignment of all n poles by static feedback alone; a dynamic compensator is needed. The theoretically lowest order of the compensator[4,18] to permit arbitrary placement of all closed-loop poles is equal to

$$\beta = \min(\nu-1, \mu-1)$$

$$= \min(\text{controllability index} - 1, \text{observatility index} - 1) \tag{2.2-4}$$

where ν and μ are the smallest integers to yield equalities (2) and (3), respectively. Since B and C were assumed to be of maximum ranks m and p, respectively, upper bounds for ν and μ are given by

$$\nu-1 \leq n-m \tag{2.2-5a}$$

$$\mu-1 \leq n-p \tag{2.2-5b}$$

However, choosing the compensator order according to equation (4) requires

(1) that all three matrices A, B and C be accurately known and;

(2) that some fixed linear combination of the states of plant (1) be acceptable for implementation.

To take parameter uncertainties into account and to enable the estimation of unavailable states from the outputs $\underline{y}$, the compensator order is chosen to equal $n-p$, assuming, that $p \geqslant m$ (in the case $p < m$, take the dual of system (1)). Estimation of all unavailable states makes the implementation of a feedback compensator computationally simple, and changes after realization of the feedback compensator can be made with static feedback, if so desired.

Let the compensator of order $q = (n-p)$ be described by

$$\dot{\underline{z}}(t) = F\,\underline{z}(t) + G\,\underline{y}(t) \tag{2.2-6a}$$

$$\underline{u}(t) = H\,\underline{z}(t) + J\,\underline{y}(t) \tag{2.2-6b}$$

Combining equations (1) and (6) yields the closed-loop system

$$\begin{bmatrix} \dot{\underline{x}}(t) \\ \dot{\underline{z}}(t) \end{bmatrix} = \begin{bmatrix} A+BJC & BH \\ GC & F \end{bmatrix} \begin{bmatrix} \underline{x}(t) \\ \underline{z}(t) \end{bmatrix} \tag{2.2-7}$$

The augmented state vector $[\underline{x}, \underline{z}]^T$ will be defined

$$\underline{w}^T(t) \triangleq [\underline{x}(t), \underline{z}(t)]^T \tag{2.2-8a}$$

and the closed-loop system matrix

$$K \triangleq \begin{bmatrix} A+BJC & BH \\ GC & F \end{bmatrix} \tag{2.2-8b}$$

Thus, equation (7) becomes

$$\dot{\underline{w}}(t) = K\,\underline{w}(t) \tag{2.2-9}$$

For the sake of clarity it is now assumed, that only the plant matrix A has parameter uncertainties. Later on it will be shown that the desensitization of the eigenvalues of the closed-loop system is with respect to all elements of the matrix K, not only with respect to some of them. Let A be decomposable into a nominal matrix A_0 and a parameter variation δA, i.e.,

$$A = A_0 + \delta A \tag{2.2-10}$$

Usually δA cannot be properly defined. The best one can do is to obtain some upper bound on the uncertainties of every element of the nominal matrix A_0.

Substitution of expression (10) in (8b) yields

$$K \triangleq K_0 + \delta K = \begin{bmatrix} A_0+BJC & BH \\ GC & F \end{bmatrix} + \begin{bmatrix} \delta A & 0 \\ 0 & 0 \end{bmatrix} \tag{2.2-11}$$

and therefore

$$\dot{\underline{w}}(t) = (K_0 + \delta K)\, \underline{w}(t) \tag{2.2-12}$$

The closed-loop system is shown in figure 2.2-1.

Since the parameter uncertainty is usually not explicitly known the compensator to achieve some set of closed-loop eigenvalues will be designed with respect to the nominal system parameters. Consequently two questions arise:

(1) what is the influence of δA on the set of nominal closed-loop eigenvalues?

(2) can the influence of δA on the nominal eigenvalues be minimized?

Answers to the above questions will be given after the presentation of a number of definitions and an extension of Gersgorin's theorem.

Definitions:

D1)[14] K_0 is a simple matrix, iff for each distinct eigenvalue of K_0 the multiplicity is equal to the geometric multiplicity.

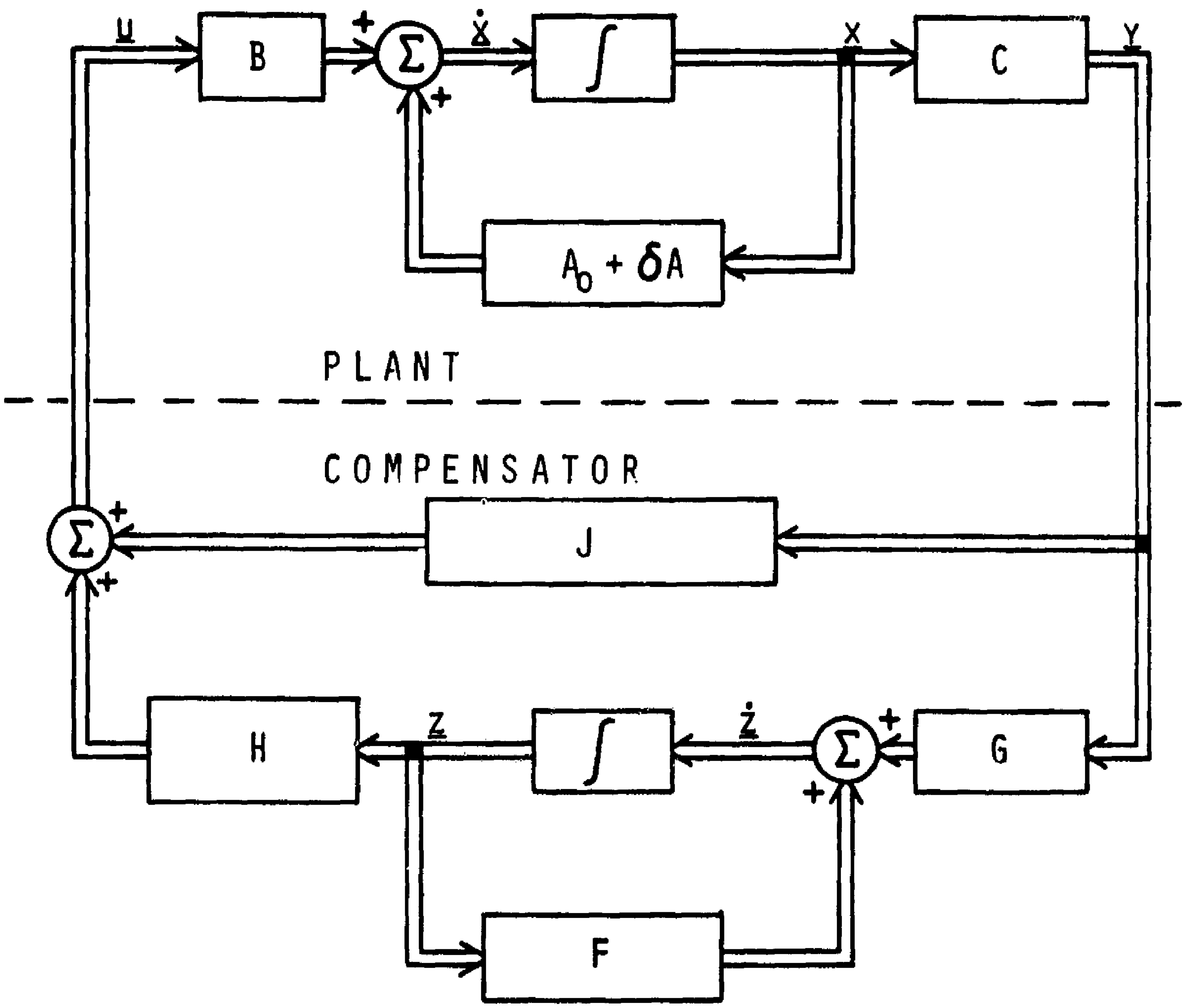

FIGURE 2.2-1

Closed-Loop System

i.e., K_0 is simple if it has a full set of linearly independent eigenvectors and thus can be transformed to diagonal form.

D2)[14] $\mathcal{C}_{n \times n}$ denotes the set of all complex valued $n \times n$ matrices.

D3) For simple matrices K_0 there exists a non-singular matrix M which transforms K_0 to diagonal form L, i.e., $K_0 = M\, L\, M^{-1}$. Matrix M is non-unique, $M_1 = M \cdot \mathrm{diag}\,(\beta_1, \ldots, \beta_{n+q})$ with $\beta_i \neq 0$ is also a similarity transformation matrix.

D4) The infimum of the condition numbers for the set of matrices M, transforming K_0 to diagonal form L, is defined to be

$$\varkappa = \inf_{M} \|M\| \, \|M^{-1}\|$$

D5) A matrix norm induced by a vector norm is defined by[17]

$$\| K_0 \| = \sup_{\underline{x} \neq \underline{0}} \frac{\| K_0\, \underline{x} \|}{\| \underline{x} \|}$$

D6)[14] An absolute vector norm $N(\underline{x})$ depends only on the absolute values of the elements of the vector argument.

Theorem[14]:

Let K_0, $\delta K \in \mathcal{C}_{(n+q)\times(n+q)}$ with K_0 simple. If K_0 has eigenvalues $\lambda_1, \ldots, \lambda_{n+q}$, M is the transformation of K_0 to diagonal form L, μ is an eigenvalue of $K = K_0 + \delta K$, and for a matrix norm induced by an absolute vector norm

$$\rho = \|\delta K\| \inf_{M} \|M\| \, \|M^{-1}\| = \|\delta K\| \, \varkappa \tag{2.2-13}$$

then μ lies in at least one of the disks

$$|s - \lambda_i| \leq \rho \qquad i = 1, \ldots, n+q \tag{2.2-14}$$

Proof:

(This proof is repeated here, since it gives some insight into equations (13) and (14) and shows the relationship to Gersgorin's theorem). Since μ is an eigenvalue of $K = K_0 + \delta K$, there is a vector $\underline{y} \neq \underline{0}$ for which

$$(K_0 + \delta K)\, \underline{y} = \mu\, \underline{y} \tag{2.2-15}$$

Because K_0 is simple, $K_0 = M\, L\, M^{-1}$ and hence

$$(L + M^{-1}\, \delta K\, M)\, \underline{z} = \mu\, \underline{z} \tag{2.2-16}$$

where $\underline{z} = M^{-1}\, \underline{y}$. Thus

$$(\mu\, I - L)\, \underline{z} = M^{-1}\, \delta K\, M\, \underline{z} \tag{2.2-17}$$

From the definition of the lower bound of a matrix with respect to an absolute vector norm N it follows that

$$\mathrm{glb}_N\, (\mu I - L) \leq \| M^{-1} \delta K\, M \| \leq \| \delta K \|\, \| M \|\, \| M^{-1} \|$$

whence

$$\min_i | \mu - \lambda_i | \leq \| \delta K \|\, \| M \|\, \| M^{-1} \| \tag{2.2-18}$$

Expression (18) must hold for every M transforming K_0 to diagonal form, thus

$$\min_i | \mu - \lambda_i | \leq \| \delta K \| \inf_M \| M \|\, \| M^{-1} \| = \rho \tag{2.2-19}$$

Hence, μ lies in at least one of the discs

$$| s - \lambda_i | \leq \rho$$

The connection to Gersgorin's[14] theorem is easily established. Consider expression (16) and define

$$D \triangleq L + M^{-1}\, \delta K\, M \tag{2.2-20}$$

and

$$\rho'_i = \sum_{\substack{k \\ k\neq i}} |d_{i_k}| = \sum_{\substack{k \\ k\neq i}} |(M^{-1} \; \delta K \; M)_{ik}| \tag{2.2-21}$$

Then according to Gersgorin's theorem[14] the eigenvalues of D lie in disks

$$|s - d_{ii}| \leq \rho'_i \qquad i=1,(n+q) \tag{2.2-22}$$

in the complex s-plane. But

$$d_{ii} = \lambda_i + (M^{-1} \; \delta K \; M)_{ii}$$

whence inequality (22) can be re-written as

$$\rho'_i \geq |s-\lambda_i - (M^{-1} \delta K \; M)_{ii}| \geq |s-\lambda_i| - |(M^{-1} \delta K \; M)_{ii}|$$

or

$$|s - \lambda_i| \leq \rho'_i + |(M^{-1} \delta K \; M)_{ii}| \tag{2.2-23}$$

Define

$$\rho_i = \rho'_i + |(M^{-1} \delta K \; M)_{ii}| = \sum_k |(M^{-1} \; K \; M)_{ik}| \tag{2.2-24}$$

Using the matrix norm induced by the infinity vector norm, ρ_i can be bounded above by

$$\rho_i \leq \|M^{-1}\|_\infty \; \|\delta K\|_\infty \; \|M\|_\infty \qquad i=1(n+q) \tag{2.2-25}$$

This must hold for all matrices M which transform K_0 to diagonal form L and thus

$$|s-\lambda_i| \leq \|\delta K\|_\infty \inf_M \|M^{-1}\|_\infty \; \|M\|_\infty \tag{2.2-26}$$

Inequality (26) is identical to expression (14) for the case of the infinity norm.

This theorem answers already the first of the two questions asked before introducing the theorem. Having designed a compensator such that K_0 has a specified set of eigenvalues $\{\lambda_i\}$, the eigenvalues of $K=K_0+\delta K$ will be in disks as given by expression (13) and (14). Clearly expressions (13) and (14) are for a worst case design.

Once an appropriate matrix norm is chosen, $\|\delta A\|$ and thus, by assumption, $\|\delta K\|$ are known. Then, by equation (13), the way to minimize the influence of $\|\delta K\|$ on the nominal eigenvalues $\{\lambda_i\}$ is by minimizing $\varkappa = \inf_M \|M^{-1}\| \, \|M\|$. $\inf_M \|M^{-1}\| \, \|M\|$ could be directly interpreted as a sensitivity measure. To desensitize the closed-loop system the eigenvalues are moved within a permissible region until some minimum of $\varkappa$ is found.

Inequality (14) says that asymptotic stability of the perturbed system matrix $K = K_0 + \delta K$ is ensured if the eigenvalues of the nominal system satisfy

$$\mathrm{Re}(\lambda_i) < -\rho \leq 0 \qquad \text{for} \quad i=1, n+p \tag{2.2-27}$$

If the actual closed-loop system $\underline{\dot{w}}(t) = K\,\underline{w}(t)$ is desired to exhibit a specified degree of stability, say $\mathrm{Re}(\mu) \leq -\gamma < 0$, where μ is an eigenvalue of K, then inequality (27) has to be amended to

$$\mathrm{Re}(\lambda_i) + \gamma < -\rho \leq 0 \tag{2.2-28}$$

Reformulation of inequality (28) leads to either

$$-\frac{\mathrm{Re}(\lambda_i)+\gamma}{\rho} > 1 \quad \text{or} \quad -\frac{\mathrm{Re}(\lambda_i)}{\rho+\gamma} > 1 \qquad \text{for } i=1,(n+p) \tag{2.2-29}$$

Both expressions are combined to form the promised new type of sensitivity function. This function is implicitly dependent on the eigenvalues λ_i of the nominal system and a parameter α_0, which depends on the ratio $\frac{\|G\|}{\|H\|}$ of the input- and output matrix of the compensator (G and H are defined in equation (6)). Then the sensitivity measure chosen becomes

$$f_s(\underline{\lambda}, \alpha_0) = 1 + \frac{\mathrm{Re}(\lambda_{max}) + \delta}{\varkappa \|\delta K\| + \gamma} \tag{2.2-30}$$

where λ_{max} denotes the least stable eigenvalue of K_0 (K_0 is assumed to be stable). When trying to minimize the sensitivity function f_s with respect to its arguments, it can be seen, that this will not necessarily yield the lowest possible $\varkappa$, but will essentially maximize $\left|[\mathrm{Re}(\lambda_{max})]/[\varkappa \|\delta K\|]\right|$. By maximizing this ratio the permissable perturbation of K_0 is maximized.

It is possible to include some region constraint in the sensitivity function, i.e., some penalty to force the desensitized eigenvalues to be close to a desired region in the s-plane. As computational results will show, this is not really necessary. If the eigenvalues of the system to be desensitized are chosen to lie within a specified region, it is most likely that the eigenvalues of the desensitized system will be in that region, too.

Equations (13) and (30) depend only on the norm of δK and not directly on the elements of this matrix. Hence, once a closed-loop system is designed, variations of the elements of matrices B, C, G, H, F, J are also taken into account, as long as $\|\delta K\|$ stays the same.

2.3 Computational Aspects

To summarize, the computational problems posed are to:

(1) determine an estimator of order $q=n-p$ having certain desired eigenvalues;

(2) use the plant outputs and estimates to compute a set of 'all-state' feedback gains yielding a plant with desired eigenvalues;

(3) combine estimator and feedback gains to obtain a compensator as described by equation (2.2-6);

(4) compute the matrices M and M^{-1} of eigenvectors of the closed-loop system, consisting of plant and compensator, and determine $\varkappa = \inf_M \| M \| \| M^{-1} \|$;

(5) compute the sensitivity measure f_s given by (2.2-30) and iterate on f_s so as to obtain a minimum.

Steps 1 to 3 have a common characteristic in so far as neither the estimator, nor the set of 'all-state' feedback gains nor the final feedback compensator are unique. This non-uniqueness can be easily demonstrated with an example for all-state feedback which is typical for all 3 cases.

Example:

Let $\dot{\underline{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t)$ (2.3-1a)

$\underline{y}(t) = C\,\underline{x}(t)$ (2.3-1b)

represent a second order system with

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad B = I_2 \quad \text{and} \quad C = I_2 \qquad (2.3\text{-}1c)$$

Where I_2 denotes the identity matrix of order 2.
Arbitrary pole-placement can be achieved by feeding back all states.

Choose the feedback law

$$\underline{u} = - G_0\, \underline{y} = - G_0\, \underline{x} = - \begin{bmatrix} g_{0_{11}} & g_{0_{12}} \\ g_{0_{21}} & g_{0_{22}} \end{bmatrix} \underline{x} \tag{2.3-2}$$

to obtain the closed-loop system

$$\dot{\underline{x}} = \begin{bmatrix} a_{11} - g_{0_{11}} & a_{12} - g_{0_{12}} \\ a_{21} - g_{0_{21}} & a_{22} - g_{0_{22}} \end{bmatrix} \underline{x} \tag{2.3-3}$$

Clearly, G_0 has infinitely many solutions for any one set of closed-loop poles. If system (1) would be a single-input system and controllable, then a unique $\underline{g}_0^T$ would exist for which

$$\dot{\underline{x}} = (A - \underline{b}\, \underline{g}_0^T)\, \underline{x} \tag{2.3-4}$$

will have a set of desired eigenvalues.

Before analyzing the above 5 problems in detail the following assumption is made to facilitate the theoretical and computational analysis.

Assumption:

The system given by equation (2.2-1) and repeated below is in observer canonical form, i.e.,

$$\dot{\underline{x}}(t) = A\, \underline{x}(t) + B\, \underline{u}(t) \tag{2.3-5a}$$

$$\underline{y}(t) = \left[I_p \;\vdots\; 0 \right] \underline{x}(t) \tag{2.3-5b}$$

This assumption is not overly restrictive, since any constant linear system, that is not initially in observer canonical form, but whose matrix C is of maximum rank p, can be transformed to an equivalent system, that is, to the desired observer canonical form (for finding the transformation see for instance Ash[2]).

For clarity the computational problems are dealt with below in separate paragraphs.

1) Estimator Design

Since the plant (eq. 2.2-1) has only p independent outputs, i.e., the first p states, $q=n-p$ states will be estimated. To do so an estimator of Luenberger[13] type is chosen. It is not the purpose of this dissertation to repeat the derivation of this type of estimator. The following equations are therefore stated without comment. The interested reader will find all necessary details in Ash[2], from where the equations are taken.

Let the estimator be described by

$$\dot{\xi}(t) = U\,\xi(t) + V\,\underline{y}(t) + W\,\underline{u}(t) \tag{2.3-6}$$

with

$$\xi(t_0) = \left[\, 0 \;\vdots\; I_q \,\right] \underline{x}(t_0)$$

The matrices U, V and W can be found by partitioning the nominal plant matrices as follows

$$A_0 = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \quad ; \qquad B = \left[\begin{array}{c} B_1 \\ \hline B_2 \end{array}\right] \tag{2.3-7}$$

where A_{11} is $(p\text{x}p)$, A_{22} is $(q\text{x}q)$, B_1 is $(p\text{x}m)$ and B_2 is $(q\text{x}m)$. The estimator matrices are determined by

$$U = A_{22} - G_e\,A_{12} \tag{2.3-8a}$$

$$V = A_{21} - G_e\,A_{12} + U\,G_e \tag{2.3-8b}$$

$$W = B_2 - G_e\,B_1 \tag{2.3-8c}$$

where G_e is an arbitrary matrix which is chosen to force U to satisfy certain conditions. In the present case, U is desired to have a specified set of eigenvalues.

Luenberger[13] and Ash[2] proved that a closed-loop system formed of a plant, an estimator and a set of all-state feedback gains G_s will have eigenvalues consisting of those of the matrix U and those of $A_0 + B\,G_s$. Hence some of the desired closed-loop system eigenvalues are selected to be realized by the estimator.

Since the eigenvalues of $U = A_{22} - G_e A_{12}$ are known, G_e has to be determined accordingly. Consider the system

$$\dot{\boldsymbol{\eta}}(t) = A_{22}^{T}\,\boldsymbol{\eta}(t) + A_{12}^{T}\,\underline{r}(t) \tag{2.3-9}$$

and choose

$$\underline{r}(t) = -\,G_e^{T}\,\boldsymbol{\eta}(t) \tag{2.3-10}$$

to force the closed-loop system to exhibit the specified eigenvalues. Details on how to find an appropriate matrix G_e are omitted here. Part 4 of this dissertation describes a numerically very efficient method for determining a matrix G_e which will yield the desired closed-loop poles. Once G_e is known, all three estimator matrices U, V and W can be determined.

The state of the plant can be constructed from the plant output and the estimates.

$$\underline{x}(t) = R_1\,\underline{y}(t) + R_2\,\boldsymbol{\xi}(t) \tag{2.3-11}$$

where

$$R_1^{T} = \left[\, I_p \;\vdots\; G_e \,\right] \qquad \text{and} \qquad R_2^{T} = \left[\, 0 \;\vdots\; I_q \,\right] \tag{2.3-12}$$

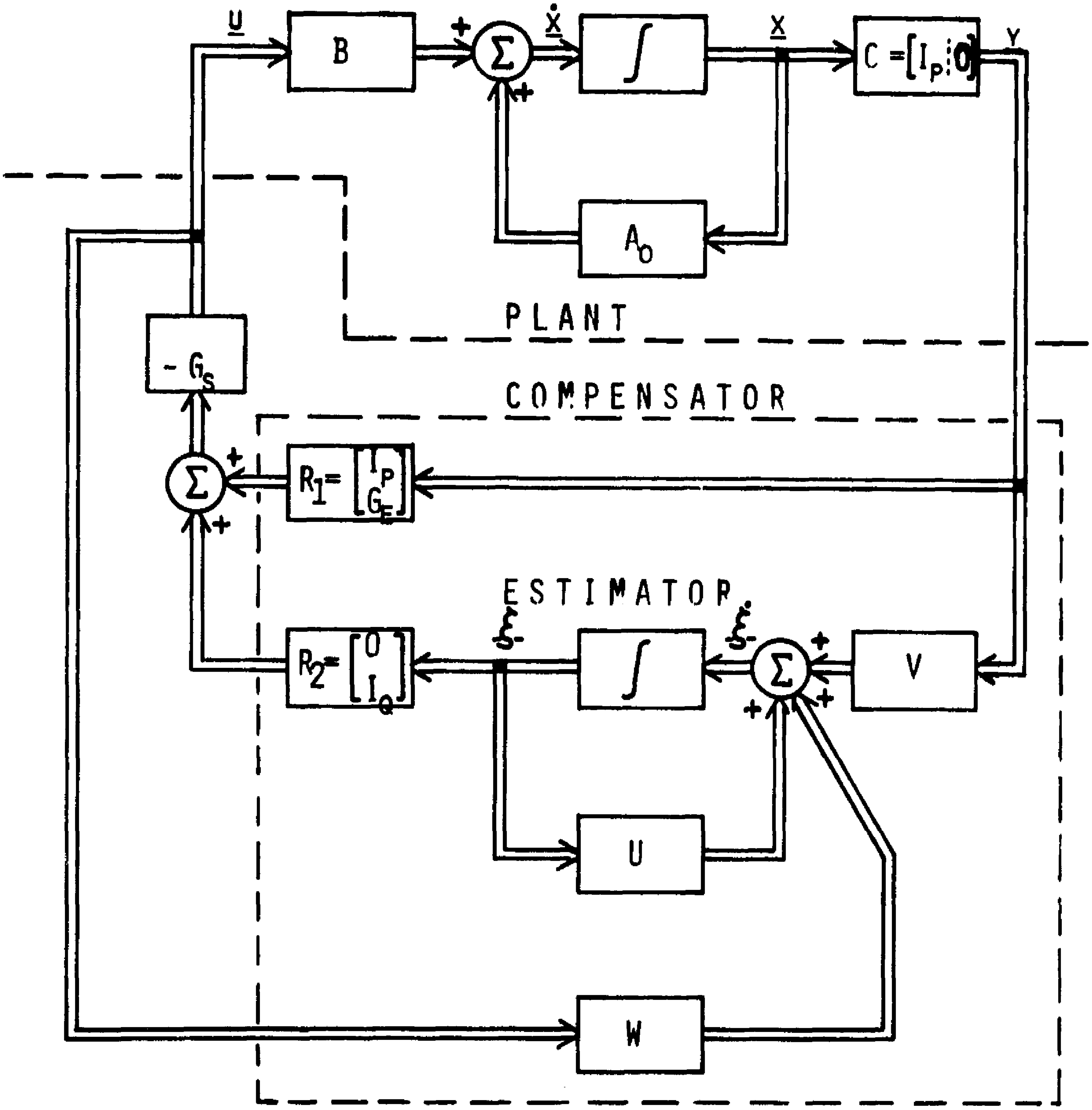

FIGURE 2.3-1

Nominal Closed-Loop System: Compensator Partitioned into Estimator and 'All-state' Feedback Gains - G_s

The structure of plant and estimator is shown in figure 2.3-1.

2) Feedback Gains G_s

q of the $n+q$ closed-loop system eigenvalues are realized by appropriate estimator design. The remaining n eigenvalues are obtained by computing appropriate feedback gains G_s for the nominal plant. For the determination of G_s it does not matter that q of the n states to be fed-back are actually state estimates. Thus the system under consideration is

$$\underline{\dot{x}}(t) = A_0\, \underline{x}(t) + B\, \underline{u}(t) \tag{2.3-13}$$

and the control law to obtain the desired closed-loop poles is

$$\underline{u}(t) = -\, G_s\, \underline{x}(t) \tag{2.3-14}$$

Again the reader is referred to Part 4 for an algorithm on how to compute an appropriate matrix G_s.

3) Compensator Design

The compensator matrices can be constructed from the estimator matrices and the gain matrix G_s by simple block diagram manipulation. Figure 2.3-2 depicts the final result. The diagram has the same structure as that of figure 2.2-1. Equating corresponding expressions yields

$$F = U - W\, G_s\, R_2 \tag{2.3-15a}$$

$$G = V - W\, G_s\, R_1 \tag{2.3-15b}$$

$$H = -\, G_s\, R_2 \tag{2.3-15c}$$

$$J = -\, G_s\, R_1 \tag{2.3-15d}$$

4) Computation of $\kappa = \inf\limits_{M} \|M\|\, \|M^{-1}\|$

This step caused some difficulties. No use could be made of existing techniques, since none seem to be available. It was not possible to

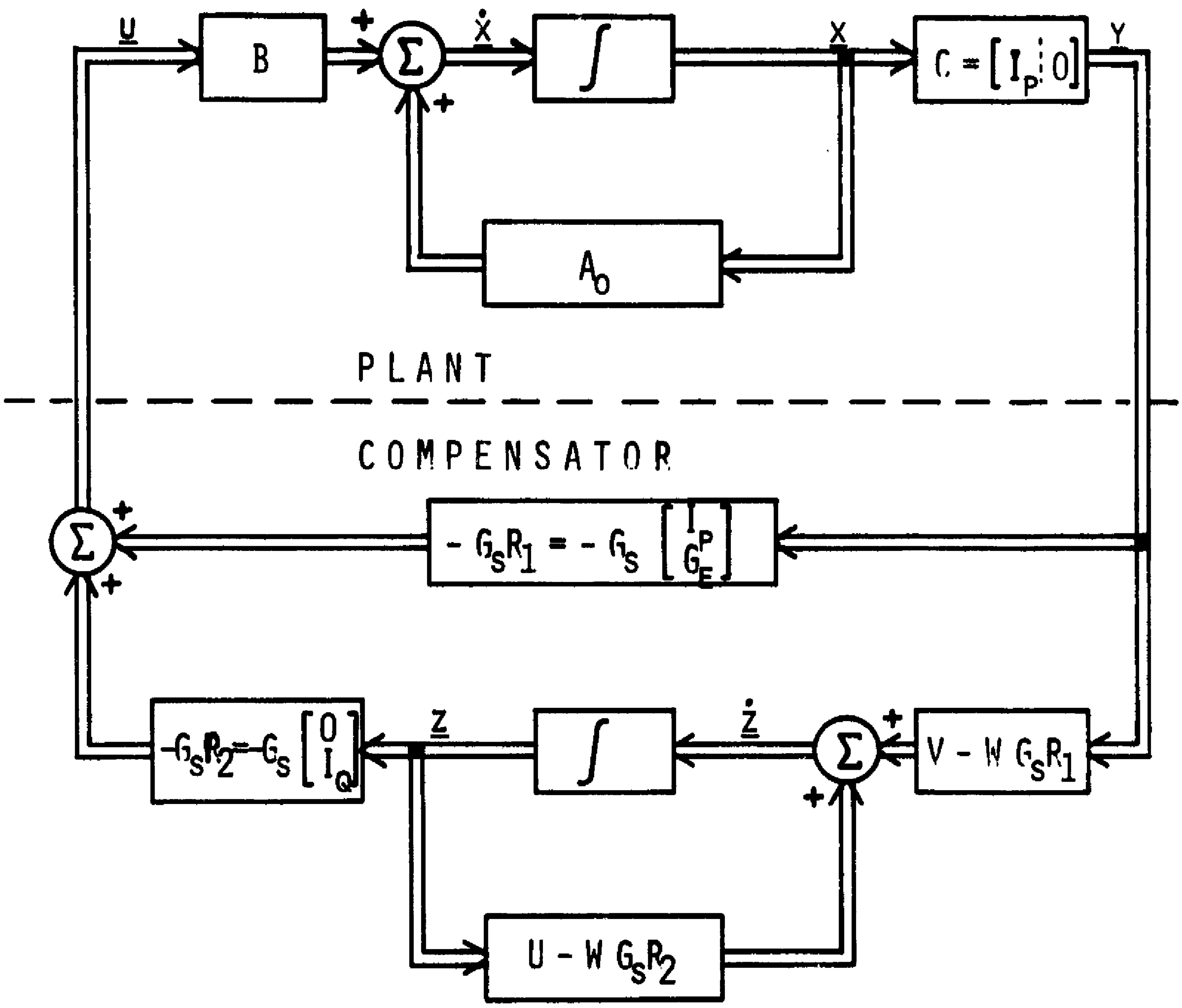

FIGURE 2.3-2

Nominal Closed-Loop System: Derived from Figure 2.3-1 by Block Diagram Manipulation solve the problem for a matrix norm induced by an arbitrary vector norm. An algorithm could be developed to determine $\varkappa$ for matrix norms induced by the 'one' and by the 'infinity' vector norm. The results for the matrix norm induced by the 'infinity' vector norm are outlined here. Full details for both norms are presented in Part 4, section 4.3. The 'infinity' vector norm induces the matrix norm

$$\| M \|_{\infty} = \sup_{\underline{x} \neq 0} \frac{\| M \underline{x} \|_{\infty}}{\| \underline{x} \|_{\infty}} = \max_{k} \sum_{j} | M_{jk} | \tag{2.3-16}$$

i.e., $\| M \|_{\infty}$ is the maximum absolute row sum.

After determining two matrices M and M^{-1} that transform K to diagonal form L proceed as follows in calculating $\varkappa_{\infty} = \inf_{M} \| M \|_{\infty} \| M^{-1} \|_{\infty}$:

α) Define $Q \triangleq M^{-1} \triangleq \left[\underline{q}_1, \underline{q}_2, \ldots, \underline{q}_{(n+q)} \right]^T$. Obtain the matrix $Q_\beta = M_\beta^{-1}$ by normalizing the rows of M^{-1}, i.e.,

$$\sum_{k} | q_{\beta_{jk}} | = 1 = \sum_{k} \frac{| q_{jk} |}{\sum_{k} | q_{jk} |} \qquad \text{for} \quad j=1,(n+q) \tag{2.3-17}$$

β) Scale $R \triangleq M$ appropriately to yield $R_\beta = M_\beta$, where $M_\beta M_\beta^{-1} = I$ and $R_\beta = \left[\underline{r}_{\beta 1}, \underline{r}_{\beta 2}, \ldots, \underline{r}_{\beta (n+q)} \right]^T$.

γ) Then $\varkappa_{\infty}$ is given by

$$\varkappa_{\infty} = \max_{j} \sum_{k} | r_{\beta_{jk}} | \tag{2.3-18}$$

5) Iteration on the Sensitivity Measure f_s

Once $\varkappa$ is determined it is not difficult to compute the sensitivity measure f_s as given by equation (2.2-30). However, the main task is to minimize f_s. One solution way would be to determine a gradient of f_s with respect to the elements of the compensator matrices, since they can be influenced directly. Then the compensator elements could be changed until the gradient of f_s is zero or very small. Since f_s is not explicitly dependent on the compensator elements such a gradient would have to be synthesized from perturbing the compensator elements. Usually the total number of elements of the compensator matrices is quite large. Thus, this approach would create a dimensionality problem. Predicting the motion of the closed-loop eigenvalues would be very complicated, too. Hence, another approach is chosen.

This new approach synthesizes the gradient of f_s with respect to perturbations of the closed-loop eigenvalues. The compensator matrices corresponding to the perturbed eigenvalues are easily calculated by proceeding through steps 1 to 4 above. It has to be pointed out that this approach restricts the degrees of freedom in design remarkably, because the compensator designed in steps 1 to 4 will always be designed in the same way. No use is made of the multitude of other compensator designs for the same set of eigenvalues. Such a use would be possible if the above approach were used. However, then the problem would no longer be computationally tractable. To restore at least some of the lost freedom the following observation is taken into consideration. Simultaneous multiplication of matrix G by some scalar α_0 and division of matrix H by α_0 does neither change the compensator output nor the closed-loop eigenvalues, but it has a large influence on the system eigenvectors and thus on $\mathbf{x}$. Hence, f_s is also considered an implicit function of α_0 and the gradient is determined accordingly.

PART 3

NUMERICAL EXAMPLES

3.1 Introduction

The theory and computational aspect of the compensator design for low-sensitivity systems was presented in the previous part. A computer program, called COMPDES, to mechanize the design procedure was written in FORTRAN IV. Several numerical examples were computed and three of them will be described in the following to show the success and limitations of the method.

3.2 Program Outline

COMPDES can be broken down into two major sections. The first section consists of a gradient procedure which tries to increase the stability of a system by computing appropriate feedback gains. The thus obtained closed-loop system is tested for stability with respect to the maximum possible parameter variations. The program terminates if stability can be guaranteed. If not, the program proceeds to program section two. Section one can be by-passed.

Program section two carries out the actual compensator design and sensitivity reduction. The compensator is composed of a state estimator and a set of 'all-state' feedback gains. The sensitivity function is minimized by means of the Davidon function minimization method. The gradient of the sensitivity function with respect to the eigenvalues and α_0 is required for the Davidon method and is synthetically generated. Thus the gradient may not be computed to be zero where the function actually has a minimum. To avoid useless numerical oscillation about such

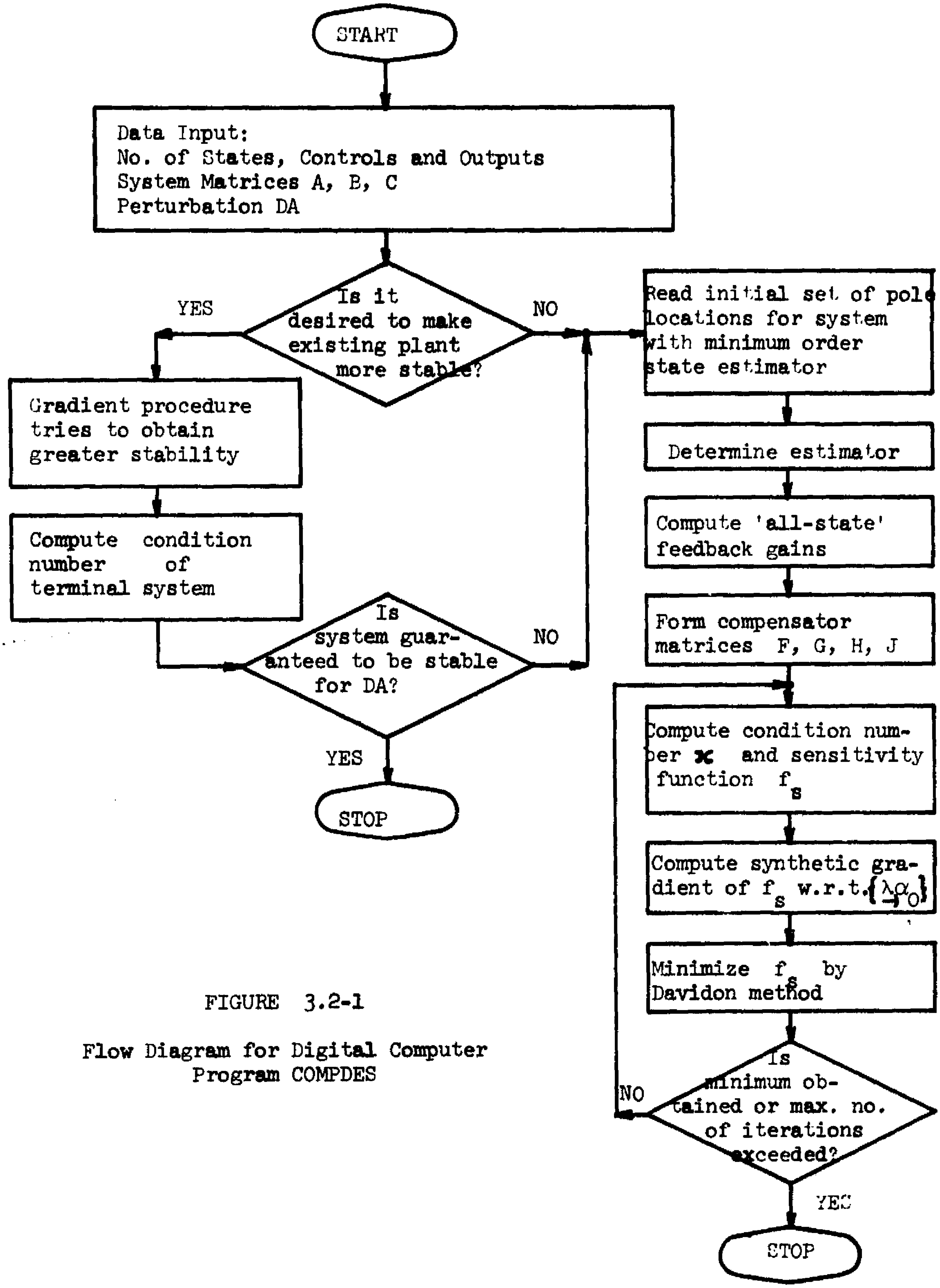

FIGURE 3.2-1

Flow Diagram for Digital Computer Program COMPDES a minimum, the total number of iterations is limited.

A flow chart of the COMPDES computer program is depicted in figure 3.2-1.

3.3 Explanation of the Computer Print-Out

The first page of each example shows the input data, and gives a listing of the eigenvalues of the system matrix A. The eigenvalues are called 'Roots' in the print-out.

The second page gives a print-out of the initial compensator matrices. FJ, FH, FG and FF denote the compensator matrices J, H, G and F. These matrices are computed for a given set of closed-loop poles. This set of poles is again denoted 'Roots' and the print-out follows that of the compensator matrices.

The line following the poles of the initial compensator design shows the value of α_0 by which the compensator input matrix G and output matrix H are scaled (see Part 2, section 2.3) and the condition number $\varkappa$. These two values are repeated in the next line together with the least stable pole denoted 'ROOT1' and the value of the sensitivity function called 'FUNCTION VALUE'. The following line contains the same quantities, but for the final compensator design. The display of the matrices and the eigenvalues of the final design conclude the second page.

3.4 Examples

3.4.1 2^{nd} order system

3.4.2 3^{rd} order system

3.4.3 4^{th} order system

3.4.1 2nd Order System

```
STATES    INPUTS    OUTPUTS   COMP-ORD  IDELR     ISTOP     IWAIT     NUL
NS # 2    NC # 1    NF # 1    NFF # 0       0         0         0         0

SYSTEM MATRIX  A.
   0.50000000D 01   -0.70000000D 01    0.10000000D 01   -0.30000000D 01

MAXIMUM ABSOLUTE CHANGE  DA  OF THE ELEMENTS OF THE SYSTEM MATRIX  A
   0.50000000D 00    0.10000000D 01    0.30000000D 00    0.50000000D 00

CONTROL INPUT MATRIX B
   0.0               0.10000000D 01

OUTPUT MATRIX  C
   0.10000000D 01    0.0

ACCEPTABLE LARGEST REAL PART OF THE EIGENVALUES FOR THE WORST CASE OF
PARAMETER UNCERTAINTY HAS TO BE LESS THAN  AC #  -0.20000
EIGENVALUES OF NOMINAL SYSTEM ARE KEPT TO THE LEFT OF  ACC #  -7.00000


ROOTS      REAL PART          IMAG. PART
        0.40000000D 01       0.0
       -0.20000000D 01       0.0


DETERMINE COMPENSATOR OF ORDER  NFF #  1
AND ITERATE ON CONDITION-NUMBER.
```

```
COMPENSATOR DESIGN - INITIAL VALUES.   ISHIFT # 1

DIRECT FEEDBACK MATRIX  FJ
   0.2885714D 03

COMPENSATOR OUTPUT MATRIX  FH
  -0.2600000D 02

COMPENSATOR INPUT MATRIX  FG
   0.3502857D 03

COMPENSATOR MATRIX  FF
  -0.4600000D 02

ROOTS      REAL PART            IMAG. PART
       -0.1200000D 02        0.3600000D 02
       -0.1200000D 02       -0.3600000D 02
       -0.2000000D 02        0.0

ALO #    1.000000COND.-NUMBER #   22.855462

ALO #     1.00000  ROOT1 #  -0.1200000D 02  COND.-NUMBER #   0.2285546D 02  FUNCTION VALUE #    0.8789

ALO #     0.83440  ROOT1 #  -0.1886486D 02  COND.-NUMBER #   0.2474050D 02  FUNCTION VALUE #    0.7310

THE BELOW COMPENSATOR GUARANTEES STABILITY ONLY FOR A TOTAL
UNCERTAINTY OF        0.7472[illegible]36

COMPENSATOR DESIGN - FINAL VALUES.

DIRECT FEEDBACK MATRIX  FJ
   0.3595105D 03

COMPENSATOR OUTPUT MATRIX  FH
  -0.4705056D 02

COMPENSATOR INPUT MATRIX  FG
   0.3500695D 03

COMPENSATOR MATRIX  FF
  -0.5859061D 02

ROOTS      REAL PART            IMAG. PART
       -0.1886490D 02        0.3638530D 02
       -0.1886490D 02       -0.3638530D 02
       -0.1886090D 02        0.0
```

3.4.2 3^{rd} Order System

```
STATES     INPLTS     OLTPLTS    CCMF-CRC  ICELR      ISTCF      IMATJ      NU1
NS = 3     NC = 1     NF = 2     NFF = 0      0          0          C          C

SYSTEM MATRIX A,
 -C.4CCCCCCD C1    -C.2CC0000D 01     0.1000000C 01     0.500C0C0C CC
 -C.3CCCCCCD C1     C.2CCC0000 01     0.2500000C 01     0.8C000CCC C0
  C.2CC^CCCD C1

MAXIMUM ABSOLUTE CHANGE  DA  CF THE ELEMENTS CF THE SYSTEM MATRIX  A
  C.4CCCCCCD CC     C.2CCC0000 00     0.1000000C 0^     0.5C000C0C-01
  C.3CCCCCCD CC     C.2CCC0C0D 00     0.2500000C 00     0.800CCC0C-C1
  C.2CCCCCCD CC

CONTROL INPUT MATRIX B
 -C.2CCCCCCD C1     C.1CCC0000 01     0.3000000C 01

OUTPUT MATRIX  C
  C.1CCCCCCD C1     C.C               0.0               0.0
  C.1CCCCCCD C1     C.0

ACCEPTABLE LARGEST REAL PART CF THE EIGENVALUES FCR THE WORST CASE OF
PARAMETER UNCERTAINTY HAS TO BE LESS THAN  AC =  -0.20000
EIGENVALUES OF NOMINAL SYSTEM ARE KEPT TC THE LEFT CF  ACC =  -2.CCCCC

ROOTS     REAL PART            IMAG. PART
       -C.472C68D C1        0.0
       -C.2476990 C1        0.0
        0.2397680 C1        0.0

DETERMINE COMPENSATOR CF ORDER  NFF =  1
AND ITERATE ON CONDITICN-NUMBER.
```

```
COMPENSATOR DESIGN - INITIAL VALUES,    ISHIFT =  1

DIRECT FEEDBACK MATRIX  FJ
  -C.12311670 C1     -C.15548740 02

COMPENSATOR OUTPUT MATRIX  FH
  -C.28345640 C1

COMPENSATOR INPUT MATRIX  FS
  -C.28275170 C1      C.66718610 01

COMPENSATOR MATRIX  FF
  -C.19135890 C1

ROOTS      REAL PART             IMAG. PART
       -C.5CCCCCC0 C1          C.C
       -C.4CCCCCC0 C1          0.1000000 01
       -C.4CCCCCC0 C1         -0.1000000 01
       -C.3CCCCCC0 C1          C.C

ALO =    1.CCCCCCCCND.-NUMBER =  222.909003

ALO =     1.CCCCC  ROOT1 =  -0.30000000 01  CCND.-NUMBER =   0.22290900 C3  FUNCTION VALUE =    C.9937

ALO =     1.37C8C  ROOT1 =  -0.28945700 01  CCND.-NUMBER =   0.97205210 C2  FUNCTION VALUE =    0.9872

THE BELOW COMPENSATOR GUARANTEES STABILITY ONLY FOR A TOTAL
UNCERTAINTY OF        C.C2776448

COMPENSATOR DESIGN - FINAL VALUES.

DIRECT FEEDBACK MATRIX  FJ
  -C.16916220 C1     -C.15205810 02

COMPENSATOR OUTPUT MATRIX  FH
  -C.22696400 C1

COMPENSATOR INPUT MATRIX  FS
  -C.43288230 C1      C.83977990 01

COMPENSATOR MATRIX  FF
  -C.19685C20 C1

ROOTS      REAL PART             IMAG. PART
       -0.7583120 C1          0.0
       -C.4161680 C1          0.2419760 01
       -C.4161680 C1         -C.2419760 01
       -0.2884570 C1          C.0
```

3.4.3 4^{th} Order System

```
STATES     INPUTS     OUTPUTS    COMP-ORD   IDELR    ISIOP    IMATJ    NUL
NS = 4     NC = 1     NF = 2     NFF = 0      0        0        0        0

SYSTEM MATRIX  A.
  -0.50000000D 01     0.10000000D 01    -0.20000000D 01     0.50000000D 00
   0.0               -0.12000000D 02     0.0                0.10000000D 01
  -0.25000000D 01     0.0                0.10000000D 01     0.10000000D 01
   0.10000000D 01     0.80000000D 00    -0.40000000D 01    -0.16000000D 02

MAXIMUM ABSOLUTE CHANGE  DA  OF THE ELEMENTS OF THE SYSTEM MATRIX  A
   0.20000000D 00     0.50000000D-01     0.10000000D 00     0.20000000D-01
   0.10000000D-01     0.30000000D 00     0.0                0.50000000D-01
   0.30000000D 00     0.10000000D-01     0.10000000D 00     0.10000000D 00
   0.10000000D 00     0.40000000D-01     0.20000000D 00     0.40000000D 00

CONTROL INPUT MATRIX B
   0.50000000D 00    -0.20000000D 01     0.0                0.10000000D 01

OUTPUT MATRIX  C
   0.10000000D 01     0.0                0.0                0.0
   0.0                0.10000000D 01     0.0                0.0

ACCEPTABLE LARGEST REAL PART OF THE EIGENVALUES FOR THE WORST CASE OF
PARAMETER UNCERTAINTY HAS TO BE LESS THAN  AC =  -1.00000
EIGENVALUES OF NOMINAL SYSTEM ARE KEPT TO THE LEFT OF  ACC =  -4.00000

ROOTS      REAL PART              IMAG. PART
        -0.1180370D 02         0.0
        -0.5784650D 01         0.0
         0.1560130D 01         0.0
        -0.1597950D 02         0.0

DETERMINE COMPENSATOR OF ORDER  NFF =  2
AND ITERATE ON CONDITION-NUMBER.
```

```
COMPENSATOR DESIGN - INITIAL VALUES.    ISHIFT =  1

DIRECT FEEDBACK MATRIX  FJ
  -0.8531160D 01    -0.827[illegible] 01

COMPENSATOR OUTPUT MATRIX  FH
   0.7179732D 02     0.7275[illegible]720 01

COMPENSATOR INPUT MATRIX  FG
   0.1471371D 05     0.1421173D 05     0.7100386D 03     0.6960720D 03

COMPENSATOR MATRIX  FF
  -0.1245935D 04    -0.1060051D 03    -0.5903866D 02    -0.1950870D 02

ROOTS      REAL PART            IMAG. PART
        -0.1800000D 02        0.1000000D 02
        -0.1800000D 02       -0.1000000D 02
        -0.1200000D 02        0.0
        -0.8000000D 01        0.6000000D 01
        -0.8000000D 01       -0.6000000D 01
        -0.5000000D 01        0.0

ALO =     1.000000COND.-NUMBER =11373.949626

ALO =      1.00000   ROOT1 =  -0.5000000D 01   COND.-NUMBER =   0.1137395D 05   FUNCTION VALUE =    0.9999

ALO =      0.09851   ROOT1 =  -0.7013092D 01   COND.-NUMBER =   0.2517101D 04   FUNCTION VALUE =    0.9984
     1

THE BELOW COMPENSATOR GUARANTEES STABILITY ONLY FOR A TOTAL
UNCERTAINTY OF       0.00227429

COMPENSATOR DESIGN - FINAL VALUES.

DIRECT FEEDBACK MATRIX  FJ
  -0.1016298D 04    -0.9905536D 03

COMPENSATOR OUTPUT MATRIX  FH
   0.7676018D 03     0.6859127D 02

COMPENSATOR INPUT MATRIX  FG
   0.1971690D 04     0.1914661D 04     0.2363959D 02     0.2272584D 02

COMPENSATOR MATRIX  FF
  -0.1496160D 04    -0.1129838D 03    -0.1734208D 02    -0.1591427D 02

ROOTS      REAL PART            IMAG. PART
        -0.1872340D 02        0.9395950D 01
        -0.1872340D 02       -0.9395950D 01
        -0.7200200D 01        0.5714430D 01
        -0.7200200D 01       -0.5714430D 01
        -0.1226610D 02        0.0
        -0.7003090D 01        0.0
```

3.5 Discussion of the Results

All three examples were chosen arbitrarily and their results must be interpreted differently. The uncompensated systems are all unstable and at least the second order system cannot be stabilized by static feedback.

The first example minimizes the sensitivity function by essentially maximizing $|Re(\lambda_{max})/\varkappa|$. The final condition number $\varkappa_f = 24.74$ is slightly greater than the initial value $\varkappa_i = 22.85$. A result of this type could be expected, as already indicated in section 2.2. A check on the sensitivities of the initial and the final closed-loop system design was done by successively perturbing each of the diagonal elements of the overall systems. The perturbation had the value +1. Although the absolute changes in pole locations of the final design were slightly larger than those of the initial design, the final design maintained its 50% higher stability margin over the initial design.

The minimization of the sensitivity function in the second example is mostly achieved by decreasing the condition number from $\varkappa_i = 222.9$ to $\varkappa_f = 97.2$. Again the diagonal elements were one by one perturbed by +1. This time the changes in pole locations was markedly different for the two designs. The absolute shifts of the poles of the initial design were 2 to 3 times larger than the corresponding shifts of the poles of the final design. Both designs remained stable for the introduced perturbations.

Although the sensitivity of the third example is reduced by more than a factor of 4, this is not enough to actually obtain a low-sensitivity final design. Perturbation tests showed that both designs are extremely sensitive. Since the absolute changes in pole location of both designs are approximately the same no superiority of the final design could be established. The extreme sensitivity of either system was best demonstrated when adding +1 to the (2,2)-element of the overall systems. The (2,2) element of the initial design has the value +1643.1 and the corresponding element of the final design the value +1969.1. The perturbation caused both systems to become violently unstable, each getting an eigenvalue in the vicinity of $s = +20$.. If a less sensitive system is desired several different initial designs should be tried out.

It is quite interesting to compare the permissible uncertainties of the three systems. The values are .74, .027 and .0022 for the 2^{nd}, 3^{rd} and 4^{th} order plant. Since all three plants are different these numbers can not be compared directly. However, they roughly indicate the order of magnitude of the uncertainties permissible for the three different order systems. In general it can be said, the higher the plant order, the smaller the permissible uncertainty. Heuristically this can be explained as follows: the influence of an uncertainty δ on the characteristic equation of an n^{th} order system can be estimated to $\delta \cdot |a_{ij_{max}}|^{n-1}$ where $a_{ij_{max}}$ is the element of largest magnitude of the plant matrix. This estimate indicates an exponentially growing influence of a parameter variation δ.

None of the examples permits the specified parameter uncertainties. The uncertainties were intentionally chosen to be rather high, forcing the computation to either find a local sensitivity minimum or terminate because of too many iterations.

PART 4

TWO GENERAL PURPOSE NUMERICAL ALGORITHMS

4.1 Introduction

To solve the problems posed in the preceding chapters the need arose for an efficient pole-placement algorithm and an algorithm to determine the condition number, $\varkappa = \inf_{M} \| M \| \, \| M^{-1} \|$ (where $A = M L M^{-1}$, with $L = \text{diag}\left[\text{eigenvalues of } A\right]$) for at least some matrix norm, $\| \cdot \|$ induced by an absolute vector norm.

Investigation of the literature for pole-placement algorithms showed that, although a number of algorithms were developed previously[3-6], they all appeared to be rather complex and lengthy. The literature search uncovered another fact, namely, that the solution of the algebraic matrix Riccati equation still poses problems. Although pole-assignment and Riccati equation do not appear to have much in common, it will be shown later that a good pole-placement algorithm can be of great value for the solution of the steady-state Riccati equation.

Basically the algebraic matrix Riccati equation can be solved in two ways: one is the use of successive approximation methods, the other is the backwards integration of the time-varying matrix Riccati equation until steady-state behavior is obtained. Backward integration can be performed by direct numerical integration (e.g., Runge-Kutta or Hamming--predictor-corrector-method) or by the automatic synthesis program (ASP)[7] matrix iterative procedure; both procedures require disproportionally long computation times, especially for low order systems (for 1st and 2nd order systems up to 1.5 minutes for a Fortran H compiled program on the IBM 360/50). Several iterative procedures are available for the solution of the algebraic matrix Riccati equation. In order to converge to a positive definite solution, most algorithms, i.e., Kleinman's[9] method, requires such an initial guess P_0 of the Riccati matrix as to ensure stability of the closed-loop system. Obtaining a stabilizing guess P_0 is generally considered difficult, especially for higher order systems. Man[8] claims to have developed an algorithm which is not critically dependent on the choice of the starting matrix P_0. However, this author's experience with Man's algorithm was, that good convergence was achieved only for the examples presented in Man's paper. Examples, in which the system to be optimized was very unstable, converged very slowly or not at all.

Since Kleinman's algorithm is very efficient and exhibits quadratic convergence, it would be a very valuable tool in combination with some algorithm that automatically generates an appropriate starting matrix P_0, or an appropriate feedback gain matrix G_0. Ash[2] pointed out the usefulness of a pole-placement algorithm to generate a valid initial feedback gain matrix G_0. Ash used results from the state-estimation theory to place the closed—loop poles of a controllable, single-input system arbitrarily along the real-axis of the complex s-plane. The same results of the state estimation theory will be utilized in the following sections to develop a general pole-placement algorithm for multi-input systems, allowing arbitrary pole assignment in the whole s-plane, including multiple real and complex eigenvalues.

The algorithm consists of three parts. In the first part it is shown how to place poles arbitrarily for a single-input controllable system. The second step describes a method of converting a multi-input controllable system into a pseudo single-input controllable system. The third step eliminates the problems involved with multi-input controllable systems, that are not immediately transferable to single-input controllable systems, and problems occurring if some of the specified closed-loop poles coincide with open-loop poles.

The problem of finding the condition number $\varkappa = \inf_M \|M\| \|M^{-1}\|$ for some matrix norms induced by absolute vector norms, does not seem to be dealt with in the literature at all. In the last section of this part a simple algorithm will be presented to obtain the $\inf_M \|M\| \|M^{-1}\|$ for the 'infinity' and the 'one' norm.

4.2 Pole Placement and Initialization of the Iterative Riccati Equation

4.2.1 Relation between all-state feedback gains and Kleinman's iterative solution method for the Riccati equation.

Let the controllable linear time-invariant system be described by

$$\underline{\dot{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t) \qquad (4.2.1\text{-}1)$$

where A is a (nxn) matrix and B a (nxm) matrix. All-state feedback of the form

$$\underline{u}(t) = -G_0\,\underline{x}(t) \qquad (4.2.1\text{-}2)$$

yields the closed-loop system

$$\underline{\dot{x}}(t) = (A-BG_0)\,\underline{x}(t) \qquad (4.2.1\text{-}3)$$

where the eigenvalues of $(A-BG_0)$ can be arbitrarily assigned[3-5]. Thus, if the eigenvalues of $(A-BG_0)$ are pre-selected, the problem is to determine a G_0 which yields the desired eigenvalues. The same problem arises in Kleinman's iterative solution scheme for the algebraic matrix matrix Riccati equation.

Let the cost functional for equation (1) be given by

$$J(u) = \int_0^{\infty} \tfrac{1}{2}(\underline{x}^T(\tau)\ Q\ \underline{x}(\tau) + \underline{u}^T(\tau)\ R\ \underline{u}(\tau))\ d\tau \qquad (4.2.1\text{-}4)$$

where Q and R are (n*n) and (m*m) positive definite matrices, respectively. The control law, that minimizes (4) is well known[1,10,11] and is given by

$$\underline{u}^*(t) = -R^{-1}\ B^T\ P\ \underline{x}(t) = -\ G_0^*\ \ \underline{x}(t) \qquad (4.2.1\text{-}5)$$

where P is the unique positive definite solution of the algebraic matrix Riccati equation

$$A^T\ P + P\ A - P\ B\ R^{-1}\ B^T\ P + Q = 0 \qquad (4.2.1\text{-}6)$$

To solve (6) for the matrix P Kleinman[9] suggested a successive approximation method. The $(i+1)^{st}$ iteration of the method can be written as

$$(A-BR^{-1}B^TP_i)^TP_{i+1} + P_{i+1}(A-BR^{-1}B^TP_i) = -P_iBR^{-1}B^TP_i-Q \qquad (4.2.1\text{-}7)$$

Kleinman showed that the method enjoys quadratic convergence and will yield a unique positive definite solution if the starting matrix P_0 is chosen such that $(A-BR^{-1}B^TP_0) = (A-BG_0)$ is a stable matrix. Equation (7) can be re-written as:

$$(A-BG_i)^TP_{i+1} + P_{i+1}(A-BG_i) = -G_i^TRG_i-Q \qquad (4.2.1\text{-}8)$$

where

$$G_i = R^{-1}\ B^T\ P_i \qquad (4.2.1\text{-}9)$$

Thus, the G_0 computed for a set of stable eigenvalues of the closed-loop system (3) can be used to initialize the Kleinman iterative procedure.

4.2.2 Arbitrary pole-assignment for single-input systems

Let the control input matrix B be the vector $\underline{b}$. Then equation (4.2.1-1) becomes

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + \underline{b}\,u \tag{4.2.2-1}$$

System (1) is assumed to be single-input controllable[12], i.e.,

$$\text{Rank}\left[\,\underline{b},\ A\,\underline{b},\ \ldots\ ,\ A^{n-1}\,\underline{b}\,\right] = n \tag{4.2.2-2}$$

Let the pair $(F^T, \underline{h})$ describe the n^{th} order single-output observable system

$$\dot{\underline{z}}(t) = F\,\underline{z}(t) \tag{4.2.2-3}$$

$$u(t) = \underline{h}^T\,\underline{z}(t) \tag{4.2.2-4}$$

System ((3),(4)) is observable if

$$\text{Rank}\left[\,\underline{h},\ F^T\underline{h},\ \ldots,\ (F^T)^{n-1}\,\underline{h}\,\right] = n \tag{4.2.2-5}$$

Let T be a non-singular similarity transformation such that

$$\underline{x}(t) = T\,\underline{z}(t) \tag{4.2.2-6}$$

Substituting expressions (4) and (6) in equation (1) yields

$$T\,\dot{\underline{z}}(t) = AT\,\underline{z}(t) + \underline{b}\,\underline{h}^T\,\underline{z}(t) \tag{4.2.2-7}$$

and since $\dot{\underline{z}}(t) = F\,\underline{z}(t)$ it follows that

$$T\,F\,\underline{z}(t) = AT\,\underline{z}(t) + \underline{b}\,\underline{h}^T\,\underline{z}(t) \tag{4.2.2-8}$$

and thus

$$TF - AT = \underline{b}\,\underline{h}^T \qquad (4.2.2\text{-}9)$$

Equation (9) will have a unique solution[13,14] for T, if F and A have no common eigenvalues. Furthermore, Luenberger[14] showed that T will be invertible if $\underline{b}$ renders $(A, \underline{b})$ controllable and $\underline{h}^T$ renders $(F^T, \underline{h})$ observable. But the latter two conditions are fulfilled by assumption. Thus, since the resulting T is non-singular (as assumed in eq. (6)), equation (9) can be transformed to

$$TFT^{-1} = A + \underline{b}\,\underline{h}^T T^{-1} = A - \underline{b}\,\underline{g}_0{}^T \qquad (4.2.2\text{-}10)$$

where

$$\underline{g}_0{}^T = -\,\underline{h}^T\,T^{-1} \qquad (4.2.2\text{-}11)$$

Equation (10) simply states that, since F and $(A - \underline{b}\,\underline{g}_0{}^T)$ are similar, they have the same eigenvalues. Hence, by choosing a matrix F, which has the specified closed-loop eigenvalues, and an appropriate vector $\underline{h}^T$, equation (9) can be solved for T.

For mathematical and computational simplicity it would be advantageous for F to be a diagonal matrix. Theoretically this choice is always possible, even in the case of complex eigenvalues, which have to occur in conjugate complex pairs. However, the pure diagonal form is desirable only for real eigenvalues. A complex diagonal matrix F would not only complicate the numerical computations considerably, but also result in complex gains $\underline{g}_0$, since both, T and T^{-1} would be complex. Complex feedback gains $\underline{g}_0$ cannot be implemented in practical designs. By restraining F to be real, but still desiring the capability of complex eigenvalues, F can no longer be purely diagonal; it will have non-zero values on the super- and subdiagonal for complex eigenvalues.

Thus, T may be computed in the following way:

1. Select a vector $\underline{h}^T$ with no zero elements
2. Arrange the set of desired closed-loop poles in such a way that F becomes:

$$F = \begin{bmatrix} f_1 & 0 & 0 & & & & & \\ 0 & f_2 & 0 & & & & \bigcirc & \\ 0 & & \ddots & f_{i-1} & & & & \\ & & & & f_i & \alpha_i & & \\ & & & & -\alpha_i & f_i & & \\ & \bigcirc & & & & & f_{i+2} & \\ & & & & & & & \ddots \; f_n \end{bmatrix} \tag{4.2.2-12}$$

where f_i are the real and α_i the imaginary parts of the eigenvalues.

3. Solve equation (9), $TF - AT = \underline{b}\,\underline{h}^T$ for T.
4. Compute $\underline{g}_0$ via equation (11), $\underline{g}_0 = -\underline{h}^T\, T^{-1}$.

The simplest way to proceed at step 3 is to solve equation (9) sequentially.

Define:

$$T = \left[\underline{t}_1, \underline{t}_2, \ldots, \underline{t}_n\right] \tag{4.2.2-13}$$

and:

$$\underline{b}\,\underline{h}^T = S = \left[\underline{s}_1, \underline{s}_2, \ldots, \underline{s}_n\right] \tag{4.2.2-14}$$

Then, for any real eigenvalue f_r of F, equation (9) can be broken down to

$$(f_r I - A)\,\underline{t}_r = \underline{s}_r \qquad \forall\, r \quad \text{with } f_r \text{ real} \tag{4.2.2-15}$$

For any pair of complex eigenvalues $f_i \pm j\,\alpha_i$, equation (9) yields

$$\left[(f_i I-A)^2 + \alpha_i^{\,2} I\right] \underline{t}_i = (f_i I-A)\underline{s}_i + \alpha_i \underline{s}_{i+1} \tag{4.2.2-16a}$$

$$\underline{t}_{i+1} = -\frac{1}{\alpha_i}\,\underline{s}_i + \frac{1}{\alpha_i}\,(f_i I-A)\underline{t}_i \tag{4.2.2-16b}$$

Hence, all n column vectors t_j, $j=1,n$, of the similarity transformation T can be obtained by successively solving equations (15) and (16), respectively, via Guassian elimination or matrix inversion.

Remarks to steps 2 and 3:

If it is desired to specify multiple real or complex closed-loop eigenvalues F will have the following form:

$$F=\begin{bmatrix}
f_1 & & & & & & & & & & & & & & & \\
 & \ddots & & & & & & & & & & & & & & \\
 & & f_{i-1} & & & & & & & & & & & & & \\
 & & & f_{i_1} & 1 & & & & & & \mathbf{0} & & & & & \\
 & & & & f_{i_2} & \ddots & 1 & & & & & & & & & \\
 & & & & & \ddots & f_{i_\nu} & & & & & & & & & \\
 & & & & & & & f_{K_1} & \alpha_K & 1 & & & & & & \\
 & & & & & & & -\alpha_K & f_{K_2} & 0 & 1 & & & & & \\
 & & & & & & & & & f_{K_3} & \alpha_K & 1 & & & & \\
 & & & & & & & & & -\alpha_K & f_{K_4} & 0 & & & & \\
 & & & & & & & & & & & \ddots & & 1 & & \\
 & & & \mathbf{0} & & & & & & & & & f_{K_{\mu-1}} & \alpha_K & & \\
 & & & & & & & & & & & & -\alpha_K & f_{K_\mu} & & \\
 & & & & & & & & & & & & & & \ddots & f_n
\end{bmatrix} \tag{4.2.2-17}$$

The vectors $\underline{t}_i$ of the transformation matrix " corresponding to multiple real eigenvalues $f_{i_{1,\nu}}$ are computed from

$$(f_{i_1} I - A)\underline{t}_i = \underline{s}_i \qquad (4.2.2\text{-}18a)$$

and

$$(f_{i_j} I - A)\underline{t}_{i+j-1} = \underline{s}_{i+j-1} - \underline{t}_{i+j-2} \qquad j = 2, \nu \qquad (4.2.2\text{-}18b)$$

The vector $\underline{t}_{k+\ell-1}$, $\ell = 1, \mu$, corresponding to the multiple complex eigenvalue $f_{k,\ell} (\pm) j\alpha_k$ is calculated from

$$\left[(f_{k1} I - A) + \alpha_k{}^2 I\right] \underline{t}_k = (f_{k1} I - A)\underline{s}_k + \alpha_k \underline{s}_{k+1} \qquad (4.2.2\text{-}19a)$$

$$\underline{t}_{k+1} = -\frac{1}{\alpha_k} \cdot \underline{s}_k + \frac{1}{\alpha_k}(f_{k_1} I - A)\underline{t}_k \qquad (4.2.2\text{-}19b)$$

and

$$\left[(f_{k,2(j-1)+1} I - A)^2 + \alpha_k I\right]^2 \underline{t}_{k+2(j-1)} = (f_{k,2(j-1)} I - A)(\underline{s}_{k+2(j-1)} - \underline{t}_{k+2(j-2)-2}) +$$

$$+ \alpha_k(\underline{s}_{k+2(j-1)+1} - \underline{t}_{k+2(j-2)-1}) \qquad (4.2.2\text{-}19c)$$

$$\underline{t}_{k+2(j-1)+1} = -\frac{1}{\alpha_k} (\underline{s}_{k+2(j-1)} - \underline{t}_{k+2(j-1)-2}) +$$

$$+ \frac{1}{\alpha_k} (f_{k,2(j-1)+1} I - A)\underline{t}_{k+2(j-1)} \qquad (4.2.2\text{-}19d$$

Formulas (19c) and (19d) are valid for $j=2, (\frac{\mu}{2} - 1)$.

The second index j of $f_{k,j}$ defines that position of $f_{k,j}$ within its associated Jordan block and is needed to properly identify the corresponding vectors $\underline{t}_{k+j-1}$ and $\underline{s}_{k+j-1}$. Clearly, $f_{k,1} = f_{k,2} = \ldots = f_k$.

Although equation sets (18) and (19) look rather involved they are not too difficult to program. Together with (15) and (16) they produce an algorithm which is numerically very efficient.

4.2.3 Conversion of a multi-input system to a pseudo single-input system

In the previous section the algorithm for arbitrary pole-assignment in controllable single-input system was presented. As it stands the algorithm will work and yield an invertible matrix T, for multi-input systems only in special cases, namely, if any one column $\underline{b}_i$, $i=1,m$, of the input matrix B renders the pair $(A, \underline{b}_i)$ completely controllable. Also, if the multiple output system is given by $\dot{\underline{z}}(t) = F\underline{z}(t)$, $\underline{u}(t)=H\underline{z}(t)$, then any one row $\underline{h}_i^T$ of the matrix H has to render $(F^T, \underline{h}_i)$ completely observable. The latter requirement is easily fulfilled, because H can be freely chosen by the designer. The condition on B is generally not satisfied and the designer has no influence on it.

Let the multi-input system, as in equation (4.2.1-1), be given by

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t) \tag{4.2.3-1}$$

If (A,B) constitutes a controllable pair, then there exists a linear feedback law[3,5]

$$\underline{u}(t) = -G_0\,\underline{x}(t) \tag{4.2.3-2}$$

that assigns an arbitrarily specified set of eigenvalues to the closed-loop system

$$\dot{\underline{x}}(t) = (A - BG_0)\,\underline{x}(t) \tag{4.2.3-3}$$

The problem of finding G_0 for a given set of eigenvalues is generally non-linear and has many possible solutions. As already suggested by Simon and Mitter[5] one way to obtain a linear solution, is to restrict the control $\underline{u}$ to the form

$$\underline{u}(t) = \underline{\alpha}\, u_f(t) + \underline{u}_0(t) \tag{4.2.3-4}$$

where $\underline{\alpha}$ is a m-vector. When only feedback control is desired, let

$$\underline{u}_0(t) = \underline{0} \tag{4.2.3-5}$$

and

$$u_f(t) = - \underline{g}_0^T\, \underline{x}(t) \tag{4.2.3-6}$$

Substituting equation (4) to (6) in (1) yields

$$\dot{\underline{x}}(t) = (A - B\, \underline{\alpha}\, \underline{g}_0^T)\, \underline{x}(t) \tag{4.2.3-7}$$

Since B is a (nxm) matrix and $\underline{\alpha}$ a m-vector define

$$\underline{d} \triangleq B\, \underline{\alpha} \tag{4.2.3-8}$$

a n-vector. With equation (8) the closed loop expression (7) can be re-written to be

$$\dot{\underline{x}}(t) = (A - \underline{d}\, \underline{g}_0^T)\, \underline{x}(t) \tag{4.2.3-9}$$

When looking at equations (4) to (6), two questions arise immediately: First, is it always possible to express the feedback control as shown in equations (4) to (6)? Second, how can the vector $\underline{\alpha}$ be determined?

Simon and Mitter[5] have already given answers to both questions. It was shown in reference 5, that any controllable system can be converted into a pseudo single-input system, as long as the similar Jordan canonical form of matrix A has at most one Jordan block associated with the same multiple eigenvalue. The algorithm developed by Simon and Mitter[5] to determine the vector $\underline{\alpha}$ will not be repeated; instead a new one will be presented. The new algorithm will converge in maximally $(n+1)$ steps, compared with (n^2+1) steps of the Simon and Mitter method.

Before presenting the actual algorithm the following assumption concerning the class of systems involved is made:

A. To ensure the existence of single-input controllability of system (1) it is assumed, that the similar Jordan canonical form of A has at most one Jordan block associated with any multiple eigenvalue.

To obtain a suitable vector $\underline{\alpha}$, system (1) is transformed to Jordan canonical form. Let

$$\underline{x}(t) = M_1\, \underline{z}_1(t) \qquad (4.2.3\text{-}10)$$

be a non-singular transformation to Jordan canonical form, i.e.,

$$\dot{\underline{z}}_1(t) = M_1^{-1} A\, M_1\, \underline{z}(t) + M_1^{-1} B\, \underline{u}(t) \qquad (4.2.3\text{-}11)$$

where

$$J_1 = M_1^{-1} A M_1 = \begin{bmatrix} \lambda_1 & & & & & & & & & \\ & \lambda_2 & & & & & & & \bigcirc & \\ & & \ddots & & & & & & & \\ & & & \lambda_i & 1 & & & & & \\ & & & & \lambda_i & 1 & & & & \\ & & & & & \lambda_i & \ddots & & & \\ & & & & & & \ddots & 1 & & \\ & & & & & & & \lambda_i & & \\ & \bigcirc & & & & & & & \lambda_{i+1} & \\ & & & & & & & & & \ddots \ \lambda_n \end{bmatrix} \qquad (4.2.3\text{-}12)$$

and $D_1 \triangleq M_1^{-1} B$ (4.2.3-13)

Matrix D_1 renders the pair (J_1, D_1) completely controllable[16] iff

(1) none of the rows of D_1 corresponding to a simple eigenvalue is zero; and

(2) at least the row of D_1 corresponding to the last eigenvalue in each Jordan block is non-zero.

Since the original system (A,B) was assumed to be completely controllable, so will be (J_1,D_1).

To avoid computation with complex numbers, all conjugate complex eigenvalues $\lambda_i = \mu_i + j\nu_i$, $\lambda^*_i = \mu_i - j\nu_i$ are transformed into blocks of the form

$$\begin{bmatrix} \mu_i + j\nu_i & 0 \\ 0 & \mu_i - j\nu_i \end{bmatrix} \Rightarrow \begin{bmatrix} \mu_i & \nu_i \\ -\nu_i & \mu_i \end{bmatrix} \tag{4.2.3-14}$$

This transforms J_1 to a real valued matrix, defined to be J, and D_1 into the real-valued matrix D. A new set of state variables $\underline{z}(t)$ is obtained. $\underline{z}(t)$ and $z_1(t)$ are related by the similarity transformation M_2, i.e., $\underline{z}_1(t) = M_2\, \underline{z}(t)$. Again, for the pair (J,D) to be controllable, it has to fulfill conditions (1) and (2) above. Their meaning for the transformed complex eigenvalues is summarized in condition (3):

(3) at least one of the two rows of D corresponding to the last conjugate complex pair of each Jordan block of multiple complex eigenvalues in matrix J is non-zero.

Let the control $\underline{u}$ be again restricted to the form of equation (4), and let $\underline{u}_0 = \underline{0}$ in anticipation of a pure state feedback law. Then equation (11 becomes

$$\dot{\underline{z}}(t) = M_2^{-1} M_1^{-1} A M_1 M_2 \underline{z}(t) + M_2^{-1} M_1^{-1} B \underline{\alpha} u_f$$

$$= \bar{M}^{-1} A M \underline{z}(t) + \bar{M}^{-1} B \underline{\alpha} u_f$$

$$= J \underline{z}(t) + D \underline{\alpha} u_f \qquad (4.2.3\text{-}15)$$

Having transformed the original system (1) to the equivalent similar system (15) the computation of $\underline{\alpha}$ proceeds as follows:

(a) Define a m-vector $\underline{\alpha}'$ having all elements equal to one.

(b) Define

$$\underline{d}_J' = D \underline{\alpha}' = D \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \qquad (4.2.3\text{-}16)$$

(c) Test $\underline{d}_J'$ for controllability according to conditions (1) to (3). If these conditions are met, define

$$\underline{d}_J = \underline{d}_J' \quad \text{and} \quad \underline{\alpha} = \underline{\alpha}' \qquad (4.2.3\text{-}17)$$

and the desired vector $\underline{\alpha}$ is determined. If the controllability requirements (1) to (3) are not satisfied, vector $\underline{d}_J'$ must have a zero element, say in row k, where it should have a non-zero value.

(d) Scan row k of matrix D from left to right until a non-zero element is found, say in column j (such a non-zero element must exist, since (J,D) was assumed to be controllable).

(e) Determine element of smallest, non-zero magnitude in vector $\underline{d}_J'$, say element d_{J_i}'. Also determine element of largest magnitude in column j of matrix D, say $d_{\ell j}$.

(f) Now choose

$$\gamma_1 > \left| \frac{d_{l_j}}{d_{J_i}} \right| \tag{4.2.3-18}$$

and define

$$\underline{d}_J'' = \gamma_1 \underline{d}_J' + (\text{col. } j \text{ of matrix } D) \tag{4.2.3-19}$$

The new $\underline{\alpha}''$ is thus given by

$$\underline{\alpha}'' = \gamma_1 \underline{\alpha}' + \begin{array}{cl} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} & \begin{array}{l} \\ \\ \\ j^{th} \text{ of } m \text{ elements} \\ \\ \\ \\ \end{array} \end{array} \tag{4.2.3-20}$$

Thus $\underline{d}_J''$ has at least one zero-element less than $\underline{d}_J'$.

(g) Again test $(J, \underline{d}_J'')$ for controllability. Repeat the steps (d) to (f) until $\underline{d}_J^{(i)}$ $(1 \leq i \leq n+1,\ n = \text{order of matrix } A)$ renders $(J, \underline{d}_J^{(i)})$ controllable. Then define

$$\underline{d}_J = \underline{d}_J^{(i)}$$

and

$$\underline{\alpha} = \underline{\alpha}^{(i)}$$

Having determined a vector $\underline{\alpha}$ (one out of infinitely many) for the system (J,D), $\underline{\alpha}$ will also render the system (A,B) single-input controllable. Then by converting (A,B) into a pseudo single-input system, the algorithm presented in section 4.2.2 can be used to determine an appropriate feedback vector $\underline{g}_0$ for the arbitrary assignment of the closed-loop eigenvalues of $(A-B\,\underline{\alpha}\,\underline{g}_0^T)$.

4.2.4 Generalization of the pole-assignment algorithm.

Assumption A of section 4.2.3 excluded controllable systems, whose Jordan canonical form has more than one Jordan block associated with the same multiple eigenvalue, from the conversion to pseudo single-input systems. Except for this case, the algorithm does not exclude systems having multiple eigenvalues. However, numerical considerations make it preferable to deal with systems having distinct eigenvalues only. Finding the transformation to Jordan canonical form of systems with multiple eigenvalues is computationally very difficult.

Although actual physical systems very rarely have multiple eigenvalues[15] it would be desirable if the algorithm of the previous section could handle them with the same numerical ease as systems with distinct eigenvalues. To make the algorithm of section 4.2.3 applicable to all controllable systems (even those previously excluded by assumption A) and facilitate the computation of the similarity transformation to Jordan canonical form for systems with multiple eigenvalues use is made of the following well-known fact[3].

If the pair (A,B) is controllable, so is $(A-B\,G_m, B)$, where G_m is an $(m \times n)$ matrix.

In order to find the eigenvectors of A, it is necessary to first determine its eigenvalues. If A has multiple eigenvalues, some arbitrary matrix G_m can be chosen as feedback gain matrix. If G_m is arbitrary enough, i.e., its elements have different magnitudes, the closed-loop matrix

$$A_1 = A - B\,G_m \tag{4.2.4-1}$$

is very likely to have distinct eigenvalues. If A_1 has still some multiple eigenvalues, form

$$A_2 = A_1 - B\,G_m = A - 2\,B\,G_m$$

and continue until $A_i = A - i\,B\,G_m$ has distinct eigenvalues. This matrix A_i is then used to find a pseudo single-input system $(A_i, \underline{d})$ for which $\underline{g}_0$ can be determined easily. The final gain-matrix G_0 has to be adjusted.

$$G_0 = i \cdot G_m + \underline{\alpha}\,\underline{g}_0^T \tag{4.2.4-2}$$

Equation (4.2.2-9), $TF - AT = \underline{b}\,\underline{h}^T$, was guaranteed to have a unique solution for T only if F and A have no common eigenvalues, i.e., if none of the pre-specified closed-loop eigenvalues coincides with an open-loop eigenvalue. If it is desired to have some open and closed-loop eigenvalues in common, an arbitrary feedback loop is designed for A to obtain a matrix A_1 which does not have eigenvalues in common with F. The design of the feedback loop is done in the same way as in the case of multiple eigenvalues.

4.3 Computation of $\varkappa = \inf_M \|M\|\,\|M^{-1}\|$

Literature[17] is only concerned with determining the condition number for some fixed given matrix A. In this case it is easy to find the condition number

$$\varkappa(A) = \|A\|\,\|A^{-1}\| \tag{4.3-1}$$

for a matrix norm $\|\cdot\|$ induced by an absolute vector norm. However, given a matrix A it is generally difficult to determine the infimum of the condition numbers for the set of matrices M which transforms A to diagonal form, i.e.,

$$A = M\,L\,M^{-1} \tag{4.3-2}$$

where $L = \mathrm{diag}\left[\text{eigenvalues of } A\right]$.

Since the matrix M is non-unique, $M_\beta = M\,\mathrm{diag}\left[\{\beta_i\}\right]$ (where $\beta_i \neq 0$) again transforms A to diagonal form L. Knowledge of the condition number of the matrix M is very useful in obtaining bounds[14] on the magnitude of the eigenvalues of A and on their changes with respect to perturbations of A. To derive as accurate bounds as possible requires the determination of

$$\varkappa = \inf_{M} \left\| M \right\| \left\| M^{-1} \right\| \tag{4.3-3}$$

This author was not able to derive an algorithm by which $\varkappa$ as defined in equation (3) could be computed, irrespective of the type of absolute vector norm inducing the matrix norm $\|\cdot\|$. To come up with some results at all, attention had to be restricted to the 'one' and the 'infinity' norm[17]. The 'one' norm of a vector $\underline{x}$ is defined to be

$$\left\| \underline{x} \right\|_1 = \sum_{j=1}^{n} |x_j| \tag{4.3-4}$$

and the 'infinity' norm

$$\left\| \underline{x} \right\|_\infty = \lim_{p\to\infty} \left(\sum_{j=1}^{n} |x_j|^p \right)^{1/p} = \max_j |x_j| \tag{4.3-5}$$

These two vector norms induce the matrix norms

$$\left\| M \right\|_1 = \sup_{\underline{x} \neq \underline{0}} \frac{\left\| M\underline{x} \right\|_1}{\left\| \underline{x} \right\|_1} = \max_k \sum_{j=1}^{n} |m_{jk}| \tag{4.3-6}$$

and

$$\| M \|_{\infty} = \sup_{\underline{x} \neq \underline{0}} \frac{\| M\underline{x} \|_{\infty}}{\| \underline{x} \|_{\infty}} = \max_{j} \sum_{k=1}^{n} | m_{jk} | \qquad (4.3\text{-}7)$$

i.e., $\| M \|_1$ is the maximum absolute column sum and

$\| M \|_{\infty}$ is the maximum absolute row sum.

To compute $\mathcal{K}$ for a given similarity transformation M proceed as follows:

(a) 'One' norm:

α) Obtain the matrix $M_\beta = \left[\underline{m}_{\beta 1}, \underline{m}_{\beta 2}, \ldots, \underline{m}_{\beta n} \right]$ by normalizing the columns of M, i.e.,

$$\sum_{j=1}^{n} | m_{\beta_{jk}} | = 1 = \sum_{j=1}^{n} \frac{| m_{jk} |}{\sum_{j=1}^{n} | m_{jk} |} \quad \text{for } k=1, n \qquad (4.3\text{-}8)$$

β) Scale $N \triangleq M^{-1}$ appropriately to yield $N_\beta = M_\beta^{-1}$ where

$M_\beta \, M_\beta^{-1} = I$ and $N_\beta = \left[\underline{n}_{\beta_1}, \underline{n}_{\beta_2}, \ldots, \underline{n}_{\beta_n} \right]$

γ) Then $\mathcal{K}_1 = \inf_{M} \| M \|_1 \, \| M^{-1} \|_1$ is given by

$$\mathcal{K}_1 = \max_{k} \sum_{j=1}^{n} | n_{\beta_{jk}} | \qquad (4.3\text{-}9)$$

(b) 'Infinity' norm:

α) Define $Q \triangleq M^{-1} = \left[\underline{q}_1, \underline{q}_2, \ldots, \underline{q}_n \right]^T$. Obtain the matrix $Q_\beta = M_\beta^{-1}$ by normalizing the rows of M^{-1}, i.e.,

$$\sum_{k=1}^{n} | q_{\beta_{jk}} | = 1 = \sum_{k=1}^{n} \frac{| q_{jk} |}{\sum_{k=1}^{n} | q_{jk} |} \quad \text{for } j=1,n \qquad (4.3\text{-}10)$$

β) Scale $R \cdot M$ appropriately to yield $R_\beta = M_\beta$, where $M_\beta M_\beta^{-1} = I$ and $R_\beta = \left[\underline{r}_{\beta_1}, \underline{r}_{\beta_2}, \ldots, \underline{r}_{\beta_n}\right]^T$.

γ) Then $\varkappa_\infty = \inf\limits_M \| M \|_\infty \| M^{-1} \|_\infty$ is given by

$$\varkappa_\infty = \max_j \sum_{k=1}^{n} | r_{\beta_{jk}} | \tag{4.3-11}$$

In the following a proof will be given, that expression (11) really is the $\inf\limits_M \| M \|_\infty \| M^{-1} \|_\infty$. Assume that all but one absolute row sum of matrix $Q = M^{-1}$ equal one. Let the j^{th} row be the row with an absolute row sum not equal to one.

1. Assume $\sum\limits_{k=1}^{n} | q_{jk} | = \delta_1 > 1$. Then $\| Q \|_\infty = \delta_1$

   Let the greatest absolute row sum of matrix $R = M$ occur in row ℓ, i.e., $\| R \|_\infty = \sum\limits_{k=1}^{n} | r_{\ell k} | > 0$. Thus $\| M^{-1} \|_\infty \| M \|_\infty =$

$$\| Q \|_\infty \| R \|_\infty = \delta_1 \sum_{k=1}^{n} | r_{\ell k} | \tag{4.3-12}$$

   If now the j^{th} row of matrix $Q = M^{-1}$ is normalized such that its absolute row sum is

$$\sum_{k=1}^{n} | q_{\beta_{jk}} | = 1 = \sum_{k=1}^{n} \frac{| q_{jk} |}{\delta_1} \tag{4.3-13}$$

   Then $\| Q_\beta \| = 1$

   Due to the scaling of $Q = M^{-1}$, the matrix $R = M$ has to be scaled, too. In this case the j^{th} column of R has to be multiplied by δ_1. Thus the norm of R_β is given by

$$\| R_\beta \|_\infty \geq \delta_1 | r_{\ell j} | + \sum_{\substack{k=1 \\ k \neq j}}^{n} | r_{\ell k} | \tag{4.3-14}$$

The equal sign holds, if the ℓ^{th} row of R_β still yields the greatest absolute row sum. If a row other than row ℓ yields the largest row sum, $\| R_\beta \|_\infty$ can be bounded above by

$$\delta_1 \cdot \sum_{k=1}^{n} | r_{\ell k} | \geq \| R_\beta \|_\infty \tag{4.3-15}$$

Thus $\| M_\beta^{-1} \|_\infty \; \| M_\beta \|_\infty$ is bounded by

$$1 \cdot \delta_1 \cdot \sum_{k=1}^{n} | r_{\ell k} | \geq \| M^{-1}{}_\beta \|_\infty \; \| M_\beta \|_\infty \geq 1 \cdot \Big(\delta_1 \, | r_{\ell j} | + \sum_{\substack{k=1 \\ k \neq j}}^{n} | r_{\ell k} | \Big) \tag{4.3-16}$$

A comparison of expressions (12) and (16) gives

$$\| M^{-1} \|_\infty \; \| M \|_\infty \geq \| M_\beta^{-1} \|_\infty \; \| M_\beta \|_\infty \tag{4.3-17}$$

2. Assume $\sum_{j=1}^{n} | q_{jk} | = \delta_2 < 1$, then $\| Q \|_\infty = 1$.

Following a similar line of reasoning as in case 1, leads to an expression which is identical with inequality (17). The new expressions corresponding to relations (12) to (17) are noted below without comment.

$$(12) \rightarrow \| M^{-1} \|_\infty \; \| M \|_\infty = \| Q \|_\infty \; \| R \|_\infty = 1 \cdot \sum_{k=1}^{n} | r_{\ell k} | \tag{4.3-18}$$

$$(13) \rightarrow \| Q_\beta \|_\infty = 1 \tag{4.3.19}$$

$$(14) \rightarrow \| R_\beta \|_\infty \geq \delta_2 \, | r_{\ell j} | + \sum_{\substack{k=1 \\ k \neq j}}^{n} | r_{\ell k} | \tag{4.3-20}$$

$$(15) \rightarrow \sum_{k=1}^{n} |r_{ik}| \geq \|R_\beta\|_\infty \tag{4.3-21}$$

$$(16) \rightarrow 1 \cdot \sum_{k=1}^{n} |r_{ik}| \geq \|M_\beta^{-1}\|_\infty \, \|M_\beta\|_\infty \geq 1 \cdot \Big(\delta_2 |r_{ij}| + \sum_{\substack{k=1 \\ k \neq j}}^{n} |r_{ik}|\Big) \tag{4.3.22}$$

$$\text{Thus} \quad \|M^{-1}\|_\infty \, \|M\|_\infty \geq \|M_\beta^{-1}\|_\infty \, \|M_\beta\|_\infty \tag{4.3-23}$$

Inequalities (17) and (23) show that the condition number of the matrix M^{-1}, when it is not normalized, is always larger than or equal to the condition number of the normalized matrix M_β^{-1}.

Q.E.D.

Similar proof can be given for expression (4.3-9).

4.4 Numerical Examples for the Pole-Placement Algorithm

A Fortran IV computer program was written to mechanize the algorithm presented in section 4.2. A listing of the program can be found in Appendix B. The program was written in such a way as to allow either pole-placement or pole-placement and solution of the algebraic matrix Riccati equation. The subroutine to solve the Riccati equation is based on Kleinman's[9] iterative solution technique.

If the computer program is used only to determine feedback gains for the pole-assignment task, about 80% of the computation time is spent on checking the controllability of the pair (A,B) and converting it to a pseudo single-input system. Thus, if it is known that every column $\underline{b}_i$ of the matrix B renders (A, b_i) controllable the conversion step can be omitted.

The following pages present 11 examples. Each of them is run through all steps of the program. These steps are:

1. Check for multiple eigenvalues or common open- and closed-loop eigenvalues.
2. Conversion to a pseudo single-input system.
3. Determination of feedback gains to assign desired pole-locations.
4. Check of closed-loop eigenvalues.
5. Computation of Riccati matrix.
6. Back substitution of solution into matrix Riccati equation.

Steps 4 to 6 are optional.

After the program was debugged it never failed to determine a set of appropriate feedback gains, whether for distinct or multiple real or complex eigenvalues. The results are very accurate as can be seen from the following examples. The computer print-out gives all necessary information for easy understanding; the notation is the same as in the previous sections.

```
STATES #   2      INPUTS #   1

A - SYSTEM MATRIX
    0.20000000D 01       0.C
   -0.30000000D 01      -0.600C0000D 01

B - INPUT MATRIX
    0.50000000D 00
    0.0

DESIRED EIGENVALUES OF %A - B*G<
        -1.00000000           0.0
        -6.00000000           0.0

MATRIX  SINV.
    0.38067791D 00       0.100C0000D 01
    0.10000000D 01      -0.12689264D 00

MATRIX  SINV*B .
    0.19033896D 00
    0.50000000D 00

VECTOR ALPHA%TRANS< %VECTCR  D # B*ALPHA<.
    0.10000000D 01

VECTOR  D%TRANS<.
    0.1^033896D 00       0.500C0000D 00

DIAGONALIZED MATRIX  A, OR DIAGONALIZED MATRIX %A-B*ZG< IN THE CASE
OF MULTIPLE EIGENVALUES AND/OR  COMMON OPEN- AND CLOSEC-LOOP EIGENVALUES.
   -0.63806779D 01       0.C
    0.0                  0.18806779D 01

G - GAIN MATRIX
    0.60000000D 01      -0.22204460D-15

T - SOLUTION MATRIX
    0.35374531D-01       0.500C0000D 00
   -0.17357026D 00      -0.63446318D-01

COMPUTED EIGENVALUES OF %A - B*G<.
        -1.00000000           0.0
        -6.00000000           0.0

MATRIX  Q .
    0.20000000D 01       0.0
    0.0                  0.200C0000D 01

MATRIX  R .
    0.40000000D 00

GAIN TOLERANCE .LE.   0.56011145D-06WAS ACHEIVED AFTER    5 ITERATIONS.

MATRIX  R%INVERSE< .
    0.25000000D 01

RICCATI MATRIX  P .
    0.69436492D 01      -0.59886422D-01
   -0.59886422D-01       0.16647988D 00

RESIDUAL MATRIX. MATRIX IS ZERO. IF MATRIX P IS ACCURATE.
   -0.94753494D-11      -0.63143935D-13
   -0.63199446D-13       0.45514807D-15
```

```
STATES #   2        INPUTS #   1

A - SYSTEM MATRIX
    0.100000000D 01        0.100000000D 01
    0.0                    0.100000000D 01

B - INPUT MATRIX
    0.0
    0.100000000D 01

DESIRED EIGENVALUES OF %A - B*G<
       -5.00000000          0.0
       -7.00000000          0.0

MATRIX  SINV.
    0.585786644D 00        0.100000000D 01
    0.100000000D 01        0.292893322D 00

MATRIX  SINV*B .
    0.100000000D 01
    0.292893322D 00

VECTOR ALPHA%TRANS< %VECTOR  D # B*ALPHA<.
    0.100000000D 01

VECTOR  D%TRANS<.
    0.100000000D 01        0.292893322D 00

DIAGONALIZED MATRIX  A, OR DIAGONALIZED MATRIX %A-B*ZG< IN THE CASE
OF MULTIPLE EIGENVALUES AND/OR  COMMON OPEN- AND CLOSED-LOOP EIGENVALUES.
   -0.241421136D 01        0.0
    0.0                    0.414213560D 00

G - GAIN MATRIX
    0.480000000D 02        0.140000000D 02

T - SOLUTION MATRIX
   -0.386729954D 00       -0.218065100D 00
   -0.540970940D-01       -0.395042870D-01

COMPUTED EIGENVALUES OF %A - B*G<.
       -7.00000000          0.0
       -5.00000000          0.0

MATRIX  Q .
    0.100000000D 01        0.0
    0.0                    0.100000000D 01

MATRIX  R .
    0.500000000D 00

GAIN TOLERANCE .LE.   0.87523780D-07WAS ACHEIVED AFTER   8 ITERATIONS.

MATRIX  R%INVERSE< .
    0.200000000D 01

RICCATI MATRIX  P .
    0.894566880D 01        0.307338070D 01
    0.307338070D 01        0.245534670D 01

RESIDUAL MATRIX. MATRIX IS ZERO, IF MATRIX P IS ACCURATE.
   -0.145661260D-12       -0.512923040D-13
   -0.512923040D-13       -0.142108550D-13
```

```
 STATES #    2       INPUTS #    1

A - SYSTEM MATRIX
     0.500000000 01      0.200000000 01
    -0.100000000 01      0.300C00000 01

B - INPUT MATRIX
     0.100000000 00
     0.0

DESIRED EIGENVALUES OF %A - B*G<
        -5.00000000          0.0
        -7.00000000          0.0

MATRIX  SINV.
     0.100000000 01      0.200C00000 01
    -0.100000000 01      0.0

MATRIX  SINV*B .
     0.100000000 00
    -0.100000000 00

VECTOR ALPHA%TRANS< %VECTCR  D # B*ALPHA<.
     0.100000000 01

VECTOR  D%TRANS<.
     0.100000000 00     -C.100C0000D 00

DIAGONALIZED MATRIX  A, OR DIAGONALIZED MATRIX %A-B*ZG< IN THE CASE
OF MULTIPLE EIGENVALUES ANC/OR  COMMON OPEN- AND CLOSEC-LOOP EIGENVALUES.
     0.400000000 01      0.100C0000D 01
    -0.100000000 01      C.400C00000 01

G - GAIN MATRIX
     0.200000000 03     -0.780C00000 03

T - SOLUTION MATRIX
    -0.121951220-01     -0.983606560-02
     0.975609760-02      0.819672130-02

COMPUTED EIGENVALUES OF %A - B*G<.
        -7.00000000          0.0
        -5.00000000          0.0

MATRIX  Q .
     0.800000000 00      0.C
     0.0                 0.800000000 00

MATRIX  R .
     0.500000000 00

GAIN TOLERANCE .LE.   0.52252930-07WAS ACHEIVED AFTER    5 ITERATIONS.

MATRIX  R%INVERSE< .
     0.200000000 01

RICCATI MATRIX  P .
     0.800079400 03     -0.240047350 04
    -0.240047350 04      0.208C77580 05

RESIDUAL MATRIX. MATRIX IS ZERO, IF MATRIX P IS ACCURATE.
    -0.436557460-10      0.591171560-10
     0.609361450-10      0.174622980-09
```

```
STATES #    2        INPUTS #    1

A - SYSTEM MATRIX
   -0.200000000D 01        0.300C00000D 01
    0.100000000D 01        0.100000000D 01

B - INPUT MATRIX
    0.0
    0.100000000D 01

DESIRED EIGENVALUES OF %A - B*G<
        -3.00000000             0.0
        -5.00000000             0.0

MATRIX  SINV.
    0.100000000D 01       -0.791287850 00
    0.263762620 00         0.100C00000D 01

MATRIX  SINV*B .
   -0.791287850 00
    0.100000000D 01

VECTOR ALPHA%TRANS< %VECTCR  D # B*ALPHA<.
    0.100000000D 01

VE.TOR  D%TRANS<.
   -0.791287850 00         0.100CC0000D 01

DIAGONALIZED MATRIX  A, OR DIAGONALIZED MATRIX %A-B*ZG< IN THE CASE
OF MULTIPLE EIGENVALUES AND/OR  COMMON OPEN- AND CLOSEC-LOOP EIGENVALUES.
   -0.279128780 01         0.0
    0.0                    0.179128780 01

G - GAIN MATRIX
    0.200000000D 01        0.700C0000D 01

T - SOLUTION MATRIX
    0.379128780 01         0.358257570 00
   -0.208712150 00        -0.147247480 00

COMPUTED EIGENVALUES OF %A - B*G<.
        -5.000C0000             0.0
        -3.00000000             0.0

MATRIX  Q .
    0.100000000D 01        0.0
    0.0                    0.100C00000D 01

MATRIX  R .
    0.500000000D 00

GAIN TOLERANCE .LE.   0.82324680-07WAS ACHEIVED AFTER    5 ITERATIONS.

MATRIX  R%INVERSE< .
    0.200000000D 01

RICCATI MATRIX  P .
    0.368294610 00         0.615804930 00
    0.615804930 00         0.211164970 01

RESIDUAL MATRIX. MATRIX IS ZERO, IF MATRIX P IS ACCURATE.
   -0.49821258D-14        -0.162092560-13
   -0.159872120-13        -C.528466160-13
```

```
STATES #   3        INPUTS #   2

A - SYSTEM MATRIX
   -0.30000000D 01       0.C                      0.50000000C 01
    0.17000000D 02       0.200CC000D 01          -0.50000000D 01
    0.0                  0.100C0000D 01           0.0

B - INPUT MATRIX
    0.0                 -0.30000000D 00
    0.0                  C.0
    0.15000000D 02       0.C

DESIRED EIGENVALUES OF %A - B*G<
      -20.00000C00           0.0
      -10.00000000           5.00000000
      -10.00000C00          -5.C0000000

MATRIX  SINV.
   -0.12572636D 01      -0.27966999D 00           0.20000000D 01
   -0.11691541D 01       0.21828704D 00           0.0
    0.10000000D 01       0.40517459D 00           0.76495667D 00

MATRIX  SINV*B .
    0.30000000D 02       0.37717908D 00
    0.0                  0.35074623D 00
    0.11474350D 02      -0.300C0000D 00

VECTOR ALPHA%TRANS< %VECTOR  D # B*ALPHA<.
    0.10000000D 01       C.100CC000D 01

VECTOR  D%TRANS<.
    0.30377179D 02       0.35074623D 00           0.11174350D 02

DIAGONALIZED MATRIX  A, OR DIAGONALIZED MATRIX %A-B*ZG< IN THE CASE
OF MULTIPLE EIGENVALUES AND/OR  COMMON OPEN- AND CLOSED-LOOP EIGENVALUES.
   -0.24439840D 01       0.34686028D 01           0.0
   -0.34686028D 01      -0.24439840D 01           0.0
    0.0                  0.0                      0.38879680D 01

G - GAIN MATRIX
    0.10933687D 02       0.42804852D 01           0.28186737D 01
    0.10933687D 02       0.42804852D 01           0.28186737D 01

T - SOLUTION MATRIX
   -0.16614961D 01      -0.14368850D 01          -0.41926565D 01
   -0.34824621D 00       0.41608385D 00          -0.16957332D 01
   -0.46778152D 00      -0.45584379D 00          -0.96872119D 00

COMPUTED EIGENVALUES OF %A - B*G<.
      -20.00C00000           0.0
      -10.00000C00           5.00000000
      -10.00C00000          -5.0000C000

MATRIX  Q .
    0.10000000D 01       0.0                      0.0
    0.0                  0.800C0000D 00           0.0
    0.0                  0.C                      0.40000000C 00

MATRIX  R .
    0.40000000D 00       0.0
    0.0                  0.500CC000D 00

GAIN TOLERANCE .LE.    0.37049300D-04WAS ACHEIVED AFTER    6 ITERATIONS.

MATRIX  R%INVERSE< .
    0.25000000D 01       0.C
    0.0                  0.200CCC00D 01

RICCATI MATRIX  P .
    0.17554832D 01       0.82671120D 00           0.17932502D 00
    0.82871120D 00       0.46168698D 00           0.68775734C-01
    0.17932502D 00       0.68775734D-01           0.51634691C-01

RESIDUAL MATRIX. MATRIX IS ZERO, IF MATRIX P IS ACCURATE.
   -0.34833967D-08      -0.12873418D-08          -0.26622384D-08
   -0.12873338D-08      -0.47603987D-09          -0.98371822D-09
   -0.26622267D-08      -0.98371600D-09          -0.20597821C-08
```

```
STATES #   3       INPUTS #   2

A - SYSTEM MATRIX
   -0.50000000D 00      -0.10000000D 01      -0.25000000D 01
    0.0                 -0.20000000D 01       0.0
    0.50000000D 00      -0.10000000D 01      -0.35000000D 01

B - INPUT MATRIX
    0.14600000D 02       0.54000000D 01
    0.40000000D 00      -0.40000000D 00
    0.26000000D 01       0.74000000D 01

DESIRED EIGENVALUES OF ZA - B*G<
      -15.00000000         0.0
       -3.00000000         2.00000000
       -3.00000000        -2.00000000

MATRIX  SINV.
   -0.20000000D 00       0.80000000D 00       0.10000000D 01
    0.10000000D 01       0.11102230D-15      -0.10000000D 01
   -0.99296046D-21       0.10000000D 01      -0.99295113D-21

MATRIX  SINV*B .
    0.44408921D-15       0.60000000D 01
    0.12000000D 02      -0.20000000D 01
    0.40000000D 00      -0.40000000D 00

VECTOR ALPHAZTRANS< ZVECTOR  D # B*ALPHA<.
    0.30000000D 01       0.20000000D 01

VECTOR  DZTRANS<.
    0.12000000D 02       0.32000000D 02       0.40000000D 00

DIAGONALIZED MATRIX  A, OR DIAGONALIZED MATRIX ZA-B*ZG< IN THE CASE
OF MULTIPLE EIGENVALUES AND/OR  COMMON OPEN- AND CLOSED-LOOP EIGENVALUES.
   -0.30000000D 01       0.0                  0.0
    0.0                 -0.10000000D 01       0.0
    0.0                  0.0                 -0.20000000D 01

G - GAIN MATRIX
    0.40500000D 01      -0.48270000D 03       0.75000000D 00
    0.27000000D 01      -0.32180000D 03       0.50000000D 00

T - SOLUTION MATRIX
   -0.10000000D 01       0.60000000D 01      -0.60000000D 01
   -0.22857143D 01      -0.21844748D-11      -0.16000000D 02
   -0.30769231D-01       0.80000000D-01      -0.24000000D 00

COMPUTED EIGENVALUES OF ZA - B*G<.
      -15.00000000         0.0
       -3.00000000         2.00000000
       -3.00000000        -2.00000000

MATRIX  Q .
    0.10000000D 01       0.0                  0.0
    0.0                  0.80000000D 00       0.0
    0.0                  0.0                  0.40000000D 00

MATRIX  R .
    0.40000000D 00       0.0
    0.0                  0.50000000D 00

GAIN TOLERANCE .LE.   0.32166633D-04WAS ACHEIVED AFTER  17 ITERATIONS.

MATRIX  RZINVERSE< .
    0.25000000D 01       0.0
    0.0                  0.20000000D 01

RICCATI MATRIX  P .
    0.43661382D-01      -0.58264021D-02      -0.13083129D-01
   -0.58264021D-02       0.19765597D 00       0.44087562D-02
   -0.13083129D-01       0.44087562D-02       0.45112677D-01

RESIDUAL MATRIX. MATRIX IS ZERO, IF MATRIX P IS ACCURATE.
   -0.86458618D-14       0.33702381D-12       0.25870799D-13
    0.336[illegible]519D-12      -0.86545224D-11      -0.59984483D-12
    0.25[illegible]319D-11      -0.55387085D-12      -0.47684074D-11
```

```
STATES #   4        INPUTS #   2

A - SYSTEM MATRIX
   -0.300000000 01       0.150000000 02       0.100000000 00       0.0
    0.0                  0.100000000 01       0.100000000 01       0.100000000 01
    0.0                  0.100000000 00      -0.200000000 00       0.300000000 02
    0.500000000 03       0.700000000 01      -0.300000000 01       0.0

B - INPUT MATRIX
    0.500000000 00       0.0
    0.0                  0.0
    0.0                  0.0
    0.0                  0.200000000 01

DESIRED EIGENVALUES OF $A - B*G<
      -10.00000000           5.00000000
      -10.00000000          -5.00000000
      -12.00000000           2.00000000
      -12.00000000          -2.00000000

MATRIX  SINV.
    0.100000000 01       0.65158228D 00       0.23606937D-01       0.55235819D-01
    0.200000000 01      -0.30077983D 00      -0.60282296D-01      -0.93360468D-02
    0.27755576D-16       0.12062045D 01      -0.16453554D-01      -0.92931099D-01
    0.100000000 01      -0.86415722D 00       0.42963185D-01      -0.26299772D-01

MATRIX  SINV*B .
    0.500000000 00       0.11047164D 00
    0.100000000 01      -0.18672094D-01
    0.13877788D-16      -0.18586220D 00
    0.500000000 00      -0.52599544D-01

VECTOR ALPHA$TRANS< $VECTOR  D # B*ALPHA<.
    0.100000000 01       0.100000000 01

VECTOR  D$TRANS<.
    0.61047164D 00       0.98132791D 00      -0.18586220D 00       0.44740046D 00

DIAGONALIZED MATRIX  A, OR DIAGONALIZED MATRIX $A-B*ZG< IN THE CASE
OF MULTIPLE EIGENVALUES AND/OR  COMMON OPEN- AND CLOSED-LOOP EIGENVALUES.
    0.24617910D 02       0.0                  0.0                  0.0
    0.0                 -0.53340117D 01       0.23232775D 02       0.0
    0.0                 -0.23232775D 02      -0.53340117D 01       0.0
    0.0                  0.0                  0.0                 -0.16149886D 02

G - GAIN MATRIX
    0.73586485D 02       0.24764396D 02       0.26243728D 00       0.25033788D 01
    0.73586485D 02       0.24764396D 02       0.26243728D 00       0.25033788D 01
```

```
T - SOLUTION MATRIX
   -0.14779221D-01     -0.19769182D-01     -0.15713959D-01     -0.17529661D-01
   -0.24892343D-01     -0.96020017D-02     -0.21350040D-01     -0.1 1155480-01
   -0.35173178D-01     -0.45667674D-01     -0.34708626C-01     -0.39395584D-01
    0.79407463D-01      0.81892799D-02      0.12965411D 00      0.45324673D-01

COMPUTED EIGENVALUES OF %A - B*G<.
      -10.00000001          5.00000000
      -10.00000001         -5.00000000
      -11.99999999          1.99999999
      -11.99999999         -1.99999999

MATRIX  Q .
    0.50000000D 00      0.0                 0.0                 0.0
    0.0                 0.50000000D 00      0.0                 0.0
    0.0                 0.0                 0.50000000D 00      0.0
    0.0                 0.0                 0.0                 0.50000000D 00

MATRIX  R .
    0.40000000D 00      0.0
    0.0                 0.40000000D 00

GAIN TOLERANCE .LE.   0..6079030-04WAS ACHEIVED AFTER   9 ITERATIONS.

MATRIX  R%INVERSE< .
    0.25000000D 01      0.0
    0.0                 0.25000000D 01

RICCATI MATRIX  P .
    0.79939851D 02      0.49024506D 02      0.20021188D 01      0.46934072D 01
    0.49024506D 02      0.37129250D 02      0.13380884D 01      0.29030643D 01
    0.20021188D 01      0.13380884D 01      0.84854055D-01      0.14015441D 00
    0.46934072D 01      0.29030643D 01      0.14015441D 00      0.30787147D 00

RESIDUAL MATRIX. MATRIX IS ZERO, IF MATRIX P IS ACCURATE.
   -0.50991050D-05     -0.20932569D-06     -0.11699046D-06     -0.44454683D-06
   -0.20932582D-06     -0.93814108D-07     -0.10767357C-07     -0.24776584D-07
   -0.11699046D-06     -0.10767353D-07     -0.31016853C-08     -0.10656235D-07
   -0.44454685D-06     -0.24776587D-07     -0.10656236D-07     -0.39255993D-07
```

```
STATES #   4      INPUTS #   1

A - SYSTEM MATRIX
    0.0                 0.100C00000 01      0.0                 0.0
    0.0                 0.0                 0.100000000 01      0.0
    0.0                 0.0                 0.0                 0.100000000 01
    0.100000000 01      0.200C00000 01      0.300000000 01      0.400000000 01

B - INPUT MATRIX
    0.0
    0.0
    0.0
    0.100000000 01

DESIRED EIGENVALUES OF %A - B*G<
       -2.00000000         2.0C000000
       -2.00000000        -2.00000000
       -2.00000000         2.00000000
       -2.00000000        -2.00000000

MATRIX  SINV.
    0.211299170 00      0.467245690 00      0.73262615C 00      0.1000C0000 01
   -0.841123660-02      0.127161720 01      0.200000000 01     -0.479293330 00
   -0.793219330 00     -0.142333280 01      0.0                 0.710312160-01
    0.386664540 00      0.906145270-01      0.100000000 01     -0.218992640 00

MATRIX  SINV*B .
    0.100000000 01
   -0.479293330 00
    0.710312160-01
   -0.218992640 00

VECTOR ALPHA%TRANS< %VECTER  D # B*ALPHA<.
    0.100000000 01

VECTOR  D%TRANS<.
    0.100000000 01     -0.479293330 00      0.710312160-01     -0.218992640 00

DIAGONALIZED MATRIX  A, OR DIAGONALIZED MATRIX %A-B*ZG< IN THE CASE
OF MULTIPLE EIGENVALUES AND/OR  COMMON OPEN- AND CLOSED-LOOP EIGENVALUES.
    0.473262620 01      0.C                 0.0                 0.0
    0.0                -0.831313730-01      0.60511960C 00      0.0
    0.0                -0.605'19600 00     -0.831313730-01      0.0
    0.0                 C.0                 0.0                -0.566363410 00

G - GAIN MATRIX
    0.650000000 02      0.660C00000 02      0.35000000C 02      0.120000000 02
```

```
T - SOLUTION MATRIX
   -0.959414880D-01     -0.177030920D 00     -0.101858510D 00     -0.205083110D 00
    0.915387530D-03      0.250444170D 00     -0.609546430D-01      0.319371920D 00
   -0.385868860D-01      0.174434730D-02     -0.689005790D-01     -0.721479620D-02
   -0.204827400D-01      0.124178730D 00     -0.663469620D-01      0.146813670D 00

COMPUTED EIGENVALUES OF %A - B*G<.
       -2.00003549          2.00004765
       -2.00003549         -2.00004765
       -1.99996451          1.99995235
       -1.99996451         -1.99995235

MATRIX  Q .
    0.600000000D 00      0.100000000D 00      0.0                  0.0
    0.100000000D 00      0.800000000D 00      0.0                  0.0
    0.0                  0.0                  0.400000000D 00      0.100000000D-01
    0.0                  0.0                  0.100000000D-01      0.900000000D 00

MATRIX  R .
    0.200000000D 00

MORE THAN 15 LOOPS FOR EIGENVECTOR OF -0.2000D 01  0.2000D 01 DIFFERENCE OF  0.3240E-07

EIGENVECTOR ERROR MESSAGE
SW1=0.7684D-09          ITER=   15          DIF=0.3240E-07

MORE THAN 15 LOOPS FOR EIGENVECTOR OF -0.2000D 01  0.2000D 01 DIFFERENCE OF  0.3240E-07

EIGENVECTOR ERROR MESSAGE
SW1=0.7684D-09          ITER=   15          DIF=0.3240E-07

GAIN TOLERANCE .LE.   0.2958385D-04WAS ACHEIVED AFTER  19 ITERATIONS.

MATRIX  R%INVERSE< .
    0.500000000D 01

RICCATI MATRIX  P .
    0.184663520D 01      0.264168560D 01      0.181406740D 01      0.600000000D 00
    0.264168560D 01      0.649474960D 01      0.546782840D 01      0.157331760D 01
    0.181406740D 01      0.546782840D 01      0.660916320D 01      0.222084280D 01
    0.600000000D 00      0.157331760D 01      0.222084280D 01      0.210703370D 01

RESIDUAL MATRIX. MATRIX IS ZERO, IF MATRIX P IS ACCURATE.
   -0.246465070D-10     -0.484220890D-09     -0.817526710D-09     -0.101929800D-09
   -0.484221110D-09     -0.729865880D-08     -0.125110730D-07     -0.205045230D-08
   -0.817526710D-09     -0.125110730D-07     -0.215820220D-07     -0.359341710D-08
   -0.101929130D-09     -0.205045230D-08     -0.359341360D-08     -0.519893460D-09
```

```
STATES #   6      INPUTS #   2

A - SYSTEM MATRIX
    0.0              0.C              0.0              0.0              0.100CC0000 01   0.0
    0.0              0.0              0.0              0.0              0.0              0.100000000 01
    0.0              0.0              0.0              0.10000Cr000 01  0.0              0.0
   -0.750000000 01   C.0             -0.115000000 02  -0.450000000 01  -0.155000000 02   0.0
    0.0              0.0              0.100000000 01   0.0              0.0              0.0
    0.0             -0.925C00000 01   0.0              0.0              0.0             -0.10000C000 01

B - INPUT MATRIX
    0.0              0.0
    0.0              0.C
    0.0              0.0
    0.150000000 02  -0.750C00000 01
    0.0              0.0
    0.462500000 01   0.138750000 02

DESIRED EIGENVALUES OF ZA - B*G<
       -1.000C0C00       0.0
       -2.000C0000       0.0
       -3.00000000       1.00000000
       -3.00C00000      -1.00000000
       -5.00000000       0.0
       -6.00000000       0.0

MATRIX  SINV.
    0.236711850 00   0.306917930 00   0.601452900 00   0.424433790 00   0.100C0000? 01   0.604730240 00
    0.487839110 00   0.102903330 00   0.12003981C 01   0.483578370-01   0.200C00000 01   0.167055330-01
   -0.134378620-01   0.355765900-01   0.17569219C 00   0.176083930 00  -0.277555760-16   0.987970090-01
    0.284987730 00   0.803989630-01   0.110547560 00   0.181827410-01   0.160000000 01   0.108742040-01
    0.100000000 01  -0.559485530 00  -0.314760080 00   0.140267030-01   0.904940530 00   0.854470460-03
   -0.641136350 00   0.100C00000 01   0.614880600 00  -0.120314720-01  -0.839278410 00   0.546797420-02

MATRIX  SINV*B .
    0.916338430 01   C.520737870 01
    0.801000650 00  -0.129784500 00
    0.307819520 01   0.50179C000-01
    0.373034320 00   0.145090260-01
    0.214352470 00  -0.933444930-01
   -0.155182700 00   0.166104180 00

VECTOR ALPHAZTRANS< ZVECTCR  D # B*ALPHA<.
    0.100000000 01   0.1C0CC0000 01

VECTOR  URTRANS<.
    0.143707630 02   0.673216150 00   0.31483742C 01   0.337543340 00   0.121C07970 00   0.109214820-01

DIAGONALIZED MATRIX  A, OR DIAGONALIZED MATRIX ZA-B*ZG< IN THE CASE
OF MULT.PLE EIGENVALUES AND/OR  COMMON OPEN- AND CLOSEC-LOOP EIGENVALUES.
   -0.521589330 02   0.0              0.0              0.0              0.0              0.0
    0.0             -0.213835530 01   0.911685730 01   0.0              0.0              0.0
    0.0             -0.911685730 01  -0.213835530 01   0.0              0.0              0.0
    0.0              0.0              0.0             -0.161201630 01   0.0              0.0
    0.0              0.0              0.0              0.0             -0.319552740 00   0.0
    0.0              0.C              0.0              0.0              0.0             -0.382787130 00
```

G - GAIN MATRIX

0.40096166D 02	-0.57658305D 02	-0.34027687D 02	-0.43808905D 00	0.51297629D 02	-0.19169909D 01
0.39096166D 02	-0.55658305D 02	-0.37027687D 02	0.35619110D 01	0.46297629D 02	0.40830091D 01

T - SOLUTION MATRIX

0.28090427D 00	0.28650456D 00	0.29815599D 00	0.28626754D 00	0.30473045D 00	0.31133222D 00
0.34911279D 00	0.34637632D 00	0.32391546D 00	0.35426975D 00	0.29326379D 00	0.26628297D 00
-0.30251844D-01	-0.68586479D-01	-0.14331539C 00	-0.71796159D-01	-0.16589411D C0	-0.18663299C 00
0.55152668D 00	-0.86999373D 00	-0.44750166D-01	-C.27543O8[illegible] 00	-0.99629567D-01	-0.76924476C-01
-0.17783593D 00	-0.72009385D-01	-0.24844539C-01	-0.544134[illegible]-01	-0.25853934D-01	-0.21302543D-01
-0.17694838D-01	-0.67532741D-02	-0.22500387D-02	-0.50326516D-02	-0.23653841D-02	-0.19442884D-02

COMPUTED EIGENVALUES OF %A - B*G<.

-6.0C000000	0.0
-5.00C00C00	0.0
-3.00000000	1.00000000
-3.000C0000	-1.00000000
-1.000C0000	0.0
-2.00000000	0.0

MATRIX Q .

0.25000000D 01	0.C	0.0	0.0	0.0	0.0
0.0	0.250C0000D 01	0.0	0.0	0.0	0.0
0.0	0.0	0.25000000D 01	0.0	0.0	0.0
0.0	0.0	0.0	0.2500C0000 01	0.0	0.0
0.0	0.C	0.0	0.0	0.25000000D 01	0.0
0.0	0.C	0.0	0.0	0.0	0.25000C00D 01

MATRIX R .

0.50000000D 00	0.C
0.0	0.500C0000D 00

GAIN TOLERANCE .LE. 0.50728090-04WAS ACHEIVED AFTER 22 ITERATIONS.

MATRIX R%INVERSE< .

0.20000000D 01	0.0
0.0	0.200C0000D 01

RICCATI MATRIX P .

0.61086881D 01	0.83056408D-02	0.26525001D 01	0.54934096D-01	0.62062541D 01	0.35134C24D-02
0.83056408D-02	0.26388867D 01	0.12715444D-01	0.26612129D-02	0.18359871D-01	0.58077758C-01
0.26525001D 01	0.12715444D-01	0.64035913C 01	0.14673619D 00	0.64093552D 01	0.10595823C-01
0.54934096D-01	0.26612129D-02	0.14673619D 00	0.63324688D-01	0.13834560D 00	0.47503134D-02
0.62062541D 01	0.18359871D-01	0.64093552ʳ 01	0.13834560D 00	0.12477264D 02	0.93589116D-02
0.35134024D-02	0.58077758D-01	0.10595823C-01	0.47503134D-02	0.93589116D-02	0.76458345D-01

RESIDUAL MATRIX. MATRIX IS ZERO, IF MATRIX P IS ACCURATE.

-0.23224735D-09	0.20516436D-09	-0.45963028C-09	-0.16055548D-10	-0.1C0C0960D-08	0.50425358C-11
0.20516497D-09	-0.69136830D-09	0.18567034C-08	0.46996060D-10	0.26198539D-08	-0.17409851C-10
-0.45963993D-09	0.18567034D-08	-0.47466122D-08	-0.12339594D-09	-0.69512729D-08	0.47469931C-10
-0.16059237D-10	0.46995699D-10	-0.12339457C-09	-0.31690648D-11	-0.18263341D-09	0.12802338C-11
-0.1C001040D-08	0.26198540D-08	-0.69512695D-08	-0.18263319D-09	-0.10282278D-07	0.66211508C-10
0.50432991D-11	-0.17409851D-10	0.47470111D-10	0.128C7672D-11	0.66211481D-10	-0.22781776C-12

STATES # 6 INPUTS # 2

A - SYSTEM MATRIX

0.0	C.0	0.0	0.0	0.100CC0000 01	0.0
0.0	0.0	0.0	0.0	0.0	0.100000000 01
0.0	0.C	0.0	0.100000000 01	0.0	0.0
-0.750000000 01	0.0	-0.115000000 02	-0.450000000 01	-0.155000000 02	0.0
0.0	0.0	0.100000000 01	0.0	0.0	0.0
0.0	-0.925C00000 01	0.0	0.0	0.0	-0.100000000 01

B - INPUT MATRIX

0.0	0.C
0.0	0.C
0.0	0.0
0.150000000 02	-0.750CC0000 01
0.0	0.0
0.462500000 01	0.138750000 02

DESIRED EIGENVALUES OF %A - B*G<

-10.00000000	0.0
-12.00CCC000	0.0
-14.000C0C00	0.0
-16.00C0C000	0.0
-18.000C0C00	0.0
-20.00C00000	0.0

MATRIX SINV.

0.102439020 01	0.716282980-30	0.117073170 01	0.195121950 00	0.200000000 01	0.666500500-29
-0.219512200 00	-0.187517980-28	0.463414630 00	0.243902440 00	0.111022300-15	-0.464034330-30
0.714285710 00	0.256217420-24	0.428571430 00	0.142857140 00	0.100000000 01	-0.242732290-24
0.937500000 00	0.965534830-33	0.437500000 00	0.125000000 00	0.100C00000 01	-0.861152690-33
0.0	0.200C00000 01	0.0	0.0	0.0	0.108108110 00
0.0	0.832667270-16	0.0	0.0	0.0	0.648648650 00

MATRIX SINV*B .

0.292682930 01	-0.146341460 01
0.365853660 01	-0.182926830 01
0.214285710 01	-0.107142860 01
0.187500000 01	-0.9375C0000 00
0.500000000 00	0.150C00000 01
0.300000000 01	0.900C00000 01

VECTOR ALPHA%TRANS< %VECTOR D # B*ALPHA<.

0.1C0C00000 01	C.100C00000 01

VECTOR D%TRANS<.

0.146341460 01	0.182926830 01	0.107142860 01	0.937500000 00	0.200000000 01	0.12000C000 02

DIAGONALIZED MATRIX A, OR DIAGONALIZED MATRIX %A-B*ZG< IN THE CASE OF MULTIPLE EIGENVALUES AND/OR COMMON OPEN- AND CLOSED-LOOP EIGENVALUES.

-0.100000000 01	C.200C0C000 01	0.0	0.0	0.0	0.0
-0.200000000 01	-C.100C00000 01	0.0	0.0	0.0	0.0
0.0	0.0	-0.150000000 01	0.0	0.0	0.0
0.0	C.0	0.0	-0.100000000 01	0.0	0.0
0.0	C.0	0.0	0.0	-0.5CCCC0000 00	0.300000000 01
0.0	0.C	0.0	0.0	-0.300000000 01	-0.500000000 00

G - GAIN MA.RIX

0.14997108D 06	-0.52433393D 04	-0.53018003D 04	-0.53218963D 04	0.60109883D 05	0.21620931D 04
0.14997108D 06	-0.52433393D 04	-0.53018003D 04	-0.53218963D 04	0.60109883D 05	0.21620931D 04

T - SOLUTION MATRIX

-0.11190818D 00	-0.99512195D-01	-0.88819963C-01	-0.79880711D-01	-0.72421543D-01	-0.66154360C-01
-0.22812052D 00	-0.18439024D 00	-0.15437756D 00	-0.13260198D 00	-0.11612420D 00	-0.10324C90D 00
-0.12605042D 00	-0.10204082D 00	-0.85714286D-01	-0.73891626D-01	-0.64935065D-01	-0.57915058C-01
-0.10416667D 00	-0.85227273D-01	-0.72115385D-01	-0.6250C000D-01	-0.55147059D-01	-0.49342105C-01
0.17128463D 00	0.92035398D-01	0.47058824D-01	0.20060181D-01	0.31720856D-02	-0.77071291C-02
-0.12090680D 01	-0.10194690D 01	-0.87843137D 00	-0.77031093D 00	-0.68517050D 00	-0.61657C33C 00

COMPUTED EIGENVALUES OF %A - B*G<.

-20.00007635	0.0
-15.99962899	0.0
-17.99999731	0.0
-14.00048892	0.0
-11.99977490	0.0
-10.00003358	0.0

MATRIX Q .

0.25000000D 01	0.0	0.0	0.0	0.0	0.0
0.0	0.250C0000D 01	0.0	0.0	0.0	0.0
0.0	0.0	0.25000000D 01	0.0	0.0	0.0
0.0	0.0	0.0	0.25000000D 01	0.0	0.0
0.0	0.0	0.0	0.0	0.25000000D 01	0.0
0.0	0.0	0.0	0.0	0.0	0.25000000D 01

MATRIX R .

0.50000000D 00	0.0
0.0	0.500C0000D 00

GAIN TOLERANCE .LE. 0.28361440-04WAS ACHEIVED AFTER 33 ITERATIONS.

MATRIX R%INVERSE< .

0.20000000D 01	0.0
0.0	0.200C0000D 01

RICCATI MATRIX P .

0.61086881D 01	0.83056404D-02	0.26525001D 01	0.54934096D-01	0.62062541D 01	0.35134024D-02
0.83056404D-02	C.26388867D 01	0.12715444D-01	0.26612129D-02	0.18359872D-01	0.58077758D-01
0.26525001D 01	0.12715444D-01	0.64035913C 01	0.14673619D 00	0.64093552D 01	0.10595823C-01
0.54934096D-01	0.26612129D-02	0.14673619D 00	0.63324688D-01	0.13834560D 00	0.47503134C-02
0.62062541D 01	0.18359872D-01	0.64093552D 01	0.13834560D 00	0.12477264D 02	0.93589116C-02
0.35134024D-02	G.58077758D-01	0.10595623D-01	0.47503134D-02	0.93589116D-02	0.76458345D-01

RESIDUAL MATRIX. MATRIX IS ZERO, IF MATRIX P IS ACCURATE.

-0.13254184D-08	0.32670339D-10	-0.17969625C-08	-0.46035582D-10	-0.28096348D-08	-0.27462477C-11
0.32670505D-10	-0.40600856D-11	0.44381124D-10	0.11641660D-11	0.69947534D-10	0.61950445D-13
-0.17969650D-08	C.44382123D-10	-0.24364663D-08	-0.62483125D-10	-0.38095166D-08	-0.37139458C-11
-0.46036703D-10	0.11641521D-11	-0.62483079D-10	-0.15861184D-11	-0.97629246D-10	-0.63921C91C-13
-0.28096360D-08	0.69948519D-10	-0.38095137C-08	-0.97628246D-10	-0.59562816D-08	-0.58083122C-11
-0.27459146D-11	0.62172489D-13	-0.37147924C-11	-0.63712924D-13	-0.58091865D-11	0.75273121C-13

STATES # 6 INPUTS # 3

A - SYSTEM MATRIX

0.20C000000 02	-C.100C00000 01	0.50000000C 00	0.0	0.6CCCCC000 01	0.0
0.0	0.0	-0.40000000C 00	0.1300C0000 02	0.7C0CC0000 02	0.20C00C00C 00
-0.130000000 02	0.200C00000 01	-0.900000000 01	0.0	0.0	0.400C0CC00 02
0.400000000 01	-0.110C00000 02	0.0	0.100000000 01	0.500C00000 00	0.10000C000 01
0.500000000 01	0.100C00000 00	0.0	-0.4000C0C00 01	0.200C00000 01	0.0
0.0	0.0	0.600000000 01	0.300000000 00	0.0	-0.40000C000 02

B - INPUT MATRIX

0.100000000 01	C.0	0.0
0.0	-0.400C00000 01	0.0
-0.2C0000000 01	0.200C00000 01	0.0
0.500000000 01	C.100C00000 01	0.0
0.70C000000 00	0.0	0.100000000 02
0.0	0.400C00000 01	0.0

DESIRED EIGENVALUES OF %A - B*G<

-1.C0C00C00	0.0
-2.00CC0G00	0.0
-3.00000000	1.00000000
-3.00C00000	-1.00000000
-5.00000C00	0.0
-6.00000000	0.0

MATRIX SINV.

0.100000000 01	-0.11856288D-01	0.19227209D-01	-0.62450073D-01	0.27243753D 00	0.11571457D-01
-0.33679396D 00	-0.17332179D 00	-0.15426251D-01	0.57254225D 00	0.200CC0000 01	0.40558298D-02
0.27467012D 00	-0.44604159D 00	-0.80541000D-03	-0.45997716D 00	0.0	-0.14856352D-01
-0.57980145D 00	0.18888515D 00	-0.23210706D-01	-0.14457030D 00	0.100000000 01	-0.20014C000-01
-0.30659479D-01	0.44023138D-02	-0.15995003C 00	-0.77454897D-02	-0.24859556D-02	0.100000000 01
0.52307316D 00	-0.46714529D-01	0.93950837D 00	0.11359208D 00	0.17329039D-01	0.100C00000 01

MATRIX SINV*B .

0.84000149D 00	0.69715328D-01	0.27243753D 01
0.39567698D 01	0.12512002D 01	0.200000000 02
-0.20236049D 01	0.12631530D 01	0.0
-0.54723152D 00	-0.10265883D 01	0.10000000C 02
0.24877296D 00	0.36547452D 01	-0.24859556D-01
-0.77585284D 00	0.61794669D 01	0.17329039C 00

VECTOR ALPHA%TRANS< %VECTOR D # B*ALPHA<.

0.100000000 01	0.100C00000 01	0.100000000 01

VECTOR D%TRANS<.

0.36340921D 01	0.25207970D 02	-0.76045189C 00	0.84261801D 01	0.38786586D 01	0.55769045D 01

DIAGONALIZED MATRIX A, OR DIAGONALIZED MATRIX %A-B*G< IN THE CASE
OF MULTIPLE EIGENVALUES AND/OR COMMON OPEN- AND CLOSED-LOOP EIGENVALUES.

0.20867434D 02	C.0	0.0	C.0	0.0	0.0
0.0	-0.49335091D 01	0.14902440D 02	0.0	0.0	0.0
0.0	-0.14902440D 02	-0.49335091D 01	0.0	0.0	0.0
0.0	C.0	0.0	0.11724867D 02	0.0	0.0
0.0	C.C	0.0	0.0	-0.46404866D 02	0.0
0.0	0.0	0.0	0.0	0.0	-0.23154162D 01

G - GAIN MATRIX

0.467925030 01	-0.173796670 00	0.874080560 00	-0.201614750 00	0.859433660 00	-0.484017220 01
0.467925030 01	-0.173796670 00	0.87408056C 00	-0.201614750 00	0.859433660 00	-0.48401722C 01
0.46792503D 01	-0.17375667D 00	0.87408056C 00	-0.20161475D 00	0.85943366D 00	-0.48401722D 01

T - SOLUTION MATRIX

-0.16622541D 00	-0.15895474D 00	-0.14565550D 00	-0.15839741D 00	-0.14051625D 00	-0.13528529D 00
0.36969593D 00	0.27142795D 00	0.59035202D-01	0.27749581D 00	-0.[illegible]8574706D-01	-0.17120616C 00
-0.15939516D 01	-0.16381032D 01	-0.17024944D 01	-0.16515681D 01	-0.16912718D 01	-0.16792808D 01
-0.66218218D 00	-0.61393529D 00	-0.53093056D 00	-0.60829824D 00	-0.50381149D 00	-0.47538750D 00
0.85423853D-01	0.67347603D-01	0.91370251C-01	0.87254925D-01	0.93676395D-01	0.95994839C-01
0.42396501D 01	0.17681097D 02	0.11977259C 01	-0.6[illegible]968482D 01	-0.20773814D 01	-0.15135779C 01

COMPUTED EIGENVALUES OF %A - B*G<.

-6.00000C00	0.0
-5.00000000	0.0
-3.00000C00	1.0C000000
-3.00000C00	-1.00000000
-2.000C0C00	0.0
-1.000C0000	0.0

MATRIX Q .

0.10000000D 02	0.C	0.0	0.0	0.0	0.0
0.0	C.200C0000D 01	0.0	0.0	0.0	0.0
0.0	0.0	0.80000000D 00	0.0	0.0	0.0
0.0	0.C	0.0	0.10000000D 01	0.0	0.0
0.0	0.0	0.0	0.0	0.100C0000D 01	0.0
0.0	0.0	0.0	0.0	0.0	0.16000000C 01

MATRIX R .

0.20000000D 00	0.0	0.0
0.0	0.400C0000D 00	0.0
0.0	0.0	0.50000000D 00

GAIN TOLERANCE .LE. 0.54235050D-06WAS ACHEIVED AFTER 28 ITERATIONS.

MATRIX R%INVERSE< .

0.50000000D 01	0.0	0.0
0.0	0.250C0000D 01	0.0
0.0	0.0	0.20000000D 01

RICCATI MATRIX P .

0.22208273D 02	-0.80507645D 00	0.42667526D 00	-0.20882124D 01	0.97050629D 00	0.20803720D 00
-0.80507645D 00	0.10987551D 00	-0.11096976D-01	0.65345154D-01	0.66710161D-01	-0.25462732D-02
0.42667526D 00	-0.11096976D-01	0.71396814C-01	-0.33316589D-01	0.29647119D-01	0.57281182C-01
-0.20882124D 01	0.65345154D-01	-0.33316589D-01	0.29155776D 00	-0.10136244D 00	-0.11376245C-01
0.97050629D 00	0.66710161D-01	0.29647119D-01	-0.10136244D 00	0.32452990D 00	0.21568518C-01
0.20803720D 00	-0.25462732D-02	0.572811[illegible]2D-01	-0.11376245D-01	0.21568518D-01	0.70744858C-01

RESIDUAL MATRIX. MATRIX IS ZERO, IF MATRIX P IS ACCURATE.

0.79935172D-08	-0.64117668D-09	0.15919153D-09	-0.42998935D-09	0.55991586D-10	0.58348C90C-10
-0.64117088D-09	0.21136632D-10	-0.12178006D-10	0.42129238D-10	-0.30697988D-10	-0.51795773C-11
0.15920021D-09	-0.12178203D-10	0.34009878C-11	-0.99940655D-11	0.18202384D-11	0.14945684C-11
-0.43002708D-09	0.42130440D-10	-0.99943344D-11	0.13925088D-10	-0.1C643558D-1C	-0.42295585C-11
0.55997693D-10	-0.30697694D-10	0.18196278D-11	-0.10639286D-10	-0.5C466273D-10	-0.4742C4C1C-13
0.58352878D-10	-0.51796900D-11	0.14946377C-11	-0.42294553D-11	-0.46906923D-13	0.56403493C-12

As a point of interest it should be noted, that especially for higher order systems, i.e., $n \geq 6$, it is advisable to solve Kleinman's iteration equation, which has the form

$$P\, A_K + A_K^T\, P = -\, Q \tag{4.4-1}$$

where P and Q are symmetric matrices of order n, via an eigensystem approach. I.e., assuming, that A_K is a simple matrix, determine M and M^{-1}, the matrices of column and row eigenvectors of A_K, respectively. Let L denote the diagonal matrix of eigenvalues of A_K (or, to avoid complex arithmetic in the case of complex eigenvalues, near diagonal matrix, see eq. (4.2.3-14)), then equation (1) can be re-written to

$$(M^T\, P\, M)\, L + L^T\, (M^T\, P\, M) = -\, M^T\, Q\, M \tag{4.4-2}$$

$M^T PM \triangleq T$ and $M^T QM \triangleq D$ are symmetric matrices. Equation (2) can be solved in the same way as described in section 4.2.2. Moreover in this case use can be made of the symmetry properties of T and D, so that the linear equation

$$(\lambda_i\, I + L^T)\, \underline{t}_i = -\, \underline{d}_i \tag{4.4-3}$$

will be of order $(n-i+1)$; $\underline{t}_i$ and $\underline{d}_i$ denote the i^{th} column of matrices T and D, respectively.

The eigensystem approach was experienced to be trouble free as long as A_K did not contain multiple eigenvalues.

Equation (1) can also be solved by means of the Kronecker product[14], yielding an equation of form

$$E\, \underline{p}_v = -\, \underline{q}_v \tag{4.4-4}$$

Due to the symmetry of P and Q, E is 'only' of order $(n \times (n+1)/2)$ instead of n^2. $\underline{p}_v$ and $\underline{q}_v$ are vectors formed appropriately from the upper triangular elements of P and Q. Matrix E can be generated from matrix A_K using the expansion procedure suggested by Bingulac[44]. Due to its high order, e.g., for $n = 6$ E is of order 21, E may cause numerical difficulties and may be treated as numerically singular, even if it actually is not.

Although the latter method is about two times faster than the eigensystem approach in solving equation (1) numerical problems prevent it from being consistently useful for higher order systems $(n \geqslant 6)$.

Using the eigensystem approach to solve equation (1) all 11 presented examples were computed in 2 minutes on the IBM 360/50.

PART 5

SUMMARY AND CONCLUSION

This work considers the theoretical and numerical aspects of a compensator design for low-sensitivity systems. A new concept in sensitivity design is proposed, making use of the condition number $\varkappa = \inf_{M} \|M\| \|M^{-1}\|$. M is a matrix that transforms the closed-loop system matrix K to diagonal form L. The knowledge of $\varkappa$ is valuable in determining a bound on the permissible parameter uncertainty for which the closed-loop system will still exhibit a specified minimum stability. Generally this bound will be rather conservative. The sensitivity function derived in section 2.2 essentially maximizes the ratio $\left| \mathrm{Re}(\lambda_{max})/(\varkappa \|\delta K\|) \right|$ where λ_{max} is the least stable eigenvalue of the matrix K and δK is the parameter uncertainty matrix of K.

A computer program COMPDES was written to mechanize the low-sensitivity design procedure. The program is listed in Appendix A. Three design examples are given in Part 3. These examples are representative for the types of solutions possible when minimizing the sensitivity function described in section 2.2. As could be expected, the solutions indicate a decrease in permissible parameter uncertainty with increasing system order.

Part 4 presents a detailed description of two numerical algorithms. These algorithms were needed for the compensator design, but constitute general purpose algorithms. The first algorithm deals with pole-placement by all-state feedback. Use is made of techniques developed in state-estimation theory, resulting in a very fast and efficient numerical algorithm. The algorithm can be applied to finding stabilizing gains for the initialization of Kleinman's iterative scheme for the solution of the algebraic matrix Riccati equation. A computer program based on the suggested pole-assignment algorithm was written in FORTRAN IV and very successfully applied to many examples. Eleven of the examples are included in Part 3.

Since the sensitivity design is based on a new concept, much work remains to be done. It would be very useful to know what constitutes a 'good' value of κ for a system of a given order, i.e., to find some sort of standard. The compensator design was achieved by transforming the system to a pseudo single-input system. Can this be done in a way, such that the resulting closed-loop system has a lower condition number than a system designed by the methods presented in this work? The minimization of the sensitivity function will yield local results only, depending on the initial pole-location. Based on the results of one or more runs can systematical methods be developed to generate new sets of initial pole-locations yielding lower sensitivity of the closed-loop system?

Sensitivity analysis plays a large part in real system design. Thus, a method like the one presented, which desensitizes the closed-loop system for the variations of all system parameters, is a potentially very useful tool.

LITERATURE CITED

1. DeRusso, P.M., R.J. Roy and C.M. Close, "State Variables for Engineers," Wiley, New York, 1965.

2. Ash, R.H., Jr., "State Estimation in Linear Systems - A Unified Theory of Minimum Order Observers," Ph.D. Thesis, Systems Engineering Division, Rensselaer Polytechnic Institute, Troy, New York, 1969.

3. Wonham, W.H., "On Pole Assignment in Multi-Input Controllable Linear Systems," IEEE Trans. on Automatic Control, vol. AC-12, no. 6, December 1967.

4. Bass, R.W. and I. Gura, "High Order System Design via State-Space Considerations," JACC, Troy, New York, 1965.

5. Simon, J.D. and S.K. Mitter, "A Theory of Modal Control," Information and Control, Vol. 13, October 1968.

6. Gould, L.A., A.T. Murphy and E.F. Berkman, "On the Simon-Mitter Pole Allocation Algorithm - Explicit Gains for Repeated Eigenvalues," IEEE Trans. on Automatic Control, vol. AC-15, no. 2, April 1970.

7. Kalman, R.E. and T.S. Englar, "A Users Manual for the Automatic Synthesis Program," NASA, Report CR-475, June 1966.

8. Man, F.T., "The Davidon Method of Solution of the Algebraic Matrix Riccati Equation," Inst. Journal of Control, Vol. 10, no. 6, 1969.

9. Kleinman, D.L., "On an Iterative Technique for Riccati Equation Computation," IEEE Trans. on Automatic Control, vol. AC-13, no. 1, February 1968.

10. Kalman, R.E., "Contributions to the Theory of Optimal Control," Boletin de la Sociedad Matematica Mexicana, vol. 5, 1960.

11. Athans, M., and P.L. Falb, "Optimal Control," McGraw-Hill, New York, 1966.

12. Schultz, D.G., and J.L.Melsa, "State Functions and Linear Control Systems," McGraw-Hill, New York, 1967.

13. Luenberger, D.L., "Observing the State of a Linear System," IEEE Trans. on Military Electronics, vol. MIL-8, no. 2, April 1964.

14. Lancaster, P., "Theory of Matrices," Academic Press, New York, 1969.

15. Davison, E.J., Correspondence on "A Method for Simplifying Linear Dyanmic Systems," IEEE Trans. on Automatic Control, vol. AC-12, no. 1, Feb. 1967.

16. Ho, Y.C., "What Constitutes a Controllable System?" IRE Trans. on Automatic Control, vol. AC-7, April 1962.

17. Isaacson, E., and H.B. Keller, "Analysis of Numerical Methods," John Wiley and Sons, Inc, New York, 1966.

18. Brasch, F.M., and J.B. Pearson, "Pole Placement Using Dyanmic Compensation," IEEE Trans. on Automatic Control, vol. AC-15, no. 1, February 1970.

19. Pearson, J.B. and C.Y. Ding, "Compensator Design for Multi-Variable Linear Systems," IEEE Trans. on Automatic Control, vol. AC-14, no. 2, April 1969.

20. Ferguson, J.D. and Z.V. Rekasius, "Optimal Linear Control Systems with Incomplete State Measurements," IEEE Trans. on Automatic Control, vol. AC-14, no. 2, April 1969.

21. C.T. Chen, "Design of Pole-Placement Compensators for Multivariable Systems," Preprint JACC, Atlanta, Georgia, 1970.

22. Morse, W.S., and W.M. Wonham, "Decoupling and Pole Assignment by Dynamic Compensation," Preprint JACC, Atlanta, Georgia, 1970.

23. Wolovich, W.A., "On State Estimation of Observable Systems," Preprint JACC, Ann Arbor, Michigan, 1968.

24. Kalman, R.E., "A New Approach to Linear Filtering and Prediction Theory," J. of Basic Engrg., Trans. ASME, Series D, vol. 82, March 1960.

25. Moore, J.B., "A Note on Feedback Compensators in Optimal Linear Systems," IEEE Trans. on Automatic Control, vol. AC-15, no. 4, August 1970.

26. Luenberger, D.G., "Observers for Multi-Variable Systems," IEEE Trans. on Automatic Control, vol. AC-11, no. 2, April 1966.

27. Cassidy, J.F., "Optimal Control With Unavailable States," Ph.D. Thesis, Systems Engineering Division, Rensselaer Polytechnic Institute, Troy, New York, 1969.

28. Faddeev, D.N. and V.N. Faddeeva, "Computational Methods of Linear Algebra," Freeman, 1963.

29. Shinners, S.M., "Control System Design," Wiley and Son, Inc., 1964.

30. Letov, A.M., "Analytical Controller Design I," Automatika i Telemekkanika, vol. 21, no. 4, 1960.

31. Davison, E.J., "On Pole Assignment in Linear Systems With Incomplete State Feedback," Shortpaper, IEEE Trans. on Automatic Control, vol. AC-15, no. 3, June 1970.

32. Jameson, A., "Design of a Single-Input System for Specified Roots Using Output Feedback," Short paper, IEEE Trans. on Automatic Control, vol. AC-15, no. 3, June 1970.

33. Kalman, R.E. and R.S. Bucy, "New Results in Linear Filtering and Prediction Theory," Trans. ASME, Series D, Journal of Basic Engineering, vol. 83, 1961.

34. Porter, W.A., "Sensitivity Problems in Linear Systems," IEEE Trans. on Automatic Control, vol. AC-10, no. 3, July 1965.

35. Rohrer, R.A. and M. Sobral, Jr., "Sensitivity Considerations in Optimal System Design," IEEE Trans. on Automatic Control, vol. AC-10, no. 1, Jan. 1965.

36. Takahashi, Y., M.J. Rabins and D.M. Auslander, "Control and Dynamic Systems," Addison-Wesley Publishing Company, Massachusetts, 1970.

37. Tuel, W., Jr., "Optimal Control of Unknown Systems," Ph.D. Thesis, Department of Electrical Engineering, Rensselaer Polytechnic Institute, Troy, New York, 1965.

38. Bode, H.W., "Network Analysis and Feedback Amplifier Design," New York, Van Nostrand, 1945.

39. Cruz, J.B., Jr., and W.R. Perkins, "A New Approach to the Sensitivity Problem in Multivariable Feedback System Design," IEEE Trans. on Automatic Control, vol. AC-9, July 1964.

40. Kwakernaak, H., "Optimal Low-Sensitivity Linear Feedback Systems," Automatica, vol. 5, no. 3, May 1969.

41. Morgan, B.S., "Sensitivity Analysis and Synthesis of Multi-variable Systems," IEEE Trans. on Automatic Control, vol. AC-11, no. 3, July 1966.

42. Dorato, R., P. Kokotovic and H. Kwakernaak, "Report on the Second IFAC Symposium on System Sensitivity and Adaptivity," Automatica, vol. 5, no. 3, May 1969.

43. McClamroch, N.H., L.G. Clark and J.K. Aggarwal, "Sensitivity of Linear Control Systems to Large Parameter Variations," Automatica, vol. 5, no. 3, May 1969.

44. Bingulac, S.P., "An Alternate Approach to Expanding $PA + A^{T}P = - Q$," IEEE Trans. on Automatic Control, vol. AC-15, no. 1, February 1970.

APPENDIX A

Listing of the Compensator Design Computer Program COMPDES

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                                   DATE  71.119/00.33.

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                           SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
                  C
                  C      PROGRAM NAME  @COMPDES@
                  C
                  C      THIS PROGRAM DESIGNS A LOW ORDER CONSTANT FEEDBACK COMPENSATOR
                  C      FOR A TIME-INVARIANT SYSTEM WITH LARGE PARAMETER UNCERTAINTIES.
                  C      THE RESULTING COMPENSATOR WILL EITHER GUARANTEE SYSTEM STABILITY
                  C      FOR  THE SPECIFIED PARAMETER UNCERTAINTIES OR WILL GIVE A BOUND,
                  C      UP TO WHICH THE SYSTEM IS STABLE .
                  C
   ISN 0002              REAL*8 A%6,6<,B%6,6<,C%6,6<,DA%6,6<,AH%12,12<,A2%12,12<,A2S%12,12<
   ISN 0003              REAL*8 AV%144<
   ISN 0004              REAL*8 FF%6,6<,FG%6,6<,FH%6,6<,FJ%6,6<,GF%6,6<,GG%6,6<,GH%6,6<
   ISN 0005              REAL*8 GJ%6,6<,GFM%6,6<,GGM%6,6<,GHM%6,6<,GJM%6,6<
   ISN 0006              REAL*8 GRF%6,6<,GRG%6,6<,GRH%6,6<,GRJ%6,5<,CVR%12,12<,CVI%12,12<
   ISN 0007              REAL*8 RR%12<,RI%12<,VR%12<,VI%12<,XR%12<,XI%12<,VRN%12,1<
   ISN 0008              REAL*8 VIN%12,1<,EMS%6,2<,EMF%6,2<,VER,VEI,VES,W%12,4<
   ISN 0009              REAL*8 EM(12,2)
   ISN 0010              REAL*8 AVF%36<,XRS%6<,XRF%6<,XIS%6<,XIF%5<,VRS%6<,VRF%6<,VIS%6<
   ISN 0011              REAL*8 VIF%6<,RRS%6<,RRF%6<,RIS%6<,RIF%6<,VRNS%6,1<,VRNF%6,1<
   ISN 0012              REAL*8 VINS%6,1<,VINF%6,1<,AVS%36<,WS%6,4<,WF%6,4<
   ISN 0013              REAL*8 CX%13<,CG%13<,SV%12<,SVR%12<,H%13<,CF
   ISN 0014              REAL*8 ROOTR,ROOTI,DELKI,ROOTI,T3,ALO,SUMD
   ISN 0015              REAL*8 SW1,VABS,SUM,DABS,STOPR,UNUI,ST,STS,DELR,DELRI
   ISN 0016              REAL*8 DR,GA
   ISN 0017              REAL*8 AC
   ISN 0018              DIMENSION IANA%12<,IANAS%6<,IANAF%6<,IROW%12,2<,IROWS%6,2<
   ISN 0019              DIMENSION IROWF%6,2<,ICPLX%13<
   ISN 0020              EQUIVALENCE %AV%1<,AVF%1<<,%AV%37<,AVS%1<<
   ISN 0021              EQUIVALENCE %EM%1<,EMF%1<<,%EM%13<,EMS%1<<
   ISN 0022              EQUIVALENCE %XR%1<,XRS%1<<,%XR%7<,XRF%1<<,%XI%1<,XIS%1<<
   ISN 0023              EQUIVALENCE %XI%7<,XIF%1<<,%VR%1<,VRS%1<<,%VR%7<,VRF%1<<
   ISN 0024              EQUIVALENCE %VI%1<,VIS%1<<,%VI%7<,VIF%1<<,%RR%1<,RRS%1<<
   ISN 0025              EQUIVALENCF %RR%7<,RRF%1<<,%RI%1<,RIS%1<<,%RI%7<,RIF%1<<
   ISN 0026              EQUIVALENCE %VRN%1<,VRNS%1<<,%VRN%7<,VRNF%1<<
   ISN 0027              EQUIVALENCE %VIN%1<,VINS%1<<,%VIN%7<,VINF%1<<
   ISN 0028              EQUIVALENCE %W%1<,WF%1<<,%W%25<,WS%1<<
   ISN 0029              EQUIVALENCE %IANA%1<,IANAS%1<<,%IANA%7<,IANAF%1<<
   ISN 0030              EQUIVALENCE %IROW%1<,IROWS%1<<,%IROW%13<,IROWF%1<<
   ISN 0031              COMMON /MCM/ AH,A2,A2S,AV,CVR,CVI,W,A,B,C,FF,FG,FH,FJ,GF,GG,GH,
                        1             GJ,GFM,GGM,GHM,GJM,GRF,GRG,GRH,GRJ,EM,VRN,VIN,RR,RI
                        2             VR,VI,XR,XI,IROW,IANA
   ISN 0032              COMMON /MC/ SV,SVR,ROOTI,ALO,SUMD,DANORM,ACC,NS,NC,NF,NFF,MD,MD2
   ISN 0033              COMMCN /MC2/ DR,GA,ICPLX,IAREA,M2,M2D
   ISN 0034              EXTERNAL COND
   ISN 0035            5 FORMAT.1H1<
   ISN 0036           10 FORMAT %8I10<
   ISN 0037           20 FORMAT %4D18.7<
   ISN 0038           21 FORMAT%4D18.7<
   ISN 0039           22 FORMAT%4F20.10<
   ISN 0040           25 FORMAT %5F15.6<
   ISN 0041           30 FORMAT %7F10.4<
   ISN 0042           35 FORMAT%//T3,@STATES@,4X,@INPUTS@,4X,@OUTPUTS@,3X,@COMP-ORD@,2X,
                        1@IDELR@,5X,@ISTOP@,5X,@IMATJ@,5X,@NUI@,/T3,
                        2@NS #@,I3,3X,@NC #@,I3,3X,@NF #@,I3,3X,@NFF #@,I3,5X,I3,7X,I3,7X,
                        3I3,7X,I3<
   ISN 0043           36 FORMAT %//T3,@STABILITY INCREASE INCREMENT # @,F20.8<
   ISN 0044           37 FORMAT %//T3,@MINIMUM STABILITY REQUIRED FOR TERMINATION@,3X,F20.8
```

```
                    1<
ISN 0045       38 FC.MAT&//T3,@SYSTEM MATRIX  A.@<
ISN 0046       39 FORMAT&////T3,@GRADIENT PROCEDURE TRIES TO DETERMINE A COMPENSATOR
                 1 OF CRDER  NFF #@,I3,/T3,@SUCH THAT THE CLOSED-LOOP SYSTEM DOES NO
                 2T BECOME UNSTABLE FOR THE GIVEN@,/T3,@MAXIMUM PARAMETER UNCERTAINT
                 3IES ¤SEE MATRIX  DA<.@<
ISN 0047       40 FORMAT(//T3,'CLOSED-LOOP SYSTEM MATRIX  A(C.-L.).')
ISN 0048       41 FORMAT ¤//T3,@MAXIMUM ABSOLUTE CHANGE  DA  OF THE ELEMENTS OF THE
                 1SYSTEM MATRIX  A@<
ISN 0049       42 FORMAT ¤//T3,@OUTPUT MATRIX  C@<
ISN 0050       43 FORMAT&//T3,@ACCEPTABLE LARGEST REAL PART OF THE EIGENVALUES FOR T
                 1HE WCRST CASE OF@,/T3,@PARAMETER UNCERTAINTY HAS TO BE LESS THAN
                 2AC #@,F10.5,/T3,@EIGENVALUES OF NOMINAL SYSTEM ARE KEPT TO THE LEF
                 3T OF  ACC #@,F10.5<
ISN 0051       45 FORMAT ¤//T3,@CONTROL INPUT MATRIX B@<
ISN 0052     50   FORMAT ¤//T2,@MAXIMUM REAL PART OF ROOTS@<
ISN 0053       53 FORMAT&//T3,@ALO #@,F12.6,@COND.-NUMBER #@,F12.6<
ISN 0054       54 FORMAT&//T3,@GRADIENT METHOD PLACED POLES AS REQUIRED. ACTUAL COND
                 1.-NO. WILL BE COMPUTED.@<
ISN 0055       55 FORMAT&////T3,@DETERMINE COMPENSATOR OF ORDER  NFF #@,I3,/T3,
                 1@AND ITERATE ON CONDITION-NUMBER.@<
ISN 0056       56 FORMAT&////T3,@COMPENSATOR DESIGN - INITIAL VALUES.   ISHIFT #@,
                 1I3<
ISN 0057     60   FORMAT ¤//T3,@EIGVEC ERROR MESSAGES@<
ISN 0058     65   FORMAT ¤T3,@SWI#@,E10.4,10X,@ITER#@,I5,10X,@DIF#@,E10.4<
ISN 0059     70   FORMAT ¤//T3,@EIGENVECTORS CORRES TO MRP EIGENVALUE@<
ISN 0060     75   FORMAT ¤T6,@ROW REAL PART@,5X,@ROW IMAG PART@,5X,@COL REAL PART@,
                 1 5X,@COL IMAG PART@<
ISN 0061       76 FORMAT&//,@  SUCCESSFUL COMPENSATOR IS OF ORDER  NFF # @,I3<
ISN 0062       80 FORMAT&//,@  COMPENSATOR MATRIX F@<
ISN 0063       85 FORMAT&//,@  COMP. INPUT MATRIX  G.@<
ISN 0064       90 FORMAT&//,@  COMP. OUTPUT MATRIX  H.@<
ISN 0065       95 FORMAT&//,@  FEEDBACK MATRIX  J.@<
ISN 0066       96 FORMAT&//,@  IT IS DESIRED TO OBTAIN CLOSED-LOOP POLES LOCATED IN
                 1OR CLOSE TO A RECTANGULAR REGION@,/,@  OF WIDTH  DR  AND HEIGHT  2
                 2*DR  ,  DR #@,F20.10,/,@  WEIGHTING FACTOR FOR POLE LOCATIONS  GA
                 3#@,F12.8<
ISN 0067       97 FORMAT&////T3,@COMPENSATOR DESIGN - FINAL VALUES.@<
ISN 0068       98 FORMAT&//T3,@THE COMPENSATOR GIVEN BELOW GUARANTEES THE REQUIRED M
                 1INIMUM STABILITY  FOR THE MAXIMUM UNCERTAINTY@<
ISN 0069       99 FORMAT.//T3,@THE BELOW COMPENSATOR GUARANTEES STABILITY ONLY FOR A
                 1 TOTAL@,/T3,@UNCERTAINTY OF  @,F15.8<
ISN 0070          MD#6
ISN 0071          MD2#MC*MD
ISN 0072          M2D#2*MD
ISN 0073          M2#M2D*M2D
ISN 0074          M2D1#M2D&1
ISN 0075          MH#M2D*¤M2D&7<
ISN 0076          MH#MH/2
ISN 0077          READ ¤1,10< NS,NC,NF,NFF,IDELR,ISTOP,IMATJ,NU1
ISN 0078          WRITE¤3,5<
ISN 0079          WRITE ¤3,35< NS,NC,NF,NFF,IDELR,ISTOP,IMATJ,NU1
ISN 0080          IF¤IDELR< 120,120,110
ISN 0081      110 READ ¤1,30< DELR
ISN 0082          GO TO 130
ISN 0083      120 DELR#.2D0
ISN 0084      130 DELR1=DELR
ISN 0085          IF¤ISTOP< 150,150,140
```

```
  140 READ %1,30< STOPR
  150 CONTINUE
C
C       SYSTEM DATA MATRICES ARE READ IN BY ROWS
C
        READ %1,25< %%A%I,J<,J#1,NS<,I#1,NS<
        WRITE%3,38<
        WRITE %3,20< %%A%I,J<,J#1,NS<,I#1,NS<
        READ %1,25< %%DA%I,J<,J#1,NS<,I#1,NS<
        WRITE %3,41<
        WRITE %3,20< %%DA%I,J<,J#1,NS<,I#1,NS<
        READ %1,25< %%B%I,J<,J#1,NC<,I#1,NS<
        WRITE %3,45<
        WRITE%3,20< %%B%I,J<,J#1,NC<,I#1,NS<
        READ %1,25< %%C%I,J<,J#1,NS<,I#1,NF<
        WRITE %3,42<
        WRITE %3,20< %%C%I,J<,J#1,NS<,I#1,NF<
        READ%1,25< ACC,AC
        WRITE%3,43< AC,ACC
C
        KFIL#0
        NBEST#0
        NB1#0
        NBT2#0
        NCOUNT#1
        DANORM#0.0
        DO 153 I#1,NS
        DAN#0.0
        DO 152 J#1,NS
  152 DAN#DAN&DABS%DA%I,J<<
        DIFFN#DAN-DANORM
        IF%DIFFN .GT. 0.0< DANORM#DAN
  153 CONTINUE
C
C       INITIALIZATION OF THE DIRECT FEEDBACK MATRIX ELEMENTS
C       IF IMATJ # 0, FJ%I,J< # 0.0
C       IF IMATJ # 1, READ  FJ%I,J<
C
        IF%IMATJ< 154,154,156
  154 CONTINUE
        DO 155 I#1,NC
        DO 155 J#1,NF
  155 FJ%I,J<#0.D0
        GO TO 157
  156 CONTINUE
        READ %1,25< %%FJ%I,J<,J#1,NF<,I#1,NC<
C
  157 CONTINUE
        IA#NS&NFF
        STS#%IA&NU1<*DANORM-ACC
        IF%ISTOP .EQ. 0< STOPR#-STS
        WRITE %3,37< STOPR
        DELR#DELR1
        WRITE%3,36< DELR
C
        WRITE%3,39< NFF
  190 KOUNT#1
        IWRITE=2
```

PAGE

```
ISN 0135         CALL STAH%A,B,C,FF,FG,FH,FJ,AH,AV,RR,RI,NS,NC,NF,NFF,MD,M2D,M2,
                $           IANA,IWRITE<
ISN 0136         CALL MAXRT%RR,RI,NS,NFF,M2D<
ISN 0137         IF%RR%1<-STOPR< 999,198,198
ISN 0138     198 ROOTR#RR%1<
ISN 0139         ROOTI#RI%1<
ISN 0140         IF( IMATJ .EQ. 0) GO TO 1099
ISN 0142         CALL GMR%FF,FG,FH,FJ,GFM,GGM,GHM,GJM,NC,NF,NFF,MD<
ISN 0143         NBEST#NBEST&1
ISN 0144     200 KOUNT#KOUNT&1
ISN 0145         IF%NB1-NBEST< 201,202,202
ISN 0146     201 NB1#NBEST
ISN 0147         GO TO 203
ISN 0148     202 IF%NBT2 .EQ. 1< GO TO 1099
ISN 0150         NBT2#1
ISN 0151     203 CONTINUE
ISN 0152         IF%KOUNT-10< 210,210,1099
ISN 0153     210 CONTINUE
ISN 0154         ROOTI=RR(1)
ISN 0155         WRITE%3,50<
ISN 0156         WRITE(3,20) ROOTI
            C
            C    COMPUTATION OF EIGENVECTORS
            C
ISN 0157         CALL MMULT%AH,AH,A2,IA,IA,IA,M2D<
ISN 0158         WRITE %3,40<
ISN 0159         WRITE %3,20< %%AH%I,J<,J#1,IA<,I#1,IA<
ISN 0160         NEIG#0
ISN 0161     215 CONTINUE
ISN 0162         DO 220 I#1,IA
ISN 0163         DO 220 J#1,IA
ISN 0164     220 A2S%I,J<#A2%I,J<
ISN 0165         CALL EIGVEC%3,AH,A2S,W,IROW,XR,XI,VR,VI,RR%1<,RI%1<,IA,M2D,0,
                1SW1,ITER,DIF,2<
ISN 0166         NEIG#NEIG&1
ISN 0167         IF%ITER-10< 230,230,225
ISN 0168     225 IF%%NEIG-5< .GE. 0< GO TO 230
ISN 0170         CALL RAYL%AH,RR%1<,RI%1<,XR,XI,VR,VI,IA,M2D<
ISN 0171         GO TO 215
ISN 0172     230 CONTINUE
            C
            C    SW1 # 0 FOR AN EXACT EIGENVALUE AND NO ROUND-OFF ERROR
            C    ITER # NUMBER OF ITERATIONS USED TO FIND EIGENVECTORS.
            C    IF TOLERANCE IS NOT ACHIEVED, PROGRAM ACCEPTS VALUES AT ITER # 15.
            C    DIF # LARGEST CHANGE IN ANY EIGENVECTOR COMPONENT AT FINAL ITER.
            C
ISN 0173         WRITE %3,60<
ISN 0174         WRITE %3,65< SW1,ITER,DIF
ISN 0175         WRITE %3,70<
ISN 0176         WRITE %3,75<
ISN 0177         WRITE %3,20< %VR%I<,VI%I<,XR%I<,XI%I<,I#1,IA<
            C
            C    NORMALISE EIGENVECTORS INNER PRODUCT
            C
ISN 0178         VER#0.D0
ISN 0179         VEI#0.D0
ISN 0180         DO 240 I#1,IA
ISN 0181         VER#VER&VR%I<*XR%I<-VI%I<*XI%I<
```

```
ISN 0182      240  VFI#VEI&VR%I<*XI%I<&VI%I<*XR%I<
ISN 0183           VES#VER*VER&VEI*VEI
ISN 0184           DO 250 I#1,IA
ISN 0185           VRN%I,1<#%VR%I<*VER&VI%I<*VEI</VES
ISN 0186       250 VIN%I,1<#%VI%I<*VER-VR%I<*VEI</VES
              C
              C    COMPUTE GRADIENT MATRIX
              C
ISN 0187           CALL GRADM%B,C,DELR,DELRI,GJ,GRF,GRG,GRH,GRJ,VIN,VRN,XI,XR,NS,NC,
                  S            NF,NFF,MD<
ISN 0188           DO 305 J#1,NC
ISN 0189           DO 305 L#1,NF
ISN 0190       305 GJ%J,L<#FJ%J,L<
ISN 0191           IF%NFF .EQ. 0< GO TO 325
ISN 0193           DO 310 J#1,NC
ISN 0194           DO 310 L#1,NFF
ISN 0195       310 GH%J,L<#FH%J,L<
ISN 0196           DO 315 J#1,NFF
ISN 0197           DO 315 L#1,NF
ISN 0198       315 GG%J,L<#FG%J,L<
ISN 0199           DO 320 J#1,NFF
ISN 0200           DO 320 L#1,NFF
ISN 0201       320 GF%J,L<#FF%J,L<
ISN 0202       325 CONTINUE
              C
              C    FIND MAXIMUM %OR REQUESTED< STABILITY, USING AN APPROXIMATION OF
              C         THE ACTUAL  GRADIENT. QUADRATIC CURVE-FITTING TO FIND MAXIMUM.
              C
ISN 0203           CALL APPROX%A,B,C,AH,AV,RR,RI,FF,FG,FH,FJ,GF,GG,GH,GJ,GFM,GGM,GHM,
                  A            GJM,GRF,GRG,GRH,GRJ,NS,NC,NF,NFF,MD,M2D,MD2,M2,IANA,
                  B            STOPR,DELRI,DELR,DELR1,ITEST,ROOTI,ROOTR,ROOTI,NBEST<
ISN 0204           IF%ITEST-1< 899,899,999
ISN 0205       899 CONTINUE
ISN 0206           IWRITE=2
ISN 0207           CALL GMR%GFM,GGM,GHM,GJM,FF,FG,FH,FJ,NC,NF,NFF,MD<
ISN 0208           CALL STAB%A,B,C,FF,FG,FH,FJ,AH,AV,RR,RI,NS,NC,NF,NFF,MD,M2D,M2,
                  S            IANA,IWRITE<
ISN 0209           RR%1<#ROOTR
ISN 0210           RI%1<#ROOTI
ISN 0211           GO TO 200
              C
              C    GRADIENT METHOD COULD PLACE POLES SUCH THAT THE ESTIMATE OF THE
              C         COND. NUMBER IS SATISFIED. COMPUTE ACTUAL COND. NUMBER.
              C
ISN 0212       999 KFIL#1
ISN 0213           WRITE%3,54<
ISN 0214           ISHIFT#0
ISN 0215           GO TO 400
              C
              C    GRADIENT METHOD DID NOT SUCCEED. THE METHOD GOT EITHER STUCK ON A
              C         LOCAL STABILITY MAXIMUM OR EXCEEDED 10 ITERATION STEPS.
              C
ISN 0216      1099 CONTINUE
ISN 0217           NFF#NS-NF
ISN 0218           ISHIFT=1
ISN 0219           WRITE%3,55< NFF
ISN 0220           IA#NS&NFF
ISN 0221           STS#%IA&NU1<*DANORM-ACC
```

```
ISN 0222          IF%ISTUP .EQ. 0< STOPR#-STS
ISN 0224          WRITE%3,37< STOPR
ISN 0225          READ(1,10) IPOLE,IAREA,LIMIT,KSHIFT
ISN 0226          IF%IAREA< 342,342,341
ISN 0227      341 READ%1,22< DR,GA
ISN 0228          WRITE%3,96< DR,GA
ISN 0229      342 CONTINUE
ISN 0230          IF%IPOLE< 350,350,340
ISN 0231      340 READ%1,22< %%EMS%I,J<,J#1,2<,I#1,NS<
ISN 0232          T3#EMS%NS,1<
ISN 0233          K1#0
ISN 0234          K2#0
ISN 0235          IF%NFF .LE. 0< GO TO 345
ISN 0237          READ%1,22< %%EMF%I,J<,J#1,2<,I#1,NFF<
ISN 0238          T3#EMF%NFF,1<
ISN 0239          K1#1
ISN 0240          K2#NFF
ISN 0241      345 CONTINUE
ISN 0242          GO TC 385
ISN 0243      350 NFF#NS-NF
ISN 0244          KFIL#0
ISN 0245          T3#STOPR/3.D0
ISN 0246      352 ISHIFT#ISHIFT&1
ISN 0247          IF(ISHIFT .GT. KSHIFT) GO TO 500
ISN 0249          NRC#2
ISN 0250          CALL INPOL%T3,RRS,RIS,NS,NRC,MD<
ISN 0251          DO 355 I#1,NS
ISN 0252          EMS%I,1<#RRS%I<
ISN 0253      355 EMS%I,2<#RIS%I<
ISN 0254          IF%NFF< 380,380,370
ISN 0255      370 CALL INPOL%T3,RRF,RIF,NFF,NRC,MD<
ISN 0256          DO 375 I#1,NFF
ISN 0257          EMF%I,1<#RRF%I<
ISN 0258      375 EMF%I,2<#RIF%I<
ISN 0259          K1#1
ISN 0260          K2#NFF
ISN 0261          GO TC 385
ISN 0262      380 CONTINUE
ISN 0263          K1#0
ISN 0264          K2#0
ISN 0265      385 CALL MCOPP%NS,NC,NF,NFF,MD,MD2,O,K1,K2,1,NS<
            C     DO 390 I#1,NS
            C 390 WRITE%3,21< %EMS%I,J<,J#1,2<
            C     IF%NFF .LT. 1< GO TO 400
            C     DO 395 I#1,NFF
            C 395 WRITE%3,21< %EMF%I,J<,J#1,2<
ISN 0266      400 CONTINUE
ISN 0267          IA#NS&NFF
ISN 0268          WRITE%3,56< ISHIFT
ISN 0269          IWRITE#2
ISN 0270      410 CALL STAB%A,B,C,FF,FG,FH,FJ,AH,AV,RR,RI,VS,NC,NF,NFF,MD,M2D,M2,
                 S             IANA,IWRITE<
ISN 0271          CALL MMULT%AH,AH,A2,IA,IA,IA,M2D<
ISN 0272          IVC#3
ISN 0273          CALL SIMTR2%AH,A2,A2S,CVR,CVI,W,IROW,RR,RI,XR,XI,SV,SVR,IA,M2D,
                 S             IVC<
ISN 0274          DO 415 I#1,IA
ISN 0275          XR%I<#0.D0
```

```
          DO 415 J#1,IA
      415 XR%I<#XR%I<GDABS%CVR%I,J<<
          SUMO#XR%1<
          DO 42C I#1,IA
          IF%SUMO .LT. XR%I<< SUMO#XR%I<
      420 CONTINUE
          CALL SURT%EMS,EMF,RR,RI,SV,NS,NFF,MD,M2D<
          ALO#1.DO
          WRITE%3,53< ALO,SUMO
          ST#SUMO*DANORM-ACC
          CALL MAKRT%RR,RI,NS,NFF,M2D<
          IF%RR%1<GST< 425,425,450
C
C         COMPENSATOR DESIGN WAS SUCCESSFUL.
C
      425 WRITE%3,76< NFF
          WRITE%3,80<
          DO 430 I#1,NFF
      430 WRITE%3,21< %FF%I,J<,J#1,NFF<
          WRITE%3,85<
          DO 435 I#1,NFF
      435 WRITE%3,21< %FG%I,J<,J#1,NF<
          WRITE%3,90<
          DO 440 I#1,NC
      440 WRITE%3,21< %FH%I,J<,J#1,NFF<
          WRITE%3,95<
          DO 445 I#1,NC
      445 WRITE%3,21< %FJ%I,J<,J#1,NF<
          STOP
C
C         ITERATE ON COND. NUMBER.
      450 IF%KFIL .EQ. 1< GO TO 1099
          CALL ASSIGN%CX,EMS,EMF,ALO,ICPLX,NS,NFF,MD,M2D1<
          IA1#IA-1
          CALL DFMFP%COND,IA1,CX,CF,CG,O.,1.E-4,LIMIT,IER,H,MH,ICPLX,M2D1<
          WRITE%3,10< IER
          ST#SUMO*DANORM-AC
          IF%RCOI%1<GST< 492,492,494
      492 WRITE%3,98<
          GO TO 496
      494 XI%I<#%AC-ROOTI</SUMO
          WRITE%3,99< XI%I<
      496 CONTINUE
          WRITE%3,97<
          IWRITE#2
          CALL STAB%A,B,C,FF,FG,FH,FJ,AH,AV,RR,RI,VS,NC,NF,NFF,MD,M2D,M2,
         S          IANA,IWRITE<
          T3#IA*13/3.DO
          GO TO 352
      500 STOP
          END
```

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                              DATE  71.106/19.42.

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=CCCOK,
                           SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              SUBROUTINE AMHITAMAT,BMAT,CMAT,FMATF,FMATG,FMATH,FMATJ,AHAT,NS,
                       A                   NC,NF,NFF,MD,M2D<
                C
                C                                  % AMAT & BMAT*FMATJ*CMAT    BMAT*FMATH <
                C       COMPUTES     AHAT #  %                                           <
                C                                  % FMATG*CMAT                FMATF      <
                C
  ISN 0003              REAL*8 AMAT%MD,MD<,BMAT%MD,MD<,CMAT%MC,MD<,FMATF%MD,MD<,FMATG%MD,
                       1          MD<,FMATH%MD,MD<,FMATJ%MD,MD<,AHAT%M2D,M2D<
                C
                C       FOR SYSTEMS OF ORDER HIGHER THAN 6, CHANGE THE FOLLOWING REAL*8
                C           STATEMENT
  ISN 0004              REAL*8 D%6,6<
                C
  ISN 0005              DO 10 I#1,NS
  ISN 0006              DO 10 J#1,NS
  ISN 0007           10 D%I,J<#0.DO
  ISN 0008              DO 20 I#1,NC
  ISN 0009              DO 20 J#1,NS
  ISN 0010              DO 20 L#1,NF
  ISN 0011           20 D%I,J<#D%I,J<&FMATJ%I,L<*CMAT%L,J<
  ISN 0012              DO 30 I#1,NS
  ISN 0013              DO 30 J#1,NS
  ISN 0014              AHAT%I,J<#AMAT%I,J<
  ISN 0015              DO 30 L#1,NC
  ISN 0016           30 AHAT%I,J<#AHAT%I,J<&BMAT%I,L<*D%L,J<
  ISN 0017              IF%NFF .LE. 0< GO TO 100
  ISN 0019              DO 40 I#1,NS
  ISN 0020              DO 40 J#1,NFF
  ISN 0021              JA#NS&J
  ISN 0022              AHAT%I,JA<#0.DO
  ISN 0023              DO 40 L#1,NC
  ISN 0024           40 AHAT%I,JA<#AHAT%I,JA<&BMAT%I,L<*FMATH%L,J<
  ISN 0025              DO 50 I#1,NFF
  ISN 0026              IA#NS&I
  ISN 0027              DO 50 J#1,NS
  ISN 0028              AHAT%IA,J<#0.DO
  ISN 0029              DO 50 L#1,NF
  ISN 0030           50 AHAT%IA,J<#AHAT%IA,J<&FMATG%I,L<*CMAT%L,J<
  ISN 0031              DO 60 I#1,NFF
  ISN 0032              IS#NS&I
  ISN 0033              DO 60 J#1,NFF
  ISN 0034              JS#NS&J
  ISN 0035           60 AHAT%IS,JS<#FMATF%I,J<
  ISN 0036          100 CONTINUE
  ISN 0037              RETURN
  ISN 0038              END
```

LEVEL 18 (SEPT 69)　　　　OS/360 FORTRAN H　　　　DATE 71.106/19.41.2

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=C000K,
                         SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              SUBROUTINE RAVLTA,E,EI,X,XI,V,VI,N,MD<
  ISN 0003              REAL*8 F,EI,X%MD<,XI%MD<,V%MD<,VI%MD<,A%MD,MD<,
                      1                     DVXR,CVXI,DVAXR,DVAXI,A1,A2,A3
  ISN 0004              REAL*8 DXDR%12<,DXDI%12<
                C
                C       FOR SYSTEMS OF ORDER HIGHER THAN 12 CHANGE THE FOLLOWING REAL*8
                C           STATEMENT
                C
  ISN 0005              REAL*8 DA%12,12<
                C
  ISN 0006       900    FORMAT%1X,5G12.4<
  ISN 0007              DO 1C I#1,N
  ISN 0008              DO 1C J#1,N
  ISN 0009        10    DA%I,J<#A%I,J<
  ISN 0010              DO 2C I#1,N
  ISN 0011        20    DA%I,I<#DA%I,I<-E
  ISN 0012              DVXR#C.0
  ISN 0013              DO 30 I#1,N
  ISN 0014              DVXR#DVXR&V%I<*X%I<
  ISN 0015              DXDR%I<#0.0
  ISN 0016              DO 3C L#1,N
  ISN 0017        30    DXDR%I<#DXDR%I<&DA%I,L<*X%L<
  ISN 0018              DVAXR#0.0
  ISN 0019              DO 40 I#1,N
  ISN 0020        40    DVAXR=DVAXR+V(I)*DXDR(I)
  ISN 0021              IF%EI< 60,50,60
  ISN 0022        50    F#E&DVAXR/DVXR
  ISN 0023              RETURN
  ISN 0024        60    DVXI#0.0
  ISN 0025              WRITE%3,8CC< DVXR
  ISN 0026              WRITE%3,800< %DXDR%LL<,LL#1,N<
  ISN 0027              DO 7C I#1,N
  ISN 0028              DVXR#DVXR-VI%I<*XI%I<
  ISN 0029        70    DVXI#DVXI&VI%I<*X%I<&V%I<*XI%I<
  ISN 0030              WRITE%3,800< DVXR,DVXI
  ISN 0031              DO 8C I#1,N
  ISN 0032              DXDI%I<#0.0
  ISN 0033              DO 8C J#1,N
  ISN 0034        80    DXDI%I<#DXDI%I<&DA%I,J<*XI%J<
  ISN 0035              WRITE%3,80C< %DXDI%LL<,LL#1,N<
  ISN 0036              A1#C.C
  ISN 0037              A2#0.C
  ISN 0038              A3#C.0
  ISN 0039              DO 9C I#1,N
  ISN 0040              A1#A1&VI%I<*DXDI%I<
  ISN 0041              A2#A2&VI%I<*DXDR%I<
  ISN 0042        90    A3#A3&V%I<*DXDI%I<
  ISN 0043              WRITE%3,800< A1,A2,A3
  ISN 0044              DVAXR#DVAXR&EI*DVXI-A1
  ISN 0045              DVAXI#A2&A3-EI*DVXR
  ISN 0046              A1#DVXR*DVXR&DVXI*DVXI
  ISN 0047              WRITE%3,800< DVAXR,DVAXI
  ISN 0048              WRITE%3,800< A1
  ISN 0049              E#E&%DVXR*DVAXR&DVXI*DVAXI</A1
  ISN 0050              EI#EI&%DVAXI*DVXR-DVAXR*DVXI</A1
```

```
ISN 0051            RETURN
ISN 0052            END
```



```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                           SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,IL,NOXREF
ISN 0002            SUBROUTINE GMR%FMATF,FMATG,FMATH,FMATJ,GFM,GGM,GHM,GJM,NC,NF,NFF,
                  A                  MD<
                C
                C   STORES FEEDBACK AND COMPENSATOR MATRICES OF THE EIGENVALUE WITH
                C   SMALLEST REAL PART THAT IS OBTAINED DURING THE ITERATION PROCESS.
                C
ISN 0003            REAL*8 FMATF%MD,MD<,FMATG%MD,MD<,FMATH%MD,MD<,FMATJ%MD,MD<,
                  1       GFM%MD,MD<,GGM%MD,MD<,GHM%MD,MD<,GJM%MD,MD<
ISN 0004            DO 800 J#1,NC
ISN 0005            DO 800 L#1,NF
ISN 0006        800 GJM%J,L<#FMATJ%J,L<
ISN 0007            IF%NFF .EQ. 0< GO TO 808
ISN 0009            DO 802 J#1,NC
ISN 0010            DO 802 L#1,NFF
ISN 0011        802 GHM%J,L<#FMATH%J,L<
ISN 0012            DO 804 J#1,NFF
ISN 0013            DO 804 L#1,NF
ISN 0014        804 GGM%J,L<#FMATG%J,L<
ISN 0015            DO 806 J#1,NFF
ISN 0016            DO 806 L#1,NFF
ISN 0017        806 GFM%J,L<#FMATF%J,L<
ISN 0018        808 CONTINUE
ISN 0019            RETURN
ISN 0020            END
```

```
LEVEL 18 ( SEPT 69 )                      OS/360  FORTRAN H                                    DATE  71.106/19.49.43

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                           SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002            SUBROUTINE GRADM%B,C,DELR,DELKI,GJ,GRF,GRG,GRH,GRJ,VIN,VRN,XI,XR,
                     S                   NS,NC,NF,NFF,MD<
                C
                C     COMPUTES GRADIENT OF LARGEST EIGENVALUE WITH RESPECT TO THE
                C         ELEMENTS OF THE COMPENSATOR MATRICES FF, FG, FH, FJ.
                C
                C     COMPUTED GRADIENT MATRICES ARE GRF, GRG, GRH, GRJ.
                C
  ISN 0003            REAL*8 B%MD,MD<,C%MD,MD<
  ISN 0004            REAL*8 GRF%MD,MD<,GRG%MD,MD<,GRH%MD,MD<,GRJ%MD,MD<,VIN%MD,1<,
                     1        VRN%MD,1<,XI%MD<,XR%MD<
  ISN 0005            REAL*8 SUM,DELR,DELKI
                C
                C     ROUTINE IS DIMENSIONAL FOR A MAXIMALLY 6-TH ORDER COMPENSATOR.
                C         FOR HIGHER ORDER COMP. CHANGE THE FOLLOWING REAL*8 STATEMENT.
  ISN 0006            REAL*8 GFI%6,6<,GFR%6,6<,GGI%6,6<,GGR%6,6<,GHI%6,6<,GHR%6,6<,
                     6        GJI%6,6<,GJR%6,6<
  ISN 0007            REAL*8 GJ%MD,MD)
                C
  ISN 0008         20 FORMAT%4D18.7<
  ISN 0009         80 FORMAT%//T3,@GRADIENT MATRIX DUE TO FJ@<
  ISN 0010         85 FORMAT%//T3,@GRADIENT MATRIX DUE TO FH@<
  ISN 0011         90 FORMAT%//T3,@GRADIENT MATRIX DUE TO FG@<
  ISN 0012         95 FORMAT%//T3,@GRADIENT MATRIX DUE TO FF@<
                C
  ISN 0013            SUM#0.D0
  ISN 0014            DO 265 J#1,NC
  ISN 0015            DO 265 L#1,NF
  ISN 0016            GJR%J,L<#0.D0
  ISN 0017            GJI%J,L<#0.D0
  ISN 0018            DO 255 I#1,NS
  ISN 0019            DO 255 II#1,NS
  ISN 0020        255 GJ%I,II<#B%I,J<*C%L,II<
  ISN 0021            DO 260 I#1,NS
  ISN 0022            DO 260 II#1,NS
  ISN 0023            GJR%J,L<#GJR%J,L<&VRN%II,1<*GJ%II,I<*XR%I<
  ISN 0024        260 GJI%J,L<#GJI%J,L<-VIN%II,1<*GJ%II,I<*XI%I<
  ISN 0025        265 GRJ%J,L<#GJR%J,L<&GJI%J,L<
  ISN 0026            DO 266 J#1,NC
  ISN 0027            DO 266 L#1,NF
  ISN 0028        266 SUM#SUM&GRJ%J,L<*GRJ%J,L<
  ISN 0029            WRITE%3,80<
  ISN 0030            WRITE%3,20< %%GRJ%J,L<,L#1,NF<,J#1,NC<
  ISN 0031            IF%NFF .EQ. 0< GO TO 300
  ISN 0033            DO 275 J#1,NC
  ISN 0034            DO 275 L#1,NFF
  ISN 0035            GHR%J,L<#0.D0
  ISN 0036            GHI%J,L<#0.D0
  ISN 0037            DO 270 I#1,NS
  ISN 0038            GHR%J,L<#GHR%J,L<&VRN%I,1<*B%I,J<
  ISN 0039        270 GHI%J,L<#GHI%J,L<-VIN%I,1<*B%I,J<
  ISN 0040            LA#NS&L
  ISN 0041        275 GRH%J,L<#GHR%J,L<*XR%LA<&GHI%J,L<*XI%LA<
  ISN 0042            DO 285 J#1,NFF
  ISN 0043            DO 285 L#1,NF
```

```
          GGR%J,L<#0.DO
          GSI%J,L<#0.DO
          DO 280 I#1,NS
          GGR%J,L<#GGR%J,L<&C%L,I<*XR%I<
      280 GSI%J,L<#GGI%J,L<-C%L,I<*XI%I<
          JA#NS&J
      285 GRG%J,L<#VRN%JA,1<*GGR%J,L<&VIN%JA,1<*GGI%J,L<
          DO 290 J#1,NFF
          NSJ#NS&J
          DO 290 L#1,NFF
          NSL#NS&L
      290 GRF%J,L<#VRN%NSJ,1<*XR%NSL<-VIN%NSJ,1<*XI%NSL<
          WRITE%3,85<
          WRITE%3,20< %%GRH%J,L<,L#1,NFF<,J#1,NC<
          WRITE%3,90<
          WRITE%3,20< %%GRG%J,L<,L#1,NF<,J#1,NFF<
          WRITE%3,95<
          WRITE%3,20< %%GRF%J,L<,L#1,NFF<,J#1,NFF<
          DO 295 J#1,NC
          DO 295 L#1,NFF
      295 SUM#SUM&GRH%J,L<*GHH%J,L<
          DO 296 J#1,NFF
          DO 296 L#1,NF
      296 SUM#SUM&GRG%J,L<*GRG%J,L<
          DO 297 J#1,NFF
          DO 297 L#1,NFF
      297 SUM#SUM&GRF%J,L<*GRF%J,L<
      300 CONTINUE
          DELKI#-DELR/SUM
          RETURN
          END
```

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                         DATE  71.106/19.43.

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002                SUBROUTINE MAXRT%RR,RI,NS,NFF,M2D<
                   C
                   C      COMPUTES ROOT HAVING MAXIMUN REAL PART
                   C
  ISN 0003                REAL*8 RR%M2D<,RI%M2D<,RMR,RMI
  ISN 0004                IA#NS&NFF
  ISN 0005                RMR#RR%1<
  ISN 0006                RMI#RI%1<
  ISN 0007                NM#1
  ISN 0008                DO 100 I=1,IA
  ISN 0009                IF%RMR-RR%I<< 50,100,100
  ISN 0010             50 RMR#RR%I<
  ISN 0011                RMI#RI%I<
  ISN 0012                NM#I
  ISN 0013            100 CONTINUE
  ISN 0014                RR%NM<#RR%1<
  ISN 0015                RI%NM<#RI%1<
  ISN 0016                RR%1<#RMR
  ISN 0017                RI%1<#RMI
  ISN 0018                RETURN
  ISN 0019                END
```

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                         DATE  71.106/19.42.

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002                SUBROUTINE GRDEL%FMATF,FMATG,FMATH,FMATJ,GF,GG,GH,GJ,GRADF,GRADG,
                         1                GRADH,GRADJ,DELK2,NC,NF,NFF,MD<
                   C
                   C      CALCULATES THE CHANGES IN THE FEEDBACK AND COMPENSATOR MATRICES
                   C      DUE TO DELK2 AND THE CORRESPONDING GRADIENTS .
                   C
  ISN 0003                REAL*8 FMATF%MD,MD<,FMATG%MD,MD<,FMATH%MD,MD<,FMATJ%MD,MD<,
                         1       GF%MD,MD<,GG%MD,MD<,GH%MD,MD<,GJ%MD,MD<,GRADF%MD,MD<,
                         2       GRADG%MD,MD<,GRADH%MD,MD<,GRADJ%MD,MD<,DELK2
  ISN 0004                DO 405 J#1,NC
  ISN 0005                DO 405 L#1,NF
  ISN 0006            405 FMATJ%J,L<#GJ%J,L<&DELK2*GRADJ%J,L<
  ISN 0007                IF%NFF .EQ. 0< GO TO 425
  ISN 0009                DO 410 J#1,NC
  ISN 0010                DO 410 L#1,NFF
  ISN 0011            410 FMATH%J,L<#GH%J,L<&DELK2*GRADH%J,L<
  ISN 0012                DO 415 J#1,NFF
  ISN 0013                DO 415 L#1,NF
  ISN 0014            415 FMATG%J,L<#GG%J,L<&DELK2*GRADG%J,L<
  ISN 0015                DO 420 J#1,NFF
  ISN 0016                DO 420 L#1,NFF
  ISN 0017            420 FMATF%J,L<#GF%J,L<&DELK2*GRADF%J,L<
  ISN 0018            425 CONTINUE
  ISN 0019                RETURN
  ISN 0020                END
```

```
:VEL 18 ( SEPT 69 )                          OS/360   FORTRAN H                                  DATE  71.106/19.43.35

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                          SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
 ISN 0002              SUBROUTINE HSHG%N,A,IA,MD2<
               C
               C       CONVERTS A TO UPPER HESSENBERG FORM
               C
 ISN 0003              DIMENSION A%MD2<
 ISN 0004              DOUBLE PRECISION A,PIV,T,S
 ISN 0005              DOUBLE PRECISION DABS
 ISN 0006              L#N
 ISN 0007              NIA#L*IA
 ISN 0008              LIA#NIA-IA
 ISN 0009           20 IF%L-3< 360,40,40
 ISN 0010           40 LIA#LIA-IA
 ISN 0011              L1#L-1
 ISN 0012              L2#L1-1
 ISN 0013              ISUB#LIA&L
 ISN 0014              IPIV#ISUB-IA
 ISN 0015              PIV#DABS%A%IPIV<<
 ISN 0016              IF%L-3< 90,90,50
 ISN 0017           50 M#IPIV-IA
 ISN 0018              DO 80 I#L,M,IA
 ISN 0019              T#DABS%A%I<<
 ISN 0020              IF%T-PIV< 80,80,60
 ISN 0021           60 IPIV#I
 ISN 0022              PIV#T
 ISN 0023           80 CONTINUE
 ISN 0024           90 IF%PIV< 100,320,100
 ISN 0025          100  IF%PIV-DABS%A%ISUB<<< 180,180,120
 ISN 0026          120 M#IPIV-L
 ISN 0027              DO 140 I#1,L
 ISN 0028              J#M&I
 ISN 0029              T#A%J<
 ISN 0030              K#LIA&I
 ISN 0031              A%J<#A%K<
 ISN 0032          140 A%K<#T
 ISN 0033              M#L2-M/IA
 ISN 0034              DO 160 I#L1,NIA,IA
 ISN 0035              T#A%I<
 ISN 0036              J#I-M
 ISN 0037              A%I<#A%J<
 ISN 0038          160 A%J<#T
 ISN 0039          180 DO 200 I#L,LIA,IA
 ISN 0040          200 A%I<#A%I</A%ISUB<
 ISN 0041              J#-IA
 ISN 0042              DO 240 I#1,L2
 ISN 0043              J#J&IA
 ISN 0044              LJ#L&J
 ISN 0045              DO 220 K#1,L1
 ISN 0046              KJ#K&J
 ISN 0047              KL#K&LIA
 ISN 0048          220 A%KJ<#A%KJ<-A%LJ<*A%KL<
 ISN 0049          240 CONTINUE
 ISN 0050              K#-IA
 ISN 0051              DO 300 I#1,N
 ISN 0052              K#K&IA
 ISN 0053              LK#K&L1
```

```
ISN 0054          S=AZLK<
ISN 0055          LJ=L-IA
ISN 0056          DO 280 J=1,L2
ISN 0057          JK=K+J
ISN 0058          LJ=LJ+IA
ISN 0059      280 S=S+A(LJ)*A(JK)*1.0D0
ISN 0060      300 A(LK)=S
ISN 0061          DO 310 I=L,LIA,IA
ISN 0062      310 A(I)=0.0D0
ISN 0063      320 L=LI
ISN 0064          GO TO 20
ISN 0065      360 RETURN
ISN 0066          END
```

LEVEL 18 (SEPT 69)   OS/360   FORTRAN H   DATE 71.106/19.42.09

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                           SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002          SUBROUTINE STAB(A,B,C,FF,FG,FH,FJ,AH,AV,RR,RI,NS,NC,NF,NFF,MD,
                 $           M2D,M2,IANA,IWRITE)
ISN 0003          REAL*8 A(MD,MD),B(MD,MD),C(MD,MD),FF(MD,MD),FG(MD,MD),FH(MD,MD),
                 1        FJ(MD,MD),AH(M2D,M2D),AV(M2),RR(M2D),RI(M2D)
ISN 0004          DIMENSION IANA(M2D)
ISN 0005       10 FORMAT (//T3,'DIRECT FEEDBACK MATRIX  FJ')
ISN 0006       20 FORMAT (4E18.7)
ISN 0007       30 FORMAT (//T2,'ROOTS',5X,'REAL PART',11X,'IMAG. PART')
ISN 0008       40 FORMAT (2F20.6)
ISN 0009       50 FORMAT (//T3,'COMPENSATOR OUTPUT MATRIX  FH')
ISN 0010       60 FORMAT (//T3,'COMPENSATOR INPUT MATRIX  FG')
ISN 0011       70 FORMAT (//T3,'COMPENSATOR MATRIX  FF')
ISN 0012          IA=NS+NFF
ISN 0013          IF(IWRITE .EQ. 0) GO TO 100
ISN 0015          WRITE (3,10)
ISN 0016          WRITE(3,20) ((FJ(I,J),J=1,NF),I=1,NC)
ISN 0017          IF(NFF .EQ. 0) GO TO 100
ISN 0019          WRITE(3,50)
ISN 0020          WRITE(3,20) ((FH(I,J),J=1,NFF),I=1,NC)
ISN 0021          WRITE(3,60)
ISN 0022          WRITE(3,20) ((FG(I,J),J=1,NF),I=1,NFF)
ISN 0023          WRITE(3,70)
ISN 0024          WRITE(3,20) ((FF(I,J),J=1,NFF),I=1,NFF)
ISN 0025      100 CONTINUE
ISN 0026          CALL AMHI(A,B,C,FF,FG,FH,FJ,AH,NS,NC,NF,NFF,MD,M2D)
ISN 0027          CALL MVECT(AH,AV,IA,M2D,M2)
ISN 0028          CALL HSHG(IA,AV,IA,M2)
ISN 0029          CALL ATEIG(IA,AV,RR,RI,IANA,IA,M2D,M2)
ISN 0030          IF(IWRITE .NE. 2) GO TO 110
ISN 0032          WRITE(3,30)
ISN 0033          WRITE(3,40) (RR(I),RI(I),I=1,IA)
ISN 0034      110 RETURN
ISN 0035          END
```

```
LEVEL 18 ( SEPT 69 )                          OS/360  FORTRAN H                                    DATE  71.106/19.44.

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=CCCOK,
                             SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
 ISN 0002           SUBROUTINE APPROXZAMAT,BMAT,CMAT,AHAT,AAAA,RR,RI,FMATF,FMATG,FMATH
                  1                  ,FMATJ,GF,GG,GH,GJ,GFM,GGM,GHM,GJM,GRADF,GRADG,
                  2                  GRADH,GRADJ,NS,NC,NF,NFF,MD,M2D,MD2,M2,IANA,
                  3                  STOPR,DELK1,DELR,DELR1,ITEST,RCOT1,ROOTR,ROOTI,
                  4                  NBEST<
               C
               C    ROUTINE DETERMINES MAXIMUM ZOR REQUESTED < STABILITY ALONG A GIVEN
               C        GRADIENT.QUADRATIC CURVE-FITTING IS USED TO DETERMINE THE
               C        MAXIMUM.
               C
 ISN 0003           REAL*8 AMATZMD,MD<,BMATZMD,MD<,CMATZMD,MD<,AHATZM2D,M2D<,
                  1        AAAAZM2<,RRZM2D<,RIZM2D<,FMATFZMD,MD<,FMATGZMD,MD<,FMATHZMD
                  2        ,MD<,FMATJZMD,MD<,GFZMD,MD<,GGZMD,MD<,GHZMD,MD<,GJZMD,MD<,
                  3        GFMZMD,MD<,GGMZMD,MD<,GHMZMD,MD<,GJMZMD,MD<,GRADFZMD,MD<,
                  4        GRADGZMD,MD<,GRADHZMD,MD<,GRADJZMD,MD<
 ISN 0004           DIMENSION IANAZM2D<
 ISN 0005           REAL*8 DELK1,DELK2,DELK3,DELK4,ROOTR,ROOTI,ROOT1,ROOT2,ROOT3,
                  1        RCOT4,ROOT5
 ISN 0006           REAL*8 DELR,DELR1,STOPR
               C
 ISN 0007           CALL GRDELZFMATF,FMATG,FMATH,FMATJ,GF,GG,GH,GJ,GRADF,GRADG,
                  1             GRADH,GRADJ,DELK1,NC,NF,NFF,MD<
 ISN 0008           IWRITE=0
 ISN 0009           CALL STABZAMAT,BMAT,CMAT,FMATF,FMATG,FMATH,FMATJ,AHAT,AAAA,RR,RI,
                  A            NS,NC,NF,NFF,MD,M2D,M2,IANA,IWRITE<
 ISN 0010           CALL MAXRTZRR,RI,NS,NFF,M2D<
 ISN 0011           IFZRATI<-STOPR< 999,330,330
 ISN 0012       330 RCOT2=RRI<
 ISN 0013           KOUNT1=1
 ISN 0014           KOUNT2=1
 ISN 0015           IFZRCOT2-RCOT1< 400,400,590
 ISN 0016       400 KOUNT1=KOUNT1&1
 ISN 0017           IFZRCOT2-RCOTR< 791,792,792
 ISN 0018       791 ROOTR=RRI<
 ISN 0019           RCOTI=RAI1<
 ISN 0020           CALL GMRZFMATF,FMATG,FMATH,FMATJ,GFM,GGM,GHM,GJM,NC,NF,NFF,MD<
 ISN 0021           NBEST=NBEST&1
 ISN 0022       792 CONTINUE
 ISN 0023           IFZKCUNT1-10< 401,401,899
 ISN 0024       401 DELK2=2.0*DELK1
 ISN 0025           CALL GRDELZFMATF,FMATG,FMATH,FMATJ,GF,GG,GH,GJ,GRADF,GRADG,
                  1             GRADH,GRADJ,DELK2,NC,NF,NFF,MD<
 ISN 0026           CALL STABZAMAT,BMAT,CMAT,FMATF,FMATG,FMATH,FMATJ,AHAT,AAAA,RR,RI,
                  A            NS,NC,NF,NFF,MD,M2D,M2,IANA,IWRITE<
 ISN 0027           CALL MAXRTZRR,RI,NS,NFF,M2D<
 ISN 0028           IFZRATI<-STOPR< 999,430,430
 ISN 0029       430 RCOT3=RRI<
 ISN 0030           IFZRCOT3-RCOTR< 793,794,794
 ISN 0031       793 ROOTR=RRI<
 ISN 0032           RCOTI=RAI1<
 ISN 0033           CALL GMRZFMATF,FMATG,FMATH,FMATJ,GFM,GGM,GHM,GJM,NC,NF,NFF,MD<
 ISN 0034           NBEST=NBEST&1
 ISN 0035       794 CONTINUE
 ISN 0036           IFZRCOT3-RCOT2< 435,440,640
 ISN 0037       435 IFZZRCOT1-RCOT2</ZRCOT2-RCOT3<-1.1CC< 440,500,500
```

```
ISN 0038        440 ROOT2#ROOT3
ISN 0039            DELK1#DELK2
ISN 0040            GO TO 400
ISN 0041        500 DELK3#DELK1*%ROOT3-4.D0*ROOT2&3.D0*ROOT1</%2.D0*ROOT3-4.D0*ROOT2
                   1&2.D0*ROOT1<
ISN 0042            CALL GRDEL%FMATF,FMATG,FMATH,FMATJ,GF,GG,GH,GJ,GRADF,GRADG,
                   1              GRADH,GRADJ,DELK3,NC,NF,NFF,MD<
ISN 0043            CALL STAB%AMAT,BMAT,CMAT,FMATF,FMATG,FMATH,FMATJ,AHAT,AAAA,RR,RI,
                   A              NS,NC,NF,NFF,MD,M2D,M2,IANA,IWRITE<
ISN 0044            CALL MAXRT%RR,RI,NS,NFF,M2D<
ISN 0045            IF%RR%1<-STOPR< 999,530,530
ISN 0046        530 ROOT4#RR%1<
ISN 0047            IF%ROOT4-ROOTR< 795,796,796
ISN 0048        795 ROOTR#RR%1<
ISN 0049            ROOTI#RI%1<
ISN 0050            CALL GMR%FMATF,FMATG,FMATH,FMATJ,GFM,GGM,GHM,GJM,NC,NF,NFF,MD<
ISN 0051            NBEST#NBEST&1
ISN 0052        796 CONTINUE
ISN 0053            IF%ROOT3-ROOT4< 545,535,535
ISN 0054        535 IF%ROOT1-ROOT4-DELR1< 999,540,540
ISN 0055        540 DELR#.5D0*%ROOT1-ROOT4<
ISN 0056            GO TO 899
ISN 0057        545 DELK4#DELK1&%%ROOT3-ROOT2<*%DELK3-DELK1<*%DELK3-DELK1<-%ROOT4-
                   1  ROOT2<*DELK1*DELK1</%2.D0*%%ROOT3-ROOT2<*%DELK3-DELK1<-%ROOT4-
                   2  ROOT2<*DELK1<<
ISN 0058            CALL GRDEL%FMATF,FMATG,FMATH,FMATJ,GF,GG,GH,GJ,GRADF,GRADG,
                   1              GRADH,GRADJ,DELK4,NC,NF,NFF,MD<
ISN 0059            CALL STAB%AMAT,BMAT,CMAT,FMATF,FMATG,FMATH,FMATJ,AHAT,AAAA,RR,RI,
                   A              NS,NC,NF,NFF,MD,M2D,M2,IANA,IWRITE<
ISN 0060            CALL MAXRT%RR,RI,NS,NFF,M2D<
ISN 0061            IF%RR%1<-STOPR< 999,575,575
ISN 0062        575 ROOT5#RR%1<
ISN 0063            IF%ROOT5-ROOTR< 797,798,798
ISN 0064        797 ROOTR#RR%1<
ISN 0065            ROOTI#RI%1<
ISN 0066            CALL GMR%FMATF,FMATG,FMATH,FMATJ,GFM,GGM,GHM,GJM,NC,NF,NFF,MD<
ISN 0067            NBEST#NBEST&1
ISN 0068        798 CONTINUE
ISN 0069            IF%ROOT1-ROOT5-DELR1< 999,580,580
ISN 0070        580 DELR#.5D0*%ROOT1-ROOT5<
ISN 0071            GO TO 899
ISN 0072        590 CONTINUE
ISN 0073            KOUNT2#KOUNT2&1
ISN 0074            IF%KOUNT2-10< 595,595,899
ISN 0075        595 DELK2#DELK1
ISN 0076            ROOT3#ROOT2
ISN 0077            DELK1#.5D0*DELK1
ISN 0078            CALL GRDEL%FMATF,FMATG,FMATH,FMATJ,GF,GG,GH,GJ,GRADF,GRADG,
                   1              GRADH,GRADJ,DELK1,NC,NF,NFF,MD<
ISN 0079            CALL STAB%AMAT,BMAT,CMAT,FMATF,FMATG,FMATH,FMATJ,AHAT,AAAA,RR,RI,
                   A              NS,NC,NF,NFF,MD,M2D,M2,IANA,IWRITE<
ISN 0080            CALL MAXRT%RR,RI,NS,NFF,M2D<
ISN 0081            IF%RR%1<-STOPR< 999,630,630
ISN 0082        630 ROOT2#RR%1<
ISN 0083            IF%ROOT2-ROOTR< 799,800,800
ISN 0084        799 ROOTR#RR%1<
ISN 0085            ROOTI#RI%1<
ISN 0086            CALL GMR%FMATF,FMATG,FMATH,FMATJ,GFM,GGM,GHM,GJM,NC,NF,NFF,MD<
```

```
ISN 0087          NBEST#NBEST&1
ISN 0088      800 CONTINUE
ISN 0089          IF%RCCT2-RCCT1< 640,640,590
ISN 0090      640 DELK3#DELK1*%ROOT3-4.D0*RCOT2&3.D0*RCOT1</%2.D0*RCOT3-4.D0*ROOT2
                 1&2.D0*ROOT1<
ISN 0091          CALL GRDEL%FMATF,FMATG,FMATH,FMATJ,GF,GG,GH,GJ,GRADF,GRADG,
                 1              GRADH,GRADJ,DELK3,NC,NF,NFF,MD<
ISN 0092          CALL START%AMAT,BMAT,CMAT,FMATF,FMATG,FMATH,FMATJ,AHAT,AAAA,RR,RI,
                 A              NS,NC,NF,NFF,MD,M2D,M2,IANA,IWRITE<
ISN 0093          CALL MAXRT%RR,RI,NS,NFF,M2D<
ISN 0094          IF%RR%1<-STOPR< 999,670,670
ISN 0095      670 ROOT4#RR%1<
ISN 0096          IF%ROOT4-ROOTR< 801,802,802
ISN 0097      801 ROOTR#RR%1<
ISN 0098          ROOTI#RI%1<
ISN 0099          CALL GMR%FMATF,FMATG,FMATH,FMATJ,GFM,GGM,GHM,GJM,NC,NF,NFF,MD<
ISN 0100          NBEST#NBEST&1
ISN 0101      802 CONTINUE
ISN 0102          IF%ROOT4-ROOT2< 675,675,685
ISN 0103      675 IF%ROOT1-ROOT4-DELR1< 899,680,680
ISN 0104      680 DELR#.5D0*%ROOT1-ROOT4<
ISN 0105          GO TO 899
ISN 0106      685 IF%ROOT1-ROOT2-DELR1< 899,690,690
ISN 0107      690 CONTINUE
ISN 0108          CALL GRDEL%FMATF,FMATG,FMATH,FMATJ,GF,GG,GH,GJ,GRADF,GRADG,
                 1              GRADH,GRADJ,DELK1,NC,NF,NFF,MD<
ISN 0109          DELR#.5D0*%ROOT1-ROOT2<
ISN 0110          CALL START%AMAT,BMAT,CMAT,FMATF,FMATG,FMATH,FMATJ,AHAT,AAAA,RR,RI,
                 A              NS,NC,NF,NFF,MD,M2D,M2,IANA,IWRITE<
ISN 0111          CALL MAXRT%RR,RI,NS,NFF,M2D<
ISN 0112          IF%RR%1<-STOPR< 999,725,725
ISN 0113      725 ROOT1#RR%1<
ISN 0114          IF%ROOT1-ROOTR< 803,899,899
ISN 0115      803 ROOTR#RR%1<
ISN 0116          ROOTI#RI%1<
ISN 0117          CALL GMR%FMATF,FMATG,FMATH,FMATJ,GFM,GGM,GHM,GJM,NC,NF,NFF,MD<
ISN 0118          NBEST#NBEST&1
ISN 0119      899 ITEST#1
ISN 0120          GO TO 1000
ISN 0121      999 ITEST#2
ISN 0122     1000 RETURN
ISN 0123          END
```

```
ISN 0127        400 AUX3(I,J)=AUX3(I,J)+V(K,I)*AUX1(K,J)
              C
              C     COMPUTATION OF THE GAIN MATRIX  K(I+1)  AND COMPARISON WITH  K(I)
              C
ISN 0128            DO 410 I=1,NC
ISN 0129            DO 410 J=1,NS
ISN 0130            AUX1(I,J)=0.D0
ISN 0131            DO 410 K=1,NS
ISN 0132        410 AUX1(I,J)=AUX1(I,J)+B(K,I)*AUX3(K,J)
ISN 0133            DO 420 I=1,NC
ISN 0134            DO 420 J=1,NS
ISN 0135            P1(I,J)=0.D0
ISN 0136            DO 420 K=1,NC
ISN 0137        420 P1(I,J)=P1(I,J)+P1(I,K)*AUX1(K,J)
ISN 0138            DO 430 J=1,NS
ISN 0139            DO 430 I=1,NC
ISN 0140        430 AUX1(I,J)=P1(I,J)-P(I,J)
ISN 0141            DEL=0.D0
ISN 0142            DO 440 J=1,NS
ISN 0143            DO 440 I=1,NC
ISN 0144            IF(DABS(P1(I,J)) .GT. TOL) SV(1)=DABS(AUX1(I,J)/P1(I,J))
ISN 0146            IF(SV(1) .GT. DEL) DEL=SV(1)
ISN 0148        440 CONTINUE
ISN 0149            IF(DEL-TOL) 460,460,450
ISN 0150        450 IF(ITER-MAXIT) 462,470,470
ISN 0151        462 DO 464 J=1,NS
ISN 0152            DO 464 I=1,NC
ISN 0153        464 P(I,J)=P1(I,J)
ISN 0154            GO TO 100
ISN 0155        460 WRITE(3,25) DEL,ITER
ISN 0156            RETURN
ISN 0157        470 WRITE(3,20) TOL,MAXIT
ISN 0158            RETURN
ISN 0159        480 WRITE(3,10)
ISN 0160            RETURN
ISN 0161        490 WRITE(3,15) IER
ISN 0162            RETURN
ISN 0163        500 WRITE(3,30)
ISN 0164            RETURN
ISN 0165            END
```

```
LEVEL 18 ( SEPT 69 )                              OS/360  FORTRAN H                                   DATE  71.096/22.24.

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002                   SUBROUTINE ATFIG(M,A,RR,RI,IANA,IA,MD,MD2)
                 C
                 C           COMPUTES ROOTS OF UPPER HESSENBERG MATRIX A
                 C
  ISN 0003                   DIMENSION A(MD2),RR(MD),RI(MD),PRR(2),PRI(2),IANA(MD)
  ISN 0004                   DOUBLE PRECISION E7,E6,E10,DELTA,PRR,PRI,PAN,PAN1,R,S,T,A,U,V,RR,
                        1                     RI,RMOD,EPS,D,G1,G2,G3,CAP,PSI1,PSI2,ALPHA,ETA
  ISN 0005                   DOUBLE PRECISION DABS,DSQRT,DMAX1
  ISN 0006                   INTEGER P,P1,Q
  ISN 0007                   E7=1.0D-8
  ISN 0008                   E6=1.0D-6
  ISN 0009                   E10=1.0D-10
  ISN 0010                   DELTA=0.5D0
  ISN 0011                   MAXIT=30
  ISN 0012                   N=M
  ISN 0013                20 N1=N-1
  ISN 0014                   IN=N1*IA
  ISN 0015                   NN=IN&N
  ISN 0016                   IF(N1) 30,1300,30
  ISN 0017                30 NP=N&1
  ISN 0018                   IT=0
  ISN 0019                   DO 40 I=1,2
  ISN 0020                   PRR(I)=0.0D0
  ISN 0021              40   PRI(I)=0.0D0
  ISN 0022                   PAN=0.0D0
  ISN 0023                   PAN1=0.0D0
  ISN 0024                   R=0.0D0
  ISN 0025                   S=0.0D0
  ISN 0026                   N2=N1-1
  ISN 0027                   IN1=IN-IA
  ISN 0028                   NN1=IN1&N
  ISN 0029                   N1N=IN&N1
  ISN 0030                   N1N1=IN1&N1
  ISN 0031                60 T=A(N1N1)-A(NN)
  ISN 0032                   U=T*T
  ISN 0033                   V=4.0D0*A(N1N)*A(NN1)
  ISN 0034                   IF(DABS(V)-U*E7) 100,100,65
  ISN 0035                65 T=U&V
  ISN 0036                   IF(DABS(T)-DMAX1(U,DABS(V))*E6) 67,67,68
  ISN 0037              67   T=0.0D0
  ISN 0038              68   U=(A(N1N1)&A(NN))/2.0D0
  ISN 0039                   V=DSQRT(DABS(T))/2.0D0
  ISN 0040                   IF(T)140,70,70
  ISN 0041                70 IF(U) 80,75,75
  ISN 0042                75 RR(N1)=U&V
  ISN 0043                   RR(N)=U-V
  ISN 0044                   GO TO 130
  ISN 0045                80 RR(N1)=U-V
  ISN 0046                   RR(N)=U&V
  ISN 0047                   GO TO 130
  ISN 0048               100 IF(T)120,110,110
  ISN 0049               110 RR(N1)=A(N1N1)
  ISN 0050                   RR(N)=A(NN)
  ISN 0051                   GO TO 130
  ISN 0052               120 RR(N1)=A(NN)
```

```
      RR(N)=A(NIN1)
130   RI(N)=C.CDO
      RI(N1)=0.CDO
      RI(N1)=0.0
      GO TC 160
 140  RR(N1)=U
      RR(N)=U
      RI(N1)=V
      RI(N)=-V
 160  IF(N2)1280,1280,180
 180  N1N2=N1N1-IA
      RMOC=RK(N1)*RR(N1)&RI(N1)*RI(N1)
      EPS=E10*CSQRT(RMOD)
      IF(DABS(A(N1N2))-EPS) 1280,1280,240
240   IF(DABS(A(NN1))-E10*DABS(A(NN))) 1300,1300,250
250   IF(DABS(PAN1-A(N1N2))-DABS(A(N1N2))*E6) 1240,1240,260
260   IF(DABS(PAN-A(NN1))-DABS(A(NN1))*E6) 1240,1240,300
 300  IF(IT-MAXIT) 320,1240,124C
 320  J=1
      DO 36C I=1,2
      K=NP-I
      IF(DABS(RR(K)-PRR(I))&DABS(RI(K)-PRI(I))-DELTA*(DABS(RR(K))
     1     &DABS(RI(K)))) 340,360,360
 340  J=J&I
 360  CONT INUE
      GO TC (440,460,46C,48C),J
440   R=0.CDC
      S=0.CDO
      GO TC 50C
 460  J=N&2-J
      R=RR(J)*RR(J)
      S=RR(J)&RR(J)
      GO TC 500
 480  R=RR(N)*RR(N1)-RI(N)*RI(N1)
      S=RR(N)&RR(N1)
 500  PAN=A(NN1)
      PAN1=A(N1N2)
      CO 52C I=1,2
      K=NP-I
      PRR(I)=RR(K)
 520  PRI(I)=RI(K)
      P=N2
      IF(N-3)6C0,6C0,525
 525  IPI=N1N2
      CO 5€0 J=2,N2
      IPI=IPI-IA-1
      IF(CABS(A(IPI))-EPS) 600,600,530
 530  IPIP=IPI&IA
      IPIP2=IPIP&IA
      C=A(IPIP)*(A(IPIP)-S)&A(IPIP2)*A(IPIP&1)&R
      IF(C)540,56C,540
540   IF(DABS(A(IPI)*A(IPIP&1))*(DABS(A(IPIP)&A(IPIP2&1)-S)&DABS(A(IPIP2
     1     &2)))-DABS(D)*EPS) 620,620,560
 560  P=N1-J
 580  CONTINUE
 600  Q=P
      GU TC 680
 620  P1=P-1
```

```
          Q=P1
          IF(P1-1)680,680,650
      650 DO 660 I=2,P1
          IPI=IPI-IA-1
          IF(DABS(A(IPI))-EPS) 680,680,660
      660 Q=Q-1
      680 II=(P-1)*IA&P
          DO 1220 I=P,N1
          III=II-IA
          IIP=II&IA
          IF(I-P)720,700,720
      700 IPI=II&1
          IPIP=IIP&1
          G1=A(II)*(A(III)-S)&A(IIP)*A(IPI)&R
          G2=A(IPI)*(A(IPIP)&A(III)-S)
          G3=A(IPI)*A(IPIP&1)
          A(IPI&1) =0.0D0
          GO TO 780
      720 G1=A(III)
          G2=A(III&1)
          IF(I-N2)740,740,760
      740 G3=A(III&2)
          GO TO 780
     760  G3=0.0D0
     780  CAP=DSQRT(G1*G1&G2*G2&G3*G3)
          IF(CAP)800,860,800
      800 IF(G1)820,840,840
      820 CAP=-CAP
      840 T=G1&CAP
          PSI1=G2/T
          PSI2=G3/T
          ALPHA=2.0D0/(1.0D0&PSI1*PSI1&PSI2*PSI2)
          GO TO 880
     860  ALPHA=2.0D0
          PSI1=0.0D0
          PSI2=0.0D0
      880 IF(I-Q)900,960,900
      900 IF(I-P)920,940,920
      920 A(III)=-CAP
          GO TO 960
      940 A(III)=-A(III)
      960 IJ=I
          DO 1040 J=I,N
          T=PSI1*A(IJ&1)
          IF(I-N1)980,1000,1000
      980 IP2J=IJ&2
          T=T&PSI2*A(IP2J)
     1000 ETA=ALPHA*(T&A(IJ))
          A(IJ)=A(IJ)-ETA
          A(IJ&1)=A(IJ&1)-PSI1*ETA
          IF(I-N1)1020,1040,1040
     1020 A(IP2J)=A(IP2J)-PSI2*ETA
     1040 IJ=IJ&IA
          IF(I-N1)1080,1060,1060
     1060 K=N
          GO TO 1100
     1080 K=I&2
     1100 IP=IIP-1
```

```
ISN 0167          DO 1160 J=Q,K
ISN 0168          JIP=IP&J
ISN 0169          JI=JIP-IA
ISN 0170          T=PSI1*A(JIP)
ISN 0171          IF(I-N1)1120,1140,1140
ISN 0172     1120 JIP2=JIP&IA
ISN 0173          T=T&PSI2*A(JIP2)
ISN 0174     1140 ETA=ALPHA*(T&A(JI))
ISN 0175          A(JI)=A(JI)-ETA
ISN 0176          A(JIP)=A(JIP)-ETA*PSI1
ISN 0177          IF(I-N1)1160,1180,1180
ISN 0178     1160 A(JIP2)=A(JIP2)-ETA*PSI2
ISN 0179     1180 CONTINUE
ISN 0180          IF(I-N2)1200,1220,1220
ISN 0181     1200 JI=II&3
ISN 0182          JIP=JI&IA
ISN 0183          JIP2=JIP&IA
ISN 0184          ETA=ALPHA*PSI2*A(JIP2)
ISN 0185          A(JI)=-ETA
ISN 0186          A(JIP)=-ETA*PSI1
ISN 0187          A(JIP2)=A(JIP2)-ETA*PSI2
ISN 0188     1220 II=IIP&1
ISN 0189          IT=IT&1
ISN 0190          GO TC 60
ISN 0191     1240 IF(DABS(A(NN1))-DABS(A(N1N2))) 1300,1280,1280
ISN 0192     1280 IANA(N)=0
ISN 0193          IANA(N1)=2
ISN 0194          N=N2
ISN 0195          IF(N2)1400,1400,20
ISN 0196     1300 RR(N)=A(NN)
ISN 0197          RI(N)=0.0D0
ISN 0198          IANA(N)=1
ISN 0199          IF(N1)1400,1400,1320
ISN 0200     1320 N=N1
ISN 0201          GO TC 20
ISN 0202     1400 RETURN
ISN 0203          END
```

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                                      DATE  71.106/19.45.21

           COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                              SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002                   SUBROUTINE EIGVEC%IVC,  A, B, W, IROW, XR, XI, VR, VI, ROOTRE,     ESY1   0
                        1      ROOTIE, NE, NMAX, T2, SW1, COUNTE, ERR,MMM<                      ESY1  10
                   C           SUBROUTINE TO FIND THE EIGENVECTORS OF A NON-SYMMETRIC MATRIX    ESY1  20
                   C           BY A MODIFIED WILKINSON@S INVERSE ITERATION METHOD.              ESY1  30
                   C           CONTROL IVC CODE IS                                              ESY1  40
                   C                1  FIND ONLY THE REGULAR EIGENVECTORS   %A X # LAMBDA X<   ESY1  50
                   C                2  FIND ONLY THE TRANSPOSED EIGENVECTORS %AT V # LAMBDA V<ESY1  60
                   C                3  FIND BOTH TYPES OF EIGENVECTORS.                        ESY1  70
  ISN 0003                   DIMENSION A%NMAX,NMAX<,B%NMAX,NMAX<,W%NMAX,4<,XR%NMAX<,XI%NMAX<,
                        1                VR%NMAX<,VI%NMAX<,IROW%NMAX,2<
  ISN 0004                   DOUBLE PRECISION ROOTR,ROOTI,ROOTRE,ROOTIE,TEMP,TEMP2,AMAX,C1,C2,
                        1                         SW1,W,XR,XI,VR,VI,B,ZERO,DCERR,A
  ISN 0005                   DOUBLE PRECISION DABS,DSIGN,DSQRT,DMAX1
  ISN 0006                   INTEGER COUNT, COUNTE, T2                                           ESY1 100
  ISN 0007                   IO1#1
  ISN 0008                   IO3#3
  ISN 0009                   ROOTR # ROOTRE                                                      ESY1 110
  ISN 0010                   ROOTI # ROOTIE                                                      ESY1 120
  ISN 0011                   N # NE                                                              ESY1 130
  ISN 0012                   MM # MMM - 1                                                        ESY1 140
  ISN 0013                   N1 # N - 1                                                          ESY1 150
  ISN 0014                   NP1 # N & 1                                                         ESY1 160
  ISN 0015                   IVC1 # IVC - 1                                                      ESY1 170
  ISN 0016                   IVC2 # IVC1 - 1                                                     ESY1 180
  ISN 0017                   COUNT # 1                                                           ESY1 190
  ISN 0018                   DO 400 I#1,N
  ISN 0019                   W%I,1<#0.0D0
  ISN 0020                   XR%I<#0.0D0
  ISN 0021              400  CONTINUE
  ISN 0022                   CLIM # 1.0E-4                                                       ESY1 200
  ISN 0023                   IF%ROOTI< 1, 60, 1                                                  ESY1 210
                   C                                                                             ESY1 220
                   C            COMPLEX EIGENVALUE.                                              ESY1 230
                   C                                                                             ESY1 240
  ISN 0024                 1 TEMP # - ROOTR - ROOTR                                              ESY1 250
  ISN 0025                   ISW # 2                                                             ESY1 260
  ISN 0026                   TEMP2#ROOTR*ROOTR&ROOTI*ROOTI
  ISN 0027                   JJ # 300                                                            ESY1 280
  ISN 0028                   DO 606 I # 1, N                                                     ESY1 290
  ISN 0029                   IF%T2< 600, 603, 600                                                ESY1 300
  ISN 0030               600 DO 602 J # 1, N                                                     ESY1 310
  ISN 0031                   JJ # JJ & 1                                                         ESY1 320
  ISN 0032                   IF%JJ - 251< 602, 601, 601                                          ESY1 330
  ISN 0033               601 JJ # 1                                                              ESY1 340
  ISN 0034                   READ %T2< %W%LL,1<, LL # 1,250<                                     ESY1 350
  ISN 0035               602 B%I,J< # A%I,J<*TEMP & W%JJ,1<                                      ESY1 360
  ISN 0036                   GO TO 605                                                           ESY1 370
  ISN 0037               603 DO 604 J # 1, N                                                     ESY1 380
  ISN 0038               604 B%I,J< # A%I,J<*TEMP & B%I,J<                                       ESY1 390
  ISN 0039               605 B%I,I< # B%I,I< & TEMP2                                             ESY1 400
  ISN 0040               606 A%I,I< # A%I,I< - ROOTR                                             ESY1 410
  ISN 0041                   IF%T2 .NE. 0< REWIND T2                                             ESY1 420
  ISN 0043                   GO TO 700                                                           ESY1 430
  ISN 0044               607 IF%ISW< 622, 608, 622                                               ESY1 440
                   C                                                                             ESY1 450
```

```
C        MATRIX SINGULAR.
C
  622 IF(IVC2< 623, 625, 623
  623 DO 624 LL # 1, N
      W(LL,2<#0.0D0
 624  XI(LL<#0.0D0
      IF(IVC1< 625, 514, 625
  625 DO 626 LL # 1, N
      W(LL,4<#0.0D0
 626  VI(LL<#0.0D0
      GO TO 511
C
C        MATRIX NOT SINGULAR.
C
  608 DO 609 LL # 1, N
      W(LL,1<#1.0D0
      W(LL,2<#1.0D0
      W(LL,3<#1.0D0
 609  W(LL,4<#1.0D0
  699 IF(IVC2< 610, 612, 610
  610 DO 611 I # 1, N
      I2 # IROW(I,2<
      XI(I2< # W(I,1<*RCOTI
      DO 611 J # 1, N
  611 XI(I2< # XI(I2< & A(I,J<*W(J,2<
      IF(IVC1< 612, 500, 612
  612 DO 613 I # 1, N
      VI(I< # W(I,3<*ROOTI
      DO 613 J # 1,N
  613 VI(I< # VI(I< & A(J,I<*W(J,4<
      GO TO 499
  615 CERR # 0.0
      DCERR#0.0D0
      IF(IVC2< 616, 619, 616
  616 DO 618 I # 1, N
      XR(I< # -W(I,2<
      DO 617 J # 1, N
  617 XR(I< # XR(I< & A(I,J<*XI(J<
  618 XR(I< # XR(I</ROOTI
      IF(IVC1< 619, 633, 619
  619 DO 621 I # 1, N
      VR(I< # -W(I,4<
      DO 620 J # 1, N
  620 VR(I< # VR(I< & A(J,I<*VI(J<
  621 VR(I< # VR(I</ROOTI
C
C        SEARCH VECTORS FOR LARGEST ELEMENT AND NORMALIZE.
C
 627  AMAX#0.0D0
      DO 629 L # 1, N
      TEMP # VR(L<**2 & VI(L<**2
      IF(TEMP - AMAX< 629, 629, 628
  628 AMAX # TEMP
      I2 # L
  629 CONTINUE
      C1 # VR(I2</AMAX
      C2 # -VI(I2</AMAX
      DO 630 L # 1, N
```

```
ISN 0095          TEMP # VI%L<                                                  ESVI 980
ISN 0096          VI%L< # VR%L<*C2 & TEMP*C1                                    ESVI 990
ISN 0097      630 VR%L< # VR%L<*C1 - TEMP*C2                                    ESV11000
ISN 0098          IF%COUNT .EQ. 1< GO TO 632                                    ESV11010
ISN 0100          DO 631 LL # 1, N                                              ESV11020
ISN 0101     631  DCERR#DMAX1%DCERR,DABS%VR%LL<-W%LL,3<<,DABS%VI%LL<-W%LL,4<<<
ISN 0102      632 IF%INC2< 633, 638, 633                                        ESV11040
ISN 0103     633  AMAX#0.0D0
ISN 0104          DO 635 L # 1, N                                               ESV11060
ISN 0105          TEMP # XR%L<**2 & XI%L<**2                                    ESV11070
ISN 0106          IF%TEMP - AMAX< 635, 635, 634                                 ESV11080
ISN 0107      634 AMAX # TEMP                                                   ESV11090
ISN 0108          I2 # L                                                        ESV11100
ISN 0109      635 CONTINUE                                                      ESV11110
ISN 0110          C1 # XR%I2</AMAX                                              ESV11120
ISN 0111          C2 # -XI%I2</AMAX                                             ESV11130
ISN 0112          DO 636 L # 1, N                                               ESV11140
ISN 0113          TEMP # XI%L<                                                  ESV11150
ISN 0114          XI%L< # XR%L<*C2 & TEMP*C1                                    ESV11160
ISN 0115      636 XR%L< # XR%L<*C1 - TEMP*C2                                    ESV11170
ISN 0116          IF%COUNT .EQ. 1< GO TO 646                                    ESV11180
ISN 0118          DO 637 LL # 1, N                                              ESV11190
ISN 0119     637  DCERR#DMAX1%DCERR,DABS%XR%LL<-W%LL,1<<,DABS%XI%LL<-W%LL,2<<<
                C                                                               ESV11210
                C        TEST FOR CONVERGENCE.                                  ESV11220
                C                                                               ESV11230
ISN 0120      638 IF%COUNT .EQ. 1< GO TO 646                                    ESV11240
ISN 0122          CERR#DCERR
ISN 0123          IF%CERR .GE. 1.0E-4< GO TO 639                                ESV11250
ISN 0125          IF%CERR .GE. CLIM< GO TO 648                                  ESV11260
ISN 0127          CLIM # CERR                                                   ESV11270
ISN 0128          IF%CLIM .LE. 1.0E-8< GO TO 648                                ESV11280
ISN 0130      639 IF%COUNT .GE. 15< GO TO 68                                    ESV11290
ISN 0132      647 COUNT # COUNT & 1                                             ESV11300
ISN 0133          IF%RCCI< 642, 673, 642                                        ESV11310
ISN 0134      642 IF%INC2< 640, 644, 640                                        ESV11320
ISN 0135      640 DO 641 LL # 1, N                                              ESV11330
ISN 0136          W%LL,1< # XR%LL<                                              ESV11340
ISN 0137      641 W%LL,2< # XI%LL<                                              ESV11350
ISN 0138          IF%INC1< 644, 610, 644                                        ESV11360
ISN 0139      644 DO 645 LL # 1, N                                              ESV11370
ISN 0140          W%LL,3< # VR%LL<                                              ESV11380
ISN 0141      645 W%LL,4< # VI%LL<                                              ESV11390
ISN 0142          GO TO 699                                                     ESV11400
ISN 0143      646 CERR # 0.0                                                    ESV11410
ISN 0144          DCERR#0.0D0
ISN 0145          IF%ICC< 648, 647, 648                                         FSV11420
ISN 0146      648 ERR # CERR                                                    ESV11430
ISN 0147          COUNTF # COUNT                                                ESV11440
ISN 0148          IF%RCCI< 667, 668, 667                                        ESV11450
ISN 0149      667 DO 649 I # 1, N                                               ESV11460
ISN 0150      649 A%I,I< # A%I,I< & RCOTR                                       ESV11470
ISN 0151          RETURN                                                        ESV11480
ISN 0152       68 PRINT 101, RCOTR, RCOTI, CERR                                 ESV11490
ISN 0153          GO TO 648                                                     ESV11500
                C                                                               ESV11510
                C        REAL EIGENVECTORS.                                     ESV11520
                C                                                               ESV11530
```

```
ISN 0154          60 ISN # 1                                   ESV11540
ISN 0155             DO 651 I # 1, N                           ESV11550
ISN 0156             DO 650 J # 1, N                           ESV11560
ISN 0157         650 B%I,J< # A%I,J<                           ESV11570
ISN 0158         651 B%I,I< # B%I,I< - RCOTR                   ESV11580
ISN 0159             GO TC 700                                 ESV11590
ISN 0160         652 IF%ICC< 680, 685, 680                     ESV11600
            C                                                  ESV11610
            C           SINGULAR MATRIX.                       ESV11620
            C                                                  ESV11630
ISN 0161         680 IF%IVC2< 681, 683, 681                    ESV11640
ISN 0162         681 DO 682 L # 1, N                           ESV11650
ISN 0163        682  XI%L<#0.0D0
ISN 0164             IF%IVC1< 683, 514, 683                    ESV11670
ISN 0165         683 DO 684 L # 1, N                           ESV11680
ISN 0166        684  VI%L<#0.0D0
ISN 0167             GO TC 511                                 ESV11700
            C                                                  ESV11710
            C           MATRIX NOT SINGULAR.                   ESV11720
            C                                                  ESV11730
ISN 0168         685 IF%IVC2< 653, 656, 653                    ESV11740
ISN 0169         653 DO 654 L # 1, N                           ESV11750
ISN 0170        654  XI%L<#1.0D0
ISN 0171             IF%IVC1< 656, 500, 656                    ESV11770
ISN 0172         656 DO 657 L # 1, N                           ESV  78
ISN 0173        657  VI%L<#1.0D0
ISN 0174             GO TC 499                                 ESV118C0
            C                                                  ESV11810
            C           NORMALIZE REAL VECTORS.                ESV11820
            C                                                  ESV11830
ISN 0175         655 CERR # 0.0                                ESV11840
ISN 0176             DCERR#0.0D0
ISN 0177             IF%IVC2< 658, 662, 658                    ESV11850
ISN 0178        658  C1#C.CD0
ISN 0179             C2#C.CD0
ISN 0180             DO 66C L # 1, N                           ESV11870
ISN 0181             TEMP#DABS%XI%L<<
ISN 0182             IF%TEMP - C1< 660, 660, 659               .SV11890
ISN 0183         659 C1 # TEMP                                 ESV11900
ISN 0184             C2 # XI%L<                                ESV11910
ISN 0185         660 CONTINUE                                  ESV11920
ISN 0186             DO 661 L # 1, N                           ESV11930
ISN 0187             XI%L< # XI%L</C2                          ESV11940
ISN 0188             DCERR#DMAX1%DCERR,DABS%XI%L<-XR%L<<<
ISN 0189         661 XR%L< # XI%L<                             ESV11960
ISN 0190             IF%IVC1< 662, 638, 662                    ESV11970
ISN 0191        662  C2#C.CD0
ISN 0192             C1#C.0D0
ISN 0193             DO 664 L # 1, N                           CSV11990
ISN 0194             TEMP#DABS%VI%L<<
ISN 0195             IF%TEMP - C1< 664, 664, 663               ESV12010
ISN 0196         663 C1 # TEMP                                 ESV12C20
ISN 0197             C2 # VI%L<                                ESV12C30
ISN 0198         664 CONTINUE                                  ESV12040
ISN 0199             DO 665 LL # 1, N                          ESV12050
ISN 0200             VI%LL< # VI%LL</C2                        CSV12060
ISN 0201             DCERR#DMAX1%DCERR,DABS%VI%LL<-W%LL,1<<<
ISN 0202             W%LL,1<#VI%LL<
```

```
      665 VK%L1<#W%LL,I<
          GO TC 638
      668 IF%IVC2< 669, 671, 669
      669 DO 670 L # 1, N
     670  Y1%L<#0.0D0
          IF%IVC1< 671, 70, 671
      671 DO 672 L # 1, N
     672  V1%L<#0.0D0
       70 RETURN
      673 IF%IVC2< 674, 502, 674
      674 DO 675 I # 1, N
          I2 # IKO#%I,2<
      675 X1%I2< # XR%I<
          GO TC 500
C
C         BACK SUBSTITUTION SECTION.
C
      499 IF%IVC2< 500, 502, 500
      500 DO 501 I # 2, N
          I1 # I - 1
          DO 501 J # 1, I1
      501 X1%I< # X1%I< - B%I,J<*X1%J<
      511 IF%IVC1< 502, 514, 502
      502 DO 510 I # 1, N
          I1 # I - 1
          IF%I1< 503, 505, 503
      503 DO 504 J # 1, I1
      504 V1%I< # V1%I< - B%J,I<*V1%J<
          IF%ICC< 505, 506, 505
      505 IF%B%I,I<< 506, 507, 506
      506 V1%I< # V1%I</B%I,I<
          GO TC 510
      507 IF%V1%I<< 508, 509, 508
      508 V1%I< # V1%I<*1.0E&15
          GO TC 510
      509 V1%I< # 1.0
      510 CONTINUE
          IF%IVC2< 514, 525, 514
      514 DC 522 I # 1, N
          IR # NP1 - I
          IF%I - 1< 515, 517, 515
      515 I2 # IR & 1
          DO 516 J # I2, N
      516 X1%IR< # X1%IR< - B%IR,J<*X1%J<
          IF%ICC< 517, 518, 517
      517 IF%B%IR,IR<< 518, 519, 518
      518 X1%IR< # X1%IR</B%IR,IR<
          GO TC 522
      519 IF%X1%IR<< 520, 521, 520
      520 X1%IR< # X1%IR<*1.0E&15
          GO TC 522
     521  X1%IR<#1.0D0
      522 CONTINUE
          IF%IVC1< 525, 529, 525
      525 DO 526 I # 2, N
          IK # NP1 - I
          I2 # IK & 1
          DO 526 J # I2, N
```

```
ISN 0258       526 VI%IR< # VI%IR< - B%J,IR<*VI%J<                          ESV12660
ISN 0259           DO 527 I # 1, N                                          ESV12670
ISN 0260           I2 # IROW%L,I<                                           ESV12680
ISN 0261       527 VR%I2< # VI%L<                                           ESV12690
ISN 0262           DO 528 L # 1, N                                          ESV12700
ISN 0263       528 VI%L< # VR%L<                                            ESV12710
ISN 0264       529 IF%ICCI1< 615, 655, 615                                  ESV12720
              C                                                             ESV12730
              C        FACTOR MATRIX.                                       ESV12740
              C                                                             ESV12750
ISN 0265       700 ICC # 0                                                  ESV12760
ISN 0266           SWI#1.0D72
ISN 0267           DO 701 LL # 1, N                                         ESV12780
ISN 0268       701 IROW%LL,I< # LL                                          ESV12790
ISN 0269           DO 708 K # 1, N1                                         ESV12800
ISN 0270           AMAX#DABS%B%K,K<<
ISN 0271           IMAX # K                                                 ESV12820
ISN 0272           K1 # K & 1                                               ESV12830
ISN 0273           DO 702 I # K1, N                                         ESV12840
ISN 0274           IF%AMAX.GT.DABS%B%I,K<<< GO TO 702
ISN 0276           AMAX#DABS%B%I,K<<
ISN 0277           IMAX # I                                                 ESV12870
ISN 0278       702 CONTINUE                                                 ESV12880
ISN 0279           IF%AMAX .LT. SWI< SWI # AMAX                             ESV12890
ISN 0281           IF%AMAX.GE.1.0D-25< GO TO 723
ISN 0283           B%K,K<#0.0D0
ISN 0284           ICC # ICC & 1                                            ESV12920
ISN 0285           GO TO 708                                                ESV12930
ISN 0286       723 IF%IMAX .EQ. K< GO TO 704                                ESV12940
ISN 0288           DO 703 J # 1, N                                          ESV12950
ISN 0289           AMAX # B%K,J<                                            ESV12960
ISN 0290           B%K,J< # B%IMAX,J<                                       ESV12970
ISN 0291       703 B%IMAX,J< # AMAX                                         ESV12980
ISN 0292           I2 # IROW%K,I<                                           ESV12990
ISN 0293           IROW%K,I< # IROW%IMAX,I<                                 ESV13000
ISN 0294           IROW%IMAX,I< # I2                                        ESV13010
ISN 0295       704 DO 707 I # K1, N                                         ESV13020
ISN 0296           IF%B%I,K<< 705, 707, 705                                 ESV13030
ISN 0297       705 B%I,K< # B%I,K</B%K,K<                                   ESV13040
ISN 0298           DO 706 J # K1, N                                         ESV13050
ISN 0299       706 B%I,J< # B%I,J< - B%K,J<*B%I,K<                          ESV13060
ISN 0300       707 CONTINUE                                                 ESV13070
ISN 0301       708 CONTINUE                                                 ESV13080
ISN 0302           AMAX#DABS%B%N,N<<
ISN 0303           IF%AMAX-1.0D-25< 712,712,713
ISN 0304       712 B%N,N<#0.0D0
ISN 0305           SWI#0.0D0
ISN 0306           ICC # ICC & 1                                            ESV13120
ISN 0307           GO TO 709                                                ESV13130
ISN 0308       713 IF%AMAX .LT. SWI< SWI # AMAX                             ESV13140
ISN 0310       709 IF%ICC .LE. ISW< GO TO 710                               ESV13150
ISN 0312           IF%KK< 1050,1050,1051
ISN 0313      1050 WRITE%IO3,1O2< ICC
ISN 0314           COUNTF # 0                                               ESV13180
ISN 0315           RETURN                                                   ESV13190
ISN 0316      1051 WRITE%IO3,1052< ICC
ISN 0317       710 DO 711 LI # 1, N                                         ESV13210
ISN 0318           I2 # IROW%LL,I<                                          ESV13220
```

```
ISN 0319        711 IROW(I2,2< # LL                                                  ESV13230
ISN 0320            IF(ABC(I)< 607, 652, 607                                         ESV13240
ISN 0321       1052 FORMAT(///23H ****** WARNING ******     ,@   SUBROUTINE EIGVEC HAS ESV13250
                   1FOUND AN EIGENVALUE OF APPARENT MULTIPLICITY@,                   ESV13260
                   1                                      I4,/23X,@   COMPUTATION OF EIESV13270
                   2GENVECTORS< CONTINUES AT USER S OPTION@//<                       ESV13280
ISN 0322        101 FORMAT(38HOMORE THAN 15 LOOPS FOR EIGENVECTOR OF,2E12.4,         ESV13290
                   2  14H DIFFERENCE OF,E12.4<                                       ESV13300
ISN 0323        102 FORMAT(16HO****WARNING**** , I4, 71H ZEROS ON DIAGONAL OF FACTOREDESV13310
                   1 MATRIX.  CHECK FOR MULTIPLE EIGENVALUES./20X,                   ESV13320
                   2@ SUBROUTINE EIGVEC WILL NOT PERFORM COMPUTATION FOR THIS EIGENVECESV13330
                   3TOR @//<                                                         ESV13340
ISN 0324            END
```

```
LEVEL 18 ( SEPT 69 )                         OS/360  FORTRAN H                                  DATE  71.106/19.50.4

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                                   SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002             SUBROUTINE SINGLE%SB,D,RI,GG,IMULT,NCON,NS,NC,MD<
                 C
                 C     PROGRAM CONVERTS MULTI-INPUT SYSTEM INTO PSEUDO SINGLE-INPUT
                 C         SYSTEM
                 C
  ISN 0003             REAL*8 SB%MD,MD<,D%MD<,RI%MD<,GG%MD<,P1,PIV,DABS,GSE
  ISN 0004             DIMENSION IMULT%MD,2<
                 C
                 C     PROGRAM CHECKS CONTROLLABILITY OF %A,B<
                 C
  ISN 0005             NP#0
  ISN 0006             GSE#0.D0
  ISN 0007             DO 100 J#1,NC
  ISN 0008             DO 100 I#1,NS
  ISN 0009         100 GSE#GSE&DABS%SB%I,J<<
  ISN 0010             GSE#GSE/%NS*NC<
  ISN 0011             DO 140 I#1,NS
  ISN 0012             NCON#0
  ISN 0013             IMULT%I,2<#0
  ISN 0014             DO 110 J#1,NC
  ISN 0015         110 IF%%DABS%SB%I,J<<-GSE*1.D-8< .GT. 0.D0< NCON#NCON&1
  ISN 0017             IF%DABS%RI%I<<-1.D-8< 130,130,120
  ISN 0018         120 NP#NP&1
  ISN 0019             IF%NCON .EQ. 0< IMULT%I,2<#1
  ISN 0021             IF%NP-1< 140,140,125
  ISN 0022         125 I1#I-1
  ISN 0023             IF%%IMULT%I,2<&IMULT%I1,2<< .EQ. 2< GO TO 330
  ISN 0025             NP#0
  ISN 0026             GO TO 140
  ISN 0027         130 IF%NCON .EQ. 0< GO TO 330
  ISN 0029         140 CONTINUE
                 C
                 C     COMPUTATION OF SINGLE-INPUT VECTOR  D # SB*G .
                 C
  ISN 0030             DO 170 I#1,NS
  ISN 0031             D%I<#0.D0
  ISN 0032             DO 170 J#1,NC
  ISN 0033         170 D%I<#D%I<&SB%I,J<
  ISN 0034             DO 180 I#1,NC
  ISN 0035         180 GG%I<#1.D0
  ISN 0036             IF%NC .EQ. 1< GO TO 325
                 C
                 C     TEST WHETHER  D  RENDERS %L,D< CONTROLLABLE
                 C
  ISN 0038             NI#1
  ISN 0039         185 NP#0
  ISN 0040             I#NI-1
  ISN 0041         188 I#I&1
  ISN 0042             NCON#0
  ISN 0043             NI#I
  ISN 0044             IF%%DABS%D%I<< .GT. GSE*1.D-8< NCON#1
  ISN 0046             IF%DABS%RI%I<<-1.D-8< 210,210,190
  ISN 0047         190 NP#NP&1
  ISN 0048             IF%NCON .EQ. 0< IMULT%I,2<#1
  ISN 0050             IF%NP-1< 220,220,200
```

```
ISN 0051         200 I1#I-1
ISN 0052             IF%%IPULT%I,2<<IPULT%I1,2<< .EQ. 2< GO TO 230
ISN 0054             NP#0
ISN 0055             GO TO 220
ISN 0056         210 IF%NCON .EQ. 0< GO TO 230
ISN 0058         220 IF%I-NS< 188,225,225
ISN 0059         225 GO TO 325
               C
               C     FIND NON-ZERO ELEMENT IN ROW N1 OF MATRIX  SB .
               C
ISN 0060         230 PIV#SB%N1,1<
ISN 0061             MSB#1
ISN 0062             DO 250 I#2,NC
ISN 0063             IF%DABS%PIV<-DABS%SB%N1,I<<< 240,250,250
ISN 0064         240 PIV#SB%N1,I<
ISN 0065             MSB#I
ISN 0066         250 CONTINUE
               C
               C     FIND ELEMENT OF LARGEST MAGNITUDE, PIV, IN COL.-NO. MSB OF MATRIX
               C         SB. FIND NON-ZERO ELEMENT OF SMALLEST MAGNITUDE, P1, IN VEC. D
               C
ISN 0067         260 DO 270 I#1,NS
ISN 0068             N2#I
ISN 0069             IF%DABS%D%I<<-GSE*1.D-8< 280,270,280
ISN 0070         270 CONTINUE
ISN 0071         280 P1#D%N2<
ISN 0072             DO 290 I#1,NS
ISN 0073             IF%DABS%PIV< .LT. DABS%SB%I,MSB<<< PIV#SB%I,MSB<
ISN 0075             IF%DABS%D%I<< .LT. GSE*1.D-8< GO TO 290
ISN 0077             IF%DABS%P1< .LT. DABS%D%I<<< P1#D%I<
ISN 0079         290 CONTINUE
ISN 0080             P1#DABS%PIV/P1<&1.D-8
ISN 0081             N2#P1&1
ISN 0082             P1#N2
ISN 0083             DO 300 I#1,NS
ISN 0084         300 D%I<#P1*D%I<&SB%I,MSB<
ISN 0085             DO 310 I#1,NC
ISN 0086         310 GG%I<#P1*GG%I<
ISN 0087             GG%MSB<#GG%MSB<&1.D0
ISN 0088             IF%N1-NS< 320,325,325
ISN 0089         320 N1#N1&1
ISN 0090             GO TO 185
ISN 0091         325 NCON#1
ISN 0092             GO TO 340
ISN 0093         330 NCON#0
ISN 0094         340 RETURN
ISN 0095             END
```

```
VEL 18 ( SEPT 69 )                      OS/360  FORTRAN H                          DATE  71.106/19.51.54

       COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                          SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002             SUBROUTINE SIMTP2%A,AA,AA1,S,SINV,W,IROW,RR,RI,XR,XI,VR,VI,NS,MD,
                    SIVC<
               C
               C     COMPUTES SIMILARITY TRANSFORMATION MATRIX  SINV  FOR MATRIX  A
               C         WITH SIMPLE EIGENVALUES. YIELDS REAL-VALUED TRANSFORMATION
               C         MATRIX.
               C
ISN 0003             REAL*8 A%MD,MD<,AA%MD,MD<,AA1%MD,MD<,SINV%MD,MD<,RR%MD<,RI%MD<
ISN 0004             REAL*8 XR%MD<,XI%MD<,VR%MD<,VI%MD<,DABS,DSQRT
ISN 0005             REAL*8 W%MD,4<,S%MD,MD<,SW1
ISN 0006             DIMENSION IROW%MD,2<
ISN 0007          10 FORMAT(//T3,'EIGENVECTOR ERROR MESSAGE')
ISN 0008          20 FORMAT(T3,'SW1=',E10.4,10X,'ITER=',I5,10X,'DIF=',E10.4)
ISN 0009             K=0
ISN 0010         100 CONTINUE
ISN 0011             DO 110 J=1,NS
ISN 0012             DO 110 I=1,NS
ISN 0013         110 AA1(I,J)=AA(I,J)
ISN 0014             K=K+1
ISN 0015             CALL EIGVEC%IVC,A,AA1,W,IROW,XR,XI,VR,VI,RR%K<,RI%K<,NS,MD,0,SW1,
                    EITER,DIF,2<
ISN 0016             IF(ITER .LT. 15) GO TO 111
ISN 0018             WRITE%3,10<
ISN 0019             WRITE%3,20< SW1,ITER,DIF
ISN 0020         111 CONTINUE
ISN 0021             IF%DABS%RI%K<< .GT. 1.D-8< GO TO 130
               C
               C     COL. AND/OR ROW EIGENVECTORS CORRESPONDING TO A REAL EIGENVALUE
               C
ISN 0023             W(1,1)=0.D0
ISN 0024             DO 120 I=1,NS
ISN 0025             W(1,1)=W(1,1)+DABS(VR(I))
ISN 0026         120 SINV%K,I<=VR%I<
ISN 0027             IF%IVC-2< 126,126,122
ISN 0028         122 W(1,3)=0.D0
ISN 0029             DO 124 I=1,NS
ISN 0030             SINV(K,I)=SINV(K,I)/W(1,1)
ISN 0031             W(1,3)=W(1,3)+SINV(K,I)*XR(I)
ISN 0032         124 S%I,K<=XR%I<
ISN 0033             DO 123 I=1,NS
ISN 0034         123 S(I,K)=S(I,K)/W(1,3)
ISN 0035         126 IF%K-NS< 100,150,150
               C
               C     COMPLEX COL. AND/OR ROW EIGENVECTORS ARE CONVERTED TO A SET OF TWO
               C          REAL-VALUED TRANSFORMATION VECTORS
               C
ISN 0036         130 K1=K+1
ISN 0037             W(1,1)=0.D0
ISN 0038             W(1,2)=0.D0
ISN 0039             DO 140 I=1,NS
ISN 0040             W(1,1)=W(1,1)+DABS(VR(I))
ISN 0041             W(1,2)=W(1,2)+DABS(VI(I))
ISN 0042             SINV%K,I<=2.D0*VR%I<
ISN 0043         140 SINV%K1,I<=-2.D0*VI%I<
ISN 0044             IF%IVC-2< 136,136,132
```

```
 SN 0045          132 IF(W(I,1) .LT. W(I,2)) W(I,1)=W(I,2)
 SN 0047              W(I,3)=0.D0
 SN 0048              W(I,4)=0.D0
 SN 0049              DO 134 I=1,NS
ISN 0050              SINV?K,I<#SINV?K,I</?2.D0*W?I,I<<
ISN 0051              SINV?K1,I<#SINV?K1,I</?2.D0*W?I,I<<
ISN 0052              W(I,3)=W(I,3)+.5D0*SINV(K,I)*XR(I)+.5D0*SINV(K1,I)*XI(I)
ISN 0053          134 W(I,4)=W(I,4)+.5D0*SINV(K,I)*XI(I)-.5D0*SINV(K1,I)*XR(I)
ISN 0054              W(I,1)=W(I,3)*W(I,3)+W(I,4)*W(I,4)
 SN 0055              DO 135 I=1,NS
 SN 0056              S(I,K)=(XR(I)*W(I,3)+XI(I)*W(I,4))/W(I,1)
 SN 0057          135 S(I,K1)=(XI(I)*W(I,3)-XR(I)*W(I,4))/W(I,1)
 SN 0058          136 K#K1
 SN 0059              IF?K-NS< 100,150,150
 SN 0060          150 RETURN
 SN 0061              END
```

```
LEVEL 18 ( SEPT 69 )                      OS/360  FORTRAN H                    DATE  71.106/19.43.27

         COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                            SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002               SUBROUTINE MVECT?A,AV,NS,MD,MD2<
                C
                C        CONVERTS MATRIX A INTO VECTOR AV
                C
  ISN 0003               REAL*8 A?MD,MD<,AV?MD2<
  ISN 0004               DO 10 J#1,NS
  ISN 0005               DO 10 I#1,NS
  ISN 0006               K#?J-1<*NS&I
  ISN 0007            10 AV?K<#A?I,J<
  ISN 0008               RETURN
  ISN 0009               END
```

```
LEVEL 18 ( SEPT 69 )                      OS/360  FORTRAN H                    DATE  71.106/19.49.27

         COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                            SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002               SUBROUTINE MMULT?A,B,C,NS,NC,NB,MD<
                C
                C        COMPUTES MATRIX PRODUCT  C # A*B
                C
  ISN 0003               REAL*8 A?MD,MD<,B?MD,MD<,C?MD,MD<
  ISN 0004               DO 10 I#1,NS
  ISN 0005               DO 10 J#1,NB
  ISN 0006               C?I,J<#0.D0
  ISN 0007               DO 10 K#1,NC
  ISN 0008            10 C?I,J<#C?I,J<&A?I,K<*B?K,J<
  ISN 0009               RETURN
  ISN 0010               END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                           SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
 ISN 0002             SUBROUTINE POLES(A,AD,AV,A1,A2,B,B1,B2,SV,G,RR,RI,SINV,TV,TVA,T,W,
                    P                  X,XI,XR,IANA,IROW,MD,MD2,NS,NC,NREP,K1,K2)
                  C
                  C
                  C   FINDS FEEDBACK GAINS FOR A SET OF SPECIFIED EIGENVALUES AND
                  C       ALL-STATE FEEDBACK. RESULT IS IN MATRIX G .
                  C
                  C
 ISN 0003             REAL*8 A(MD,MD),AD(MD,MD),AV(MD2),A1(MD,MD),A2(MD,MD),B(MD,MD),
                    1        B1(MD,MD),B2(MD,MD),RR(MD,2),G(MD,MD),RR(MD),RI(MD),SV(MD),
                    2        SVR(MD),SINV(MD,MD),T(MD,MD),W(MD,4),X(MD,1),XI(MD),XR(MD)
 ISN 0004             DIMENSION IANA(MD),IROW(MD,2)
 ISN 0005             DOUBLE PRECISION DABS,TOL
 ISN 0006          10 FORMAT(//,'  THE PAIR (A,B) IS UNCONTROLLABLE, NCON = 0 .')
                  C 20 FORMAT(//,'  EIGENVALUES OF CLOSED-LOOP SYSTEM FOR ALL-STATE FEEDB
                  C    2ACK.')
                  C 30 FORMAT(4F20.8)
 ISN 0007          40 FORMAT(//,'  IER .NE. 0 . THUS, (A,B) SEEMS TO BE MARGINALLY CONTR
                      4OLLABLE ONLY.')
 ISN 0008          50 FORMAT(//,'  SYSTEM OR COMPENSATOR OF ORDER 0 .PROGRAM STOPS. CHEC
                      5K INPUT MATRICES.')
                  C
                  C   COMPUTATION OF EIGENVALUES AND ROW EIGENVECTORS OF MATRIX A .
                  C
 ISN 0009             IF(NREP .EQ. 1) GO TO 170
 ISN 0011             IF(NS-1) 245,90,94
 ISN 0012          90 RR(1)=A(1,1)
 ISN 0013             RI(1)=0.D0
 ISN 0014             SINV(1,1)=1.D0
 ISN 0015             I=0
 ISN 0016             XR(1)=B(1,1)
 ISN 0017             IV=1
 ISN 0018          91 I=I+1
 ISN 0019             IF(I .GT. NC) GO TO 92
 ISN 0021             IF(DABS(XR(1)) .GE. DABS(B(1,I))) GO TO 91
 ISN 0022             XR(1)=B(1,I)
 ISN 0024             IV=I
 ISN 0025             GO TO 91
 ISN 0026          92 IF(DABS(XR(1)) .LE. 1.D-8) GO TO 100
 ISN 0028             DO 93 I=1,NC
 ISN 0029          93 XI(I)=0.D0
 ISN 0030             XI(IV)=1.D0
 ISN 0031             NCON=1
 ISN 0032             GO TO 110
 ISN 0033          94 CONTINUE
 ISN 0034             CALL MVECT(A,AV,NS,MD,MD2)
 ISN 0035             CALL HSSG(NS,AV,NS,MD2)
 ISN 0036             CALL ATEIG(NS,AV,RR,RI,IANA,NS,MD,MD2)
 ISN 0037             CALL MMULT(A,A,A1,NS,NS,NS,MD)
 ISN 0038             CALL SIMTR2(A,A1,A2,B2,SINV,W,IROW,RR,RI,XR,XI,SV,SVR,NS,MD,2)
 ISN 0039             CALL MMULT(SINV,B,B1,NS,NS,NC,MD)
                  C
                  C
                  C   TRANSFORMATION OF (A,B) TO SINGLE-INPUT SYSTEM.
                  C
                  C
 ISN 0040             CALL SINGLE(B1,XR,B1,XI,IROW,NCON,NS,NC,MD)
 ISN 0041             IF(NCON) 100,100,110
 ISN 0042         100 WRITE(3,10)
```

```
ISN 0043            STOP
ISN 0044        110 DO 130 I=1,NS
ISN 0045            DO 120 J=1,NS
ISN 0046        120 AD(I,J)=0.D0
ISN 0047        130 AD(I,I)=RR(I)
ISN 0048            I=0
ISN 0049        140 I=I+1
ISN 0050            IF(DABS(RI(I))-1.D-8) 160,160,150
ISN 0051        150 IP=I+1
ISN 0052            AD(I,IP)=RI(I)
ISN 0053            AD(IP,I)=-RI(I)
ISN 0054            I=IP
ISN 0055        160 IF(I-NS) 140,170,170
ISN 0056        170 CONTINUE
ISN 0057            DO 190 I=1,NS
ISN 0058            DO 180 J=1,NS
ISN 0059        180 B2(I,J)=0.D0
ISN 0060        190 B2(I,1)=XR(I)
ISN 0061            IGAIN=1
ISN 0062            CALL APRFIS(AD,B2,EM,G,T,A1,A2,W1,X,SV,SVR,NS,1,K1,K2,IER,MD,IROW)
ISN 0063            IF(IER) 240,200,240
ISN 0064        200 CONTINUE
              C 200 WRITE(3,20)
              C     DO 210 I=1,NS
              C 210 WRITE(3,30) (EM(I,J),J=1,Z)
ISN 0065            DO 230 I=1,NC
ISN 0066        230 B1(I,1)=XI(I)
ISN 0067            CALL MMULT(B1,G,A2,NC,1,NS,MD)
ISN 0068            CALL MMULT(A2,SINV,G,NC,NS,NS,MD)
ISN 0069            GO TO 250
ISN 0070        240 WRITE(3,40)
ISN 0071            STOP
ISN 0072        245 WRITE(3,50)
ISN 0073            STOP
ISN 0074        250 RETURN
ISN 0075            END
```

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                              DATE  71.106/19.52.2

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=C000K,
                             SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              SUBROUTINE MCOMP%NS,NC,NF,NFF,MD,MD2,NREP,K1,K2,K3,K4<
                C
                C        COMPUTATION OF THE COMPENSATOR MATRICES, FF,FG,FH,FJ.
                C
  ISN 0003              REAL*8 A%6,6<,B%6,6<,C%6,6<,AH%12,12<,A2%12,12<,A2S%12,12<,AV%144<
  ISN 0004              REAL*8 FF%6,6<,FG%6,6<,FH%6,6<,FJ%6,6<,GF%6,6<,GG%6,6<,GH%6,6<
  ISN 0005              REAL*8 GJ%6,6<,GFM%6,6<,GGM%6,6<,GHM%6,6<,GJM%6,6<,GRF%6,6<
  ISN 0006              REAL*8 GRG%6,6<,GRH%6,6<,GRJ%6,6<,CVR%12,12<,CVI%12,12<,RR%12<
  ISN 0007              REAL*8 RI%12<,VR%12<,VI%12<,XR%12<,XI%12<,VRN%12,1<,VIN%12,1<
  ISN 0008              REAL*8 W(12,4),EM(12,2)
  ISN 0009              REAL*8 AVF%36<,XRS%6<,XRF%6<,XIS%6<,XIF%6<,VRS%6<,VRF%6<,VIS%6<
  ISN 0010              REAL*8 VIF%6<,RRS%6<,RRF%6<,RIS%6<,RIF%6<,VRNS%6,1<,VRNF%6,1<
  ISN 0011              REAL*8 VINS%6,1<,VINF%6,1<,AVS%36<,WS%6,4<,WF%6,4<
  ISN 0012              REAL*8 EMS%6,2<,EMF%6,2<
  ISN 0013              DIMENSION IANA%12<,IANAS%6<,IANAF%6<,IROW%12,2<,IROWS%6,2<
  ISN 0014              DIMENSION IROWF(6,2)
  ISN 0015              EQUIVALENCE %AV%1<,AVF%1<<,%AV%37<,AVS%1<<
  ISN 0016              EQUIVALENCE %EM%1<,EMF%1<<,%EM%13<,EMS%1<<
  ISN 0017              EQUIVALENCE %XR%1<,XRS%1<<,%XR%7<,XRF%1<<,%XI%1<,XIS%1<<
  ISN 0018              EQUIVALENCE %XI%7<,XIF%1<<,%VR%1<,VRS%1<<,%VR%7<,VRF%1<<
  ISN 0019              EQUIVALENCE %VI%1<,VIS%1<<,%VI%7<,VIF%1<<,%RR%1<,RRS%1<<
  ISN 0020              EQUIVALENCE %RR%7<,RRF%1<<,%RI%1<,RIS%1<<,%RI%7<,RIF%1<<
  ISN 0021              EQUIVALENCE %VRN%1<,VRNS%1<<,%VRN%7<,VRNF%1<<
  ISN 0022              EQUIVALENCE %VIN%1<,VINS%1<<,%VIN%7<,VINF%1<<
  ISN 0023              EQUIVALENCE %W%1<,WF%1<<,%W%25<,WS%1<<
  ISN 0024              EQUIVALENCE %IANA%1<,IANAS%1<<,%IANA%7<,IANAF%1<<
  ISN 0025              EQUIVALENCE %IROW%1<,IROWS%1<<,%IROW%13<,IROWF%1<<
  ISN 0026              COMMON /MCM/ AH,A2,A2S,AV,CVR,CVI,W,A,B,C,FF,FG,FH,FJ,GF,GG,GH,
                      1              GJ,GFM,GGM,GHM,GJM,GRF,GRG,GRH,GRJ,EM,VRN,VIN,RR,RI,
                      2              VR,VI,XR,XI,IROW,IANA
  ISN 0027              IF%NFF< 1561,1561,1500
  ISN 0028         1500 DO 1510 I#1,NFF
  ISN 0029              IIF#NF%I
  ISN 0030              DO 1510 J#1,NFF
  ISN 0031              JF#NF%J
  ISN 0032              FF%I,J<#A%IIF,JF<
  ISN 0033         1510 GRF%I,J<#A%JF,IIF<
  ISN 0034              DO 1520 I#1,NFF
  ISN 0035              IIF#NF%I
  ISN 0036              DO 1520 J#1,NF
  ISN 0037         1520 GJM%I,J<#A%J,IIF<
  ISN 0038              IF%K1 .EQ. 0< GO TO 1525
  ISN 0040              CALL POLES%GRF,GRG,AVF,GHM,FG,GJM,GGM,FH,EMF,GFM,RRF,RIF,GRJ,XIF,
                      P              XRF,GRH,WF,VRNS,VIF,VRF,IANAF,IROWF,MD,MD2,NFF,NF,NREP,
                      O              K1,K2<
  ISN 0041         1525 DO 1530 I#1,NFF
  ISN 0042              DO 1530 J#1,NFF
  ISN 0043              DO 1530 K#1,NF
  ISN 0044         1530 FF%I,J<#FF%I,J<-GFM%K,I<*GJM%J,K<
  ISN 0045              DO 1540 I#1,NFF
  ISN 0046              IIF#NF%I
  ISN 0047              DO 1540 J#1,NF
  ISN 0048              FG%I,J<#A%IIF,J<
  ISN 0049              DO 1540 K#1,NF
  ISN 0050         1540 FG%I,J<#FG%I,J<-GFM%K,I<*A%K,J<
```

```
ISN 0051          DO 1550 I#1,NFF
ISN 0052          DO 1550 J#1,NF
ISN 0053          DO 1550 K#1,NFF
ISN 0054     1550 FG%I,J<#FG%I,J<&FF%I,K<*GFM%J,K<
ISN 0055          DO 1560 I#1,NFF
ISN 0056          IIF#NF&I
ISN 0057          DO 1560 J#1,NC
ISN 0058          FH%I,J<#B%IIF,J<
ISN 0059          DO 1560 K#1,NF
ISN 0060          FH%I,J<#FH%I,J<-GFM%K,I<*B%K,J<
ISN 0061     1560 CONTINUE
ISN 0062     1561 CONTINUE
ISN 0063          IF%K3 .EQ. 0< GO TO 1563
ISN 0065          CALL POLES%A,GF,AVF,FJ,GGM,B,GHM,GJM,EMS,GG,RRF,RIF,GJ,XIF,XRF,GH,
                P              WF,VRNS,VIS,VRS,IANAF,IRJWF,WD,WO2,NS,NC,NREP,K3,K4<
ISN 0066     1563 IF%NFF< 1600,1600,1565
ISN 0067     1565 DO 1567 I#1,NF
ISN 0068          DO 1566 J#1,NF
ISN 0069     1566 GJM%I,J<#0.D0
ISN 0070     1567 GJM%I,I<#1.D0
ISN 0071          DO 1568 I#1,NFF
ISN 0072          INF#NF&I
ISN 0073          DO 1568 J#1,NF
ISN 0074     1568 GJM%INF,J<#GFM%J,I<
ISN 0075          DO 1569 I#1,NC
ISN 0076          DO 1569 J#1,NF
ISN 0077          GHM%I,J<#0.D0
ISN 0078          DO 1569 K#1,NS
ISN 0079     1569 GHM%I,J<#GHM%I,J<-GG%I,K<*GJM%K,J<
ISN 0080          DO 1570 I#1,NFF
ISN 0081          DO 1570 J#1,NF
ISN 0082          DO 1570 K#1,NC
ISN 0083     1570 FG%I,J<#FG%I,J<&FH%I,K<*GHM%K,J<
ISN 0084          DO 1590 I#1,NFF
ISN 0085          DO 1580 J#1,NFF
ISN 0086          JF#NF&J
ISN 0087          DO 1580 K#1,NC
ISN 0088     1580 FF%I,J<#FF%I,J<-FH%I,K<*GG%K,JF<
ISN 0089          DO 1590 I#1,NC
ISN 0090          DO 1590 J#1,NFF
ISN 0091          JF#NF&J
ISN 0092     1590 FH%I,J<#-GG%I,JF<
ISN 0093          GO TO 1645
ISN 0094     1600 DO 1620 I#1,NS
ISN 0095          DO 1610 J#1,NS
ISN 0096     1610 GJM%I,J<#0.D0
ISN 0097     1620 GJM%I,I<#1.D0
ISN 0098          IF%NFF< 1645,1645,1630
ISN 0099     1630 DO 1640 I#1,NFF
ISN 0100          IIF#NF&I
ISN 0101          DO 1640 J#1,NF
ISN 0102          GJM%IIF,J<#GF%J,I<
ISN 0103     1640 CONTINUE
ISN 0104     1645 CONTINUE
ISN 0105          DO 1650 I#1,NC
ISN 0106          DO 1650 J#1,NF
ISN 0107          FJ%I,J<#0.D0
ISN 0108          DO 1650 K#1,NS
```

```
ISN 0109      1650 FJ(I,J)=FJ(I,J)-GG(I,K)*GJM(K,J)
ISN 0110           RETURN
ISN 0111           END
```



```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              SUBROUTINE INPOL(T3,RR,RI,N,NRC,MD)
                  C
                  C     ROUTINE DETERMINES INITIAL SET OF EIGENVALUES.
                  C
  ISN 0003              REAL*8 RR(MD),RI(MD),T3
  ISN 0004           10 FORMAT(//,'  NS=0 AND/OR NFF=0 , CHECK INPUT DATA.')
  ISN 0005              IF(N .EQ. 0) GO TO 130
  ISN 0007              NPA=0
  ISN 0008          100 NPA=NPA+1
  ISN 0009              T3=T3+3.D0
  ISN 0010              N3=NPA/3
  ISN 0011              NA=NPA-3*N3
  ISN 0012              IF(NA .EQ. 0) GO TO 120
  ISN 0014              IF((N-NPA) .EQ. 0) GO TO 120
  ISN 0016              IF(NRC .EQ. 1) GO TO 120
  ISN 0018              RR(NPA)=T3
  ISN 0019              RI(NPA)=-T3
  ISN 0020              NP1=NPA+1
  ISN 0021              RR(NP1)=T3
  ISN 0022              RI(NP1)=T3
  ISN 0023              NPA=NP1
  ISN 0024              NRC=1
  ISN 0025          110 IF(NPA-N) 100,140,140
  ISN 0026          120 RR(NPA)=T3
  ISN 0027              RI(NPA)=0.D0
  ISN 0028              NRC=2
  ISN 0029              GO TO 110
  ISN 0030          130 WRITE(3,10)
  ISN 0031              STOP
  ISN 0032          140 RETURN
  ISN 0033              END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                           SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
 ISN 0002                SUBROUTINE DFMFP(FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H,MH,ICPLX,M2D1)
 ISN 0003                DOUBLE PRECISION DABS,DFLOAT,DSIGN,DBLE,DEXP,DLOG,DLOG10,DATAN
                        1,DSIN,DCOS,DSQRT,DTANH,DMOD,DMAX1,DMIN1
 ISN 0004                DIMENSION H(MH),X(M2D1),G(M2D1),ICPLX(M2D1)
 ISN 0005                DOUBLE PRECISION X,F,FX,FY,OLDF,HNRM,GNRM,H,G,DX,DY,ALFA,DALFA,
                        1AMBDA,T,Z,W
 ISN 0006                IER=0
 ISN 0007                KOUNT=0
 ISN 0008                NJUMP=0
 ISN 0009                CALL FUNCT(N,X,F,G,KOUNT,NJUMP,M2D1)
 ISN 0010                N2=N+N
 ISN 0011                N3=N2+N
 ISN 0012                N31=N3+1
 ISN 0013              1 K=N31
 ISN 0014                DO 4 J=1,N
 ISN 0015                H(K)=1.D0
 ISN 0016                NJ=N-J
 ISN 0017                IF(NJ)5,5,2
 ISN 0018              2 DO 3 L=1,NJ
 ISN 0019                KL=K+L
 ISN 0020              3 H(KL)=0.D0
 ISN 0021              4 K=KL+1
 ISN 0022              5 KOUNT=KOUNT +1
 ISN 0023                KLOOP=0
 ISN 0024                OLDF=F
 ISN 0025                DO 9 J=1,N
 ISN 0026                K=N+J
 ISN 0027                H(K)=G(J)
 ISN 0028                K=K+N
 ISN 0029                H(K)=X(J)
 ISN 0030                K=J+N3
 ISN 0031                T=0.D0
 ISN 0032                DO 8 L=1,N
 ISN 0033                T=T-G(L)*H(K)
 ISN 0034                IF(L-J)6,7,7
 ISN 0035              6 K=K+N-L
 ISN 0036                GO TO 8
 ISN 0037              7 K=K+1
 ISN 0038              8 CONTINUE
 ISN 0039              9 H(J)=T
 ISN 0040                DY=0.D0
 ISN 0041                HNRM=0.D0
 ISN 0042                GNRM=0.D0
 ISN 0043                DO 10 J=1,N
 ISN 0044                HNRM=HNRM+DABS(H(J))
 ISN 0045                GNRM=GNRM+DABS(G(J))
 ISN 0046             10 DY=DY+H(J)*G(J)
 ISN 0047                IF(DY)11,51,51
 ISN 0048             11 IF(HNRM/GNRM-EPS)51,51,12
 ISN 0049             12 FY=F
 ISN 0050                ALFA=2.D0*(EST-F)/DY
 ISN 0051                AMBDA=1.D0
 ISN 0052                IF(ALFA)15,15,13
 ISN 0053             13 IF(ALFA-AMBDA)14,15,15
 ISN 0054             14 AMBDA=ALFA
 ISN 0055             15 ALFA=0.D0
 ISN 0056             16 FX=FY
```

```
ISN 0057          DX=DY
ISN 0058          DO 17 I=1,N
ISN 0059       17 X(I)=X(I)+AMBDA*H(I)
ISN 0060          KLOOP#KLOOP&1
ISN 0061          CALL FUNCT%N,X,F,G,KOUNT,NJUMP,M2DI<
ISN 0062          IF%KLOOP .GT. 20< GO TO 50
ISN 0064          FY=F
ISN 0065          DY=0.D0
ISN 0066          DO 18 I=1,N
ISN 0067       18 DY=DY+G(I)*H(I)
ISN 0068          IF(DY)19,36,22
ISN 0069       19 IF(FY-FX)20,22,22
ISN 0070       20 AMBDA=AMBDA+ALFA
ISN 0071          ALFA=AMBDA
ISN 0072          IF(HNRM*AMBDA-1.D10)16,16,21
ISN 0073       21 IER=2
ISN 0074          RETURN
ISN 0075       22 T=0.D0
ISN 0076       23 KLOOP#KLOOP&1
ISN 0077          IF%KLOOP .GT. 20< GO TO 50
ISN 0079          IF%AMBDA< 24,36,24
ISN 0080       24 Z=3.D0*(FX-FY)/AMBDA+DX+DY
ISN 0081          ALFA=DMAX1(DABS(Z),DABS(DX),DABS(DY))
ISN 0082          DALFA=Z/ALFA
ISN 0083          DALFA=DALFA*DALFA-DX/ALFA*DY/ALFA
ISN 0084          IF(DALFA)51,25,25
ISN 0085       25 W=ALFA*DSQRT(DALFA)
ISN 0086          ALFA=DY-DX+W+W
ISN 0087          IF(ALFA)250,251,250
ISN 0088      250 ALFA=(DY-Z+W)/ALFA
ISN 0089          GO TO 252
ISN 0090      251 ALFA=(Z+DY-W)/(Z+DX+Z+DY)
ISN 0091      252 ALFA=ALFA*AMBDA
ISN 0092          DO 26 I=1,N
ISN 0093       26 X(I)=X(I)+(T-ALFA)*H(I)
ISN 0094          CALL FUNCT%N,X,F,G,KOUNT,NJUMP,M2DI<
ISN 0095          IF(F-FX)27,27,28
ISN 0096       27 IF(F-FY)36,36,28
ISN 0097       28 DALFA=0.D0
ISN 0098          DO 29 I=1,N
ISN 0099       29 DALFA=DALFA+G(I)*H(I)
ISN 0100          IF(DALFA)30,33,33
ISN 0101       30 IF(F-FX)32,31,33
ISN 0102       31 IF(DX-DALFA)32,36,32
ISN 0103       32 FX=F
ISN 0104          DX=DALFA
ISN 0105          T=ALFA
ISN 0106          AMBDA=ALFA
ISN 0107          GO TO 23
ISN 0108       33 IF(FY-F)35,34,35
ISN 0109       34 IF(DY-DALFA)35,36,35
ISN 0110       35 FY=F
ISN 0111          DY=DALFA
ISN 0112          AMBDA=AMBDA-ALFA
ISN 0113          GO TO 22
ISN 0114       36 IF(OLDF-F+EPS)51,38,38
ISN 0115       38 DO 37 J=1,N
ISN 0116          K=N+J
```

```
      H(K)=G(J)-H(K)
      K=N+N
   37 H(K)=X(J)-H(K)
      IER=0
      IF(KOUNT-N)42,39,39
   39 T=0.D0
      Z=0.D0
      DO 40 J=1,N
      K=N+J
      W=H(K)
      K=K+N
      T=T+DABS(H(K))
   40 Z=Z+W*H(K)
      IF(HNRM-EPS)41,41,42
   41 IF(T-EPS)56,56,42
   42 IF(KOUNT-LIMIT)43,50,50
   43 ALFA=0.D0
      DO 47 J=1,N
      K=J+N3
      W=0.D0
      DO 46 L=1,N
      KL=N+L
      W=W+H(KL)*H(K)
      IF(L-J)44,45,45
   44 K=K+N-L
      GO TO 46
   45 K=K+1
   46 CONTINUE
      K=N+J
      ALFA=ALFA+W*H(K)
   47 H(J)=W
      IF(Z*ALFA)48,1,48
   48 K=N31
      DO 49 L=1,N
      KL=N2+L
      DO 49 J=L,N
      NJ=N2+J
      H(K)=H(K)+H(KL)*H(NJ)/Z-H(L)*H(J)/ALFA
   49 K=K+1
      GO TO 5
   50 IER=1
      RETURN
   51 DO 52 J=1,N
      K=N2+J
   52 X(J)=H(K)
      CALL FUNCT(N,X,F,G,KOUNT,NJUMP,M2D1C
      IF(GNRM-EPS)55,55,53
   53 IF(IER)56,54,54
   54 IER=-1
      GOTO 1
   55 IER=0
   56 RETURN
      END
```

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                    DATE  71.119/00.36.2

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
   ISN 0002          SUBROUTINE SORT%EMS,EMF,RR,RI,SV,NS,NFF,ND,M2D<
               C
               C     COMPARES EM  AND ACTUAL EIGENVALUES.
               C
   ISN 0003          REAL*8 SV%M2D<,RR%M2D<,RI%M2D<,EMS%ND,2<,EMF%ND,2<,DIF,DABS
   ISN 0004          IA#NS&NFF
   ISN 0005          I#0
   ISN 0006      100 I#I&1
   ISN 0007          IF%I .GT. NS< GO TO 140
   ISN 0009          DO 110 J#1,IA
   ISN 0010      110 SV%J<#DABS%RR%J<-EMS%I,1<<&DABS%RI%J<-EMS%I,2<<
   ISN 0011          DIF#SV%1<
   ISN 0012          IEM#1
   ISN 0013          DO 120 J#1,IA
   ISN 0014          IF%DIF .LE. SV%J<< GO TO 120
   ISN 0016          DIF#SV%J<
   ISN 0017          IEM#J
   ISN 0018      120 CONTINUE
   ISN 0019          EMS%I,1<#RR%IEM<
   ISN 0020          EMS%I,2<#RI%IEM<
   ISN 0021          IF%DABS%RI%IEM<<-1.D-8< 100,100,130
   ISN 0022      130 I#I&1
   ISN 0023          EMS%I,1<#RR%IEM<
   ISN 0024          EMS%I,2<#-RI%IEM<
   ISN 0025          GO TO 100
   ISN 0026      140 IF%NFF .LT. 1< GO TO 190
   ISN 0028          I#0
   ISN 0029      150 I#I&1
   ISN 0030          IF%I .GT. NFF< GO TO 190
   ISN 0032          DO 160 J#1,IA
   ISN 0033      160 SV%J<#DABS%RR%J<-EMF%I,1<<&DABS%RI%J<-EMF%I,2<<
   ISN 0034          DIF#SV%1<
   ISN 0035          IEM#1
   ISN 0036          DO 170 J#1,IA
   ISN 0037          IF%DIF .LE. SV%J<< GO TO 170
   ISN 0039          DIF#SV%J<
   ISN 0040          IEM#J
   ISN 0041      170 CONTINUE
   ISN 0042          EMF%I,1<#RR%IEM<
   ISN 0043          EMF%I,2<#RI%IEM<
   ISN 0044          IF%DABS%RI%IEM<<-1.D-8< 150,150,180
   ISN 0045      180 I#I&1
   ISN 0046          EMF%I,1<#RR%IEM<
   ISN 0047          EMF%I,2<#-RI%IEM<
   ISN 0048          GO TO 150
   ISN 0049      190 RETURN
   ISN 0050          END
```

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                          DATE  71.119/00.36.

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                           SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002          SUBROUTINE ASSIGN%CX,EMS,EMF,ALO,ICPLX,NS,NFF,MD,M2D1<
                C
                C     FORMS VECTOR  CX  FROM  EMS  AND EMF.
                C
  ISN 0003          REAL*8 CX%M2D1<,EMS%MD,2<,EMF%MD,2<,ALO,DABS
  ISN 0004          DIMENSION ICPLX%M2D1<
  ISN 0005          IA#NS&NFF
  ISN 0006          L#0
  ISN 0007      455 L#L&1
  ISN 0008          IF%L .GT. NS< GO TO 470
  ISN 0010          CX%L<#EMS%L,1<
  ISN 0011          ICPLX%L<#0
  ISN 0012          IF%DABS%EMS%L,2<< -1.D-8< 465,465,460
  ISN 0013      460 L1#L&1
  ISN 0014          CX%L1<#EMS%L,2<
  ISN 0015          ICPLX%L<#1
  ISN 0016          L#L1
  ISN 0017      465 GO TO 455
  ISN 0018      470 IF%NFF .EQ. 0< GO TO 490
  ISN 0020          L#L-1
  ISN 0021      475 L#L&1
  ISN 0022          IF%L .GT. IA< GO TO 490
  ISN 0024          LS#L-NS
  ISN 0025          CX%L<#EMF%LS,1<
  ISN 0026          ICPLX%L<#0
  ISN 0027          IF%DABS%EMF%LS,2<< -1.D-8< 485,485,480
  ISN 0028      480 L1#L&1
  ISN 0029          LS1#LS&1
  ISN 0030          CX%L1<#EMF%LS1,2<
  ISN 0031          ICPLX%L<#1
  ISN 0032          L#L1
  ISN 0033      485 GO TO 475
  ISN 0034      490 CX%L<#ALO
  ISN 0035          RETURN
  ISN 0036          END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                           SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
 ISN 0002          SUBROUTINE CONDN%N,CX,CF,CG,KOUNT,NJUMP,M2D1<
              C
              C    ROUTINE COMPUTES FUNCTION AND GRADIENT FOR ROUTINE DFMFP.
              C
 ISN 0003          REAL*8 CX%M2D1<,CG%M2D1<,CF,DABS
 ISN 0004          REAL*8 AL1,R2,RIM,DR,GA
 ISN 0005          REAL*8 ROOT1,ALO,SUMO
 ISN 0006          REAL*8 A%6,6<,B%6,6<,C%6,6<,AH%12,12<,A2%12,12<,A2S%12,12<,AV%144<
 ISN 0007          REAL*8 FF%6,6<,FG%6,6<,FH%6,6<,FJ%6,6<,GF%6,6<,GG%6,6<,GH%6,6<
 ISN 0008          REAL*8 GJ%6,6<,GFM%6,6<,GGM%6,6<,GHM%6,6<,GJM%6,6<,GRF%6,6<
 ISN 0009          REAL*8 GRG%6,6<,GRH%6,6<,GRJ%6,6<,CVR%12,12<,CVI%12,12<,RR%12<
 ISN 0010          REAL*8 RI%12<,VR%12<,VI%12<,XR%12<,XI%12<,VRN%12,1<,VIN%12,1<
 ISN 0011          REAL*8 W%12,4<,EM%12,2<
 ISN 0012          REAL*8 EMS%6,2<,EMF%6,2<
 ISN 0013          REAL*8 AVF%36<,XRS%6<,XRF%6<,XIS%6<,XIF%6<,VRS%6<,VRF%6<,VIS%6<
 ISN 0014          REAL*8 VIF%6<,RRS%6<,RRF%6<,RIS%6<,RIF%6<,VRNS%6,1<,VRNF%6,1<
 ISN 0015          REAL*8 VINS%6,1<,VINF%6,1<,AVS%36<,WS%6,4<,WF%6,4<
 ISN 0016          REAL*8 SV%12<,SVR%12<
 ISN 0017          DIMENSION IANA%12<,IANAS%6<,IANAF%6<,IROW%12,2<,IROWS%6,2<
 ISN 0018          DIMENSION IROWF%6,2<
 ISN 0019          DIMENSION ICPLX(13)
 ISN 0020          EQUIVALENCE %AV%1<,AVF%1<<,%AV%37<,AVS%1<<
 ISN 0021          EQUIVALENCE %EM%1<,EMF%1<<,%EM%13<,EMS%1<<
 ISN 0022          EQUIVALENCE %XR%1<,XRS%1<<,%XR%7<,XRF%1<<,%XI%1<,XIS%1<<
 ISN 0023          EQUIVALENCE %XI%7<,XIF%1<<,%VR%1<,VRS%1<<,%VR%7<,VRF%1<<
 ISN 0024          EQUIVALENCE %VI%1<,VIS%1<<,%VI%7<,VIF%1<<,%RR%1<,RRS%1<<
 ISN 0025          EQUIVALENCE %RR%7<,RRF%1<<,%RI%1<,RIS%1<<,%RI%7<,RIF%1<<
 ISN 0026          EQUIVALENCE %VRN%1<,VRNS%1<<,%VRN%7<,VRNF%1<<
 ISN 0027          EQUIVALENCE %VIN%1<,VINS%1<<,%VIN%7<,VINF%1<<
 ISN 0028          EQUIVALENCE %W%1<,WF%1<<,%W%25<,WS%1<<
 ISN 0029          EQUIVALENCE %IANA%1<,IANAS%1<<,%IANA%7<,IANAF%1<<
 ISN 0030          EQUIVALENCE %IROW%1<,IROWS%1<<,%IROW%13<,IROWF%1<<
 ISN 0031          COMMON /MCM/ AH,A2,A2S,AV,CVR,CVI,W,A,B,C,FF,FG,FH,FJ,GF,GG,GH,
                 1              GJ,GFM,GGM,GHM,GJM,GRF,GRG,GRH,GRJ,EM,VRN,VIN,RR,RI,
                 2              VR,VI,XR,XI,IROW,IANA
 ISN 0032          COMMON /MC/ SV,SVR,ROOT1,ALO,SUMO,DANORM,ACC,NS,NC,NF,NFF,MD,MD2
 ISN 0033          COMMON /MC2/ DR,GA,ICPLX,IAREA,M2,M2D
 ISN 0034       10 FORMAT%//,@  ITERATION NUMBER  KOUNT#@,I3<
              C 15 FORMAT%6D20.8<
              C 20 FORMAT%//,@  EIGENVALUES OF THE COMPENSATED SYSTEM.@<
              C 25 FORMAT%//,@  EIGENVALUES OF THE COMPENSATOR.@<
 ISN 0035       30 FORMAT%//,@  ALO #@,F12.5,@  ROOT1 #@,D15.7,@  COND.-NUMBER #@,
                 3D15.7,@  FUNCTION VALUE #@,F10.4<
 ISN 0036          IF%NJUMP .EQ. 1< GO TO 125
 ISN 0038          KO=0
 ISN 0039          NJUMP#1
 ISN 0040          IA#NS&NFF
 ISN 0041          IA1#NS&NFF
 ISN 0042          GO TO 235
 ISN 0043      125 CONTINUE
 ISN 0044          NAS#0
 ISN 0045          IA#NS&NFF
 ISN 0046          IA1#IA&1
 ISN 0047          L#0
 ISN 0048      130 L#L&1
 ISN 0049          IF%L .GT. NS< GO TO 145
 ISN 0051          EMS%L,1<#CX%L<
```

```
ISN 0052           EMS%L,2<#0.DO
ISN 0053           IF%ICPLX%L<-1< 140,135,135
ISN 0054       135 L1#L&1
ISN 0055           EMS%L,2<#CX%L1<
ISN 0056           EMS%L1,2<#-CX%L1<
ISN 0057           EMS%L1,1<#CX%L<
ISN 0058           L#L1
ISN 0059       140 GO TC 130
ISN 0060       145 IF%NFF .EQ. 0< GO TO 165
ISN 0062           L#L-1
ISN 0063       150 L#L&1
ISN 0064           IF%L .GT. IA< GO TO 165
ISN 0066           LS#L-NS
ISN 0067           EMF%LS,1<#CX%L<
ISN 0068           EMF%LS,2<#0.DO
ISN 0069           IF%ICPLX%L<-1< 160,155,155
ISN 0070       155 L1#L&1
ISN 0071           LS1#LS&1
ISN 0072           EMF%LS,2<#CX%L1<
ISN 0073           EMF%LS1,2<#-CX%L1<
ISN 0074           EMF%LS1,1<#CX%L<
ISN 0075           L#L1
ISN 0076       160 GO TC 150
ISN 0077       165 ALO#CX%L<
ISN 0078           IF%NAS .EQ. 1< GO TO 235
ISN 0080           K0#0
ISN 0081           K1#1
ISN 0082           K2#NFF
ISN 0083           K3#1
ISN 0084           K4#NS
ISN 0085       195 CONTINUE
ISN 0086           CALL MCOMPENS,NC,NF,NFF,MD,MD2,1,K1,K2,K3,K4<
ISN 0087       197 DO 200 I#1,NFF
ISN 0088           DO 200 J#1,NF
ISN 0089       200 FG%I,J<#FG%I,J<*ALO
ISN 0090           DO 205 I#1,NC
ISN 0091           DO 205 J=1,NFF
ISN 0092       205 FH%I,J<#FH%I,J</ALO
ISN 0093           IF%K0 .GT. 0< GO TO 220
ISN 0095           WRITE%3,10< KOUNT
             C     WRITE%3,20<
             C     DO 210 I#1,NS
             C 210 WRITE%3,15< %EMS%I,J<,J#1,2<
             C     IF%NFF .LT. 1< GO TO 220
             C     WRITE%3,25<
             C     DO 215 I#1,NFF
             C 215 WRITE%3,15< %EMF%I,J<,J#1,2<
ISN 0096       220 CONTINUE
ISN 0097           IWRITE#0
ISN 0098           CALL STAB%A,B,C,FF,FG,FH,FJ,AH,AV,RR,RI,NS,NC,NF,NFF,MD,M2D,M2,
                 S           IANA,IWRITE<
ISN 0099           CALL MMULT%AH,AH,A2,IA,IA,IA,M2D<
ISN 0100           IVC#3
ISN 0101           CALL SIMTR2%AH,A2,A2S,CVR,CVI,W,IROW,RR,RI,XR,XI,SV,SVR,IA,M2D,IVC
                 S                          <
ISN 0102           DO 225 I#1,IA
ISN 0103           XR%I<#0.DO
ISN 0104           DO 225 J#1,IA
```

```
ISN 0105       225 XR%I<#XR%I<&DABS%CVR%I,J<<
ISN 0106           SUMO#XR%I<
ISN 0107           DO 230 I#1,IA
ISN 0108       230 IF%SUMO .LT. XR%I<<  SUMO#XR%I<
ISN 0110           IF%NAS .EQ. 2< GO TO 235
ISN 0112           CALL SORT%EMS,EMF,RR,RI,SV,NS,NFF,MD,M2D<
ISN 0113           CALL ASSIGN%CX,EMS,EMF,ALO,ICPLX,NS,NFF,MD,M2D1<
ISN 0114           NAS#1
ISN 0115           L#0
ISN 0116           GO TC 130
ISN 0117       235 CONTINUE
ISN 0118           ROOT1#RR%1<
ISN 0119           DO 500 I#1,IA
ISN 0120       500 IF%ROOT1 .LT. RR%I<< ROOT1#RR%I<
ISN 0122           IF%IAREA .EQ. 0< GO TO 530
ISN 0124           R2#RR%1<
ISN 0125           RIM#RI%1<
ISN 0126           DO 520 I#1,IA
ISN 0127           IF%R2 .GT. RR%I<< R2#RR%I<
ISN 0129       520 IF%RIM .LT. RI%I<< RIM#RI%I<
ISN 0131       530 CONTINUE
ISN 0132           IF%KO .GT. 0< GO TO 240
             C
             C     COMPUTATION OF FUNCTION  CF .
             C
ISN 0134           CF=1.D0+(-ACC+ROOT1)/(SUMO*DANORM-ACC)
ISN 0135           IF%IAREA< 237,237,236
ISN 0136       236 CF#CF&GA*%%R2-ROOT1<*%R2-ROOT1<&RIM*RIM</%DR*DR<
ISN 0137       237 CONTINUE
ISN 0138           WRITE%3,30< ALO,ROOT1,SUMO,CF
ISN 0139           N2#0
ISN 0140           K1#0
ISN 0141           K2#0
ISN 0142           LF#0
ISN 0143           NAS#2
ISN 0144           GO TC 245
             C
             C     COMPUTATION OF SYNTHETIC GRADIENT  CG .
             C
ISN 0145       240 CG(KO)=1.D0+(-ACC+ROOT1)/(SUMO*DANORM-ACC)
ISN 0146           IF%IAREA< 242,242,241
ISN 0147       241 CG%KO<#CG%KO<&GA*%%R2-ROOT1<*%R2-ROOT1<&RIM*RIM</%DR*DR<
ISN 0148       242 CONTINUE
ISN 0149           CG%KO<#1.D2*%CG%KO<-CF<
ISN 0150           IF%LF-1< 245,330,410
ISN 0151       245 KO#KO&1
ISN 0152           IF%KO .GT. NS< GO TO 305
ISN 0154           IF%N2-1< 250,280,280
ISN 0155       250 IF%DABS%EMS%KO,2<<-1.D-8< 255,255,275
             C
             C     SELECTION OF A REAL SYSTEM EIGENVALUE FOR GRADIENT COMPUTATION.
             C
ISN 0156       255 N2#0
ISN 0157           EMS%KO,1<#EMS%KO,1<&1.D-2
ISN 0158           IF%KO .EQ. 1< GO TO 270
ISN 0160           KO1#KO-1
ISN 0161           KO2#KO-2
ISN 0162           IF%DABS%EMS%KO1,2<<-1.D-8< 260,260,265
```

```
ISN 0163      260 EMS%KO1,1<#EMS%KO1,1<-1.D-2
ISN 0164          GO TC 270
ISN 0165      265 EMS%KO2,2<#EMS%KO2,2<-1.D-2
ISN 0166      270 K3#KC
ISN 0167          K4#KO
ISN 0168          GO TC 195
            C
            C     SELECTION OF A COMPLEX SYSTEM EIGENVALUE FOR GRADIENT COMPUTATION.
            C
ISN 0169      275 N2#2
ISN 0170      280 N2#N2-1
ISN 0171          IF%N2 .EQ. 0< GO TO 300
ISN 0173          EMS%KO,1<#EMS%KO,1<&1.D-2
ISN 0174          IF%KC .EQ. 1< GO TO 295
ISN 0176          KO1#KO-1
ISN 0177          KO2#KO-2
ISN 0178          IF%DABS%EMS%KO1,2<<-1.D-8< 285,285,290
ISN 0179      285 EMS%KO1,1<#EMS%KO1,1<-1.D-2
ISN 0180          GO TC 295
ISN 0181      290 EMS%KO2,2<#EMS%KO2,2<-1.D-2
ISN 0182      295 K3#KO
ISN 0183          K4#KO
ISN 0184          GO TC 195
ISN 0185      300 KO1#KO-1
ISN 0186          EMS%KO1,2<#EMS%KO1,2<&1.D-2
ISN 0187          EMS%KO1,1<#EMS%KO1,1<-1.D-2
ISN 0188          GO TC 195
ISN 0189      305 LF#1
ISN 0190          KO#KO-1
ISN 0191          KO1#KO-1
ISN 0192          IF%DABS%EMS%KO,2<<-1.D-8< 310,310,315
ISN 0193      310 EMS%KO,1<#EMS%KO,1<-1.D-2
ISN 0194          GO TC 320
ISN 0195      315 EMS%KO1,2<#EMS%KO1,2<-1.D-2
ISN 0196      320 IF%NFF .EQ. 0< GO TO 405
ISN 0198      325 N2#0
ISN 0199          K3#0
ISN 0200          K4#0
ISN 0201      330 KO#KC&1
ISN 0202          KF#KC-NS
ISN 0203          IF%KC .GT. IA< GO TO 390
ISN 0205          IF%N2-1< 335,355,365
ISN 0206      335 IF(DABS(EMF(KF,2))-1.D-8) 340,340,360
            C
            C     SELECTION OF A REAL COMP. EIGENVALUE FOR GRADIENT COMPUTATION.
            C
ISN 0207      340 N2#0
ISN 0208          EMF%KF,1<#EMF%KF,1<&1.D-2
ISN 0209          IF%KF .EQ. 1< GO TO 355
ISN 0211          KF1#KF-1
ISN 0212          KF2#KF-2
ISN 0213          IF%DABS%EMF%KF1,2<<-1.D-8< 345,345,350
ISN 0214      345 EMF%KF1,1<#EMF%KF1,1<-1.D-2
ISN 0215          GO TO 355
ISN 0216      350 EMF%KF2,2<#EMF%KF2,2<-1.D-2
ISN 0217      355 K1#KF
ISN 0218          K2#KF
ISN 0219          GO TC 195
```

```
               C
               C     SELECTION OF A COMPLEX COMP. EIGENVALUE FOR GRADIENT COMPUTATION.
               C
ISN 0220         360 N2#2
ISN 0221         365 N2#N2-1
ISN 0222             IF%N2 .EC. 0< GO TO 385
ISN 0224             EMF%KF,1<#EMF%KF,1<&1.D-2
ISN 0225             IF%KF .EC. 1< GO TO 380
ISN 0227             KF1#KF-1
ISN 0228             KF2#KF-2
ISN 0229             IF%DABS%EMF%KF1,2<<-1.D-8< 370,370,375
ISN 0230         370 EMF%KF1,1<#EMF%KF1,1<-1.D-2
ISN 0231             GO TC 380
ISN 0232         375 EMF%KF2,2<#EMF%KF2,2<-1.D-2
ISN 0233         380 K1#KF
ISN 0234             K2#KF
ISN 0235             GO TC 195
ISN 0236         385 KF1#KF-1
ISN 0237             EMF%KF1,2<#EMF%KF1,2<&1.D-2
ISN 0238             EMF%KF1,1<#EMF%KF1,1<-1.D-2
ISN 0239             GO TC 195
ISN 0240         390 LF#2
ISN 0241             KO#KO-1
ISN 0242             KF#KO-NS
ISN 0243             KF1#KF-1
ISN 0244             IF%DABS%EMF%KF,2<<-1.D-8< 395,395,400
ISN 0245         395 EMF%KF,1<#EMF%KF,1<-1.D-2
ISN 0246             GO TC 405
ISN 0247         400 EMF%KF1,2<#EMF%KF1,2<-1.D-2
ISN 0248         405 KO#KC&1
ISN 0249             AL1#ALO
ISN 0250             ALO#%ALO&1.D-2</ALO
ISN 0251             GO TC 197
ISN 0252         410 ALO#AL1
ISN 0253             RETURN
ISN 0254             END
```

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002                     SUBROUTINE LINEQS(IOP,N,MB,AA,BB,X,A,B,SV,SVR,IER,D,TOL,MD,ND)
                        C
                        C
                        C      LINEAR MATRIX EQUATION SOLVER. USES GAUSSIAN ELIMINATION WITH
                        C      FULL PIVOTAL CONDENSATION TO SOLVE  AA*X =BB FOR X, WHERE AA IS
                        C      (N * N), BB AND X ARE (N * MB)
                        C
                        C      IOP - OPERATION CODE
                        C          IOP = 1 - STANDARD SOLUTION - INPUTS AA, BB, SOLUTION IN X.
                        C              = 2 - MATRIX INVERSION - INPUT AA, SOLUTION X = AA(-1).
                        C                    BB NOT USED.
                        C              = 3 - NEW RIGHT HAND SIDE (BB) FOR EQUATIONS PREVIOUSLY
                        C                    SOLVED WITH SAME AA MATRIX. INPUT BB AND A, SV,
                        C                    SVR, FROM PREVIOUS RETURN. SOLUTION IN X.
                        C              = 4 - UPPER TRIANGULAR MATRIX INVERSION (NO REDUCTION).
                        C                    INPUT AA (MATRIX TO BE INVERTED), OUTPUT IS
                        C                    X = AA(-1), BB,SV,SVR NOT USED, AA UNCHANGED.
                        C
                        C      MD AND ND DEFINE SIZE OF ARRAYS IN PARAMETER LIST AS INDICATED BY
                        C          DIMENSION STATEMENT.
                        C      STORAGE - A,B,SV,SVR ARE STORAGE ARRAYS OF INDICATED DIMENSIONS.
                        C      AA AND BB ARE UNCHANGED BY SUBROUTINE.
                        C      D - DETERMINANT OF AA.
                        C      IER - ERROR CODE.
                        C          IER = 0 - SUCCESFUL SOLUTION.
                        C              = -1 - N IS .LE. 0 .
                        C              = K .GT. 0 - AA IS SINGULAR OF RANK (K-1).
                        C      AA MATRIX IS CONSIDERED TO BE SINGULAR IF A PIVOT LESS THAN
                        C          TOL*ABS(AAMAX) IS FOUND DURING THE ELIMINATION PROCESS.
                        C          AAMAX IS THE ELEMENT OF LARGEST MAGNITUDE IN THE AA MATRIX.
                        C      N SHOULD NOT EXCEED 100 WITHOUT INCREASING THE SIZE OF THE 'BUF'
                        C          ARRAY.
                        C      IN THE CALL TO LINEQS, THE ONLY MATRICES IN THE SET (AA,BB,X,A,B)
                        C          WHICH MUST BE DIFFERENT ARE A AND B. THAT IS, AA AND A, BB AND
                        C          B MAY BE THE SAME IF THERE IS NO DESIRE TO SAVE AA AND BB.
                        C          ALSO, X CAN BE THE SAME MATRIX AS EITHER A OR B, BUT IF X AND
                        C          A ARE COMMON, A SUBSEQUENT CALL TO LINEQS WITH A NEW BB MATRIX
                        C          (I.E., IOP=3) CANNOT BE MADE.
                        C
                        C
ISN 0003                     DIMENSION AA(MD,MD),BB(MD,ND),X(MD,ND),A(MD,MD),B(MD,ND),SV(MD),
                           1           SVR(MD)
                        C
ISN 0004                     DOUBLE PRECISION AA,BB,X,A,B,PIVOT,R,DF
ISN 0005                     DOUBLE PRECISION BUF(100)
ISN 0006                     DOUBLE PRECISION DABS
                        C
                        C
ISN 0007                     EPS=0.
ISN 0008                     N1=N-1
ISN 0009                     IB=IOP-2
ISN 0010                     IF(IB) 70,50,40
                        C
                        C      INVERSION - SET MB=N AND B=I
                        C
ISN 0011                  40 IF(IB-1) 70,70,50
ISN 0012                  50 MB=N
```

```
ISN 0013          DO 60 I=1,N
ISN 0014          DO 55 J=1,N
ISN 0015       55 B(J,I)=0.D0
ISN 0016       60 B(I,I)=1.D0
ISN 0017          IF(IB) 60,80,62
            C
            C     UPPER TRIANGULAR MATRIX INVERSION (NO ELIMINATION).
            C
ISN 0018       62 D=1.
ISN 0019          DO 64 I=1,N
ISN 0020          DO 63 J=1,N
ISN 0021       63 A(J,I)=AA(J,I)
ISN 0022       64 D=D*A(I,I)
ISN 0023          IF(ABS(D)-TOL) 66,66,200
ISN 0024       66 IER=1
ISN 0025          RETURN
ISN 0026       70 DO 75 I=1,MB
ISN 0027          DO 75 J=1,N
ISN 0028       75 B(J,I)=BB(J,I)
ISN 0029          IF(IB) 80,80,100
ISN 0030       80 DO 90 I=1,N
ISN 0031          DO 85 J=1,N
ISN 0032       85 A(J,I)=AA(J,I)
ISN 0033       90 SV(I)=I
ISN 0034          D=1.
ISN 0035      100 IF(N1) 101,150,102
ISN 0036      101 IER=-1
ISN 0037          RETURN
            C
            C
            C     ELIMINATION LOOP (THROUGH STATEMENT 126) .
            C
            C     SEARCH FOR LARGEST ELEMENT IN LOWER (NE * NE) BLOCK OF A (= PIVOT)
            C
ISN 0038      102 DO 126 NE=1,N1
ISN 0039          IF(IB) 103,103,110
ISN 0040      103 BF=DABS(A(NE,NE))
ISN 0041          PIVOT=A(NE,NE)
ISN 0042          NR=NE
ISN 0043          NC=NE
ISN 0044          DO 106 J=NE,N
ISN 0045          DO 106 I=NE,N
ISN 0046          IF(DABS(A(I,J))-BF) 106,106,104
ISN 0047      104 NR=I
ISN 0048          NC=J
ISN 0049          BF=DABS(A(I,J))
ISN 0050          PIVOT=A(I,J)
ISN 0051      106 CONTINUE
ISN 0052          IF(NE .EQ. 1) EPS=TOL*DABS(PIVOT)
ISN 0054          D=D*PIVOT
ISN 0055          SVR(NE)=NR
            C
            C     SINGULARITY CHECK
            C
ISN 0056          IF(DABS(PIVOT) - EPS) 108,108,110
ISN 0057      108 IER=NE
ISN 0058          D=0.
ISN 0059          RETURN
```

```
ISN 0060       110 NR=SVR(NE)
ISN 0061           IF(NR-NE) 117,117,111
             C
             C     ROW INTERCHANGE - A(NR,K) WITH A(NE,K) FOR K = NE TO N
             C                     - B(NR,K) WITH B(NE,K) FOR K = 1 TO MB
             C
ISN 0062       111 IF(IE) 112,112,115
ISN 0063       112 DO 114 K=NE,N
ISN 0064           BF=A(NR,K)
ISN 0065           A(NR,K)=A(NE,K)
ISN 0066       114 A(NE,K)=BF
ISN 0067       115 DO 116 K=1,MB
ISN 0068           BF=B(NR,K)
ISN 0069           B(NR,K)=B(NE,K)
ISN 0070       116 B(NE,K)=BF
ISN 0071       117 IF(IP) 1171,1171,122
ISN 0072      1171 IF(NC-NE) 122,122,118
             C
             C     COLUMN INTERCHANGE - A(K,NE) WITH A(K,NC) FOR K = 1 TO N
             C        NOTE - SV(I) IS THE ORIGINAL UNKNOWN VARIABLE NO. (I.E., COLUMN
             C        NUMBER) NOW OCCUPYING COLUMN I IN THE REDUCED ARRAY.
             C
ISN 0073       118 BF=SV(NC)
ISN 0074           SV(NC)=SV(NE)
ISN 0075           SV(NE)=BF
ISN 0076           DO 120 K=1,N
ISN 0077           BF=A(K,NC)
ISN 0078           A(K,NC)=A(K,NE)
ISN 0079       120 A(K,NE)=BF
             C
             C     REDUCTION LOOP - R = A(I,NE)/PIVOT = A(I,NE)/A(NE,NE)
             C                      A(I,J) = A(I,J) - R*A(NE,J) FOR I,J = NE1 TO N
             C                      B(I,J) = B(I,J) - R*B(NE,J) FOR I=NE1,N J=1,MB
             C        R IS STORED IN A(I,NE) (LOWER PART OF A MATRIX) FOR SUBSEQUENT
             C        CALLS WITH A NEW RIGHT HAND SIDE (BB MATRIX) - IOP = 3.
             C
ISN 0080       122 NE1=NE+1
ISN 0081           DO 126 I=NE1,N
ISN 0082           IF(IP) 1231,1231,123
ISN 0083       123 R=A(I,NE)
ISN 0084           GO TO 125
ISN 0085      1231 R=A(I,NE)/PIVOT
ISN 0086           A(I,NE)=R
ISN 0087           DO 124 J=NE1,N
ISN 0088       124 A(I,J)=A(I,J)-R*A(NE,J)
ISN 0089       125 DO 126 J=1,MB
ISN 0090       126 B(I,J)=B(I,J)-R*B(NE,J)
             C
             C     END OF ELIMINATION LOOP
             C
             C
             C     FINAL SINGULARITY CHECK
             C
ISN 0091       150 IF(DABS(A(N,N))-EPS) 152,152,170
ISN 0092       152 IER=N
ISN 0093           D=0.
ISN 0094           RETURN
ISN 0095       170 IF(IF .GT. 0) GO TO 200
```

```
ISN 0097            D=D*A(N,N)
               C
               C      BACK SUBSTITUTION AND SOLUTION
               C         X(I,K) = (B(I,K) - SUM(J=I+1 TO N) (A(I,J)*X(J,K)))/A(I,I)
               C         FOR I = N TO 1, AND EACH COLUMN OF B (K=1 TO MB).
               C
ISN 0098        200 DO 210 K=1,MB
ISN 0099            X(N,K)=B(N,K)/A(N,N)
ISN 0100            I=N
ISN 0101        202 I=I-1
ISN 0102            IF(I) 210,210,204
ISN 0103        204 M1=I+1
ISN 0104            BF=0.
ISN 0105            DO 206 J=M1,N
ISN 0106        206 BF=BF+A(I,J)*X(J,K)
ISN 0107            X(I,K)=(B(I,K)-BF)/A(I,I)
ISN 0108            GO TO 202
ISN 0109        210 CONTINUE
ISN 0110            IER=0
ISN 0111            IF(IP-2) 211,220,211
               C
               C      ROW EXCHANGE - PUT X VARIABLES INTO PROPER PLACE IN X MATRIX
               C         X(M,J) = X(I,J) WHERE M=SV(I), FOR J=1 TO MB .
               C
ISN 0112        211 DO 214 J=1,MB
ISN 0113            DO 212 I=1,N
ISN 0114        212 BUF(I)=X(I,J)
ISN 0115            DO 214 I=1,N
ISN 0116            K=SV(I)
ISN 0117        214 X(K,J)=BUF(I)
               C
ISN 0118        220 RETURN
ISN 0119            END
```

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                          DATE  71.096/22.27.1

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002               SUBROUTINE ARBEIG(A,B,EM,G,T,A1,A2,B1,X,SV,SVR,NS,NC,K1,K2,IER,MD,
                        1IMULT)
                   C
                   C
                   C     ARBITRARY PLACEMENT OF EIGENVALUES OF THE MATRIX  (A - B*G)
                   C         GAIN MATRIX G WILL ALWAYS BE REAL-VALUED.
                   C         PROGRAM HANDLES BOTH, DISTINCT AND/OR MULTIPLE EIGENVALUES.
                   C
                   C     IER = 0 - SUCCESSFUL, OTHERWISE( (A,B) IS UNCONTROLLABLE.
                   C
                   C     A - (NS * NS)  SYSTEM MATRIX
                   C     B - (NS * NC)  INPUT MATRIX
                   C     G = - C*T(-1)   - (NC *NS)  GAIN MATRIX
                   C     EM - (NS * 2)  MATRIX OF COMPLEX EIGENVALUES.  RE(EM) IN COL. 1,
                   C         IM(EM) IN COL. 2 , IF EM AND A HAVE COMMON EIGENVALUES,
                   C         EM(I,1) =EM(I,1) - 1 .
                   C         BOTH CONJUGATE COMPLEX EIGENVALUES MUST BE PLACED IN SUCCESS-
                   C         IVE ROWS IN THE EM - MATRIX, ALWAYS LIST THE COMPLEX EIGENVA-
                   C         LUE WITH POS. IM. PART FIRST.
                   C         MULTIPLE EIGENVALUES NEED NOT BE INPUTED IN SUCCESSIVE ROWS
                   C         OF THE  EM - MATRIX.
                   C
                   C     MATRICES A AND B ARE UNCHANGED BY THE SUBROUTINE.
                   C
                   C
  ISN 0003               DIMENSION A(MD,MD),B(MD,MD),EM(MD,2),G(MD,MD),A1(MD,MD),A2(MD,MD),
                        1T(MD,MD),B1(MD,MD),X(MD,1),SV(MD),SVR(MD),IMULT(MD,2)
                   C
  ISN 0004               DOUBLE PRECISION A1,A2,B,EM,G,T,A,B1,DABS,X
                   C
  ISN 0005               NC1=NC&1
  ISN 0006               IF(NC1 .GT. NS) GO TO 130
  ISN 0008               L=0
  ISN 0009               DO 120 J=NC1,NS
  ISN 0010               L=L&1
  ISN 0011               DO 100 I=1,NS
  ISN 0012           100 B(I,J)=B(I,L)
  ISN 0013               IF(L-NC) 120,110,110
  ISN 0014           110 L=0
  ISN 0015           120 CONTINUE
  ISN 0016           130 IM=K1-1
  ISN 0017               DO 132 I=1,NS
  ISN 0018               IMULT(I,1)=1
  ISN 0019           132 IMULT(I,2)=0
  ISN 0020           140 IM=IM&1
                   C
                   C     CHECK FOR MULTIPLE EIGENVALUES.
                   C         IF (EM(I,1),EM(I,2)) AND (EM(J,1),EM(J,2)) ARE EQUAL,
                   C         IMULT(J,1)=I , IMULT(J,2)=1 .
                   C
  ISN 0021           141 IF(IM .EQ. 1) GO TO 146
  ISN 0023               IMN=IM-1
  ISN 0024               DO 144 I=1,IMN
  ISN 0025               IF(DABS(EM(I,1)-EM(IM,1)) .LT. .1D-7 .AND. DABS(EM(I,2)-EM(IM,2))
                        1.LT. .1D-7) GO TO 142
```

```
ISN 0027          GO TC 144
ISN 0028      142 IMULT(IM,1)=I
ISN 0029          IMULT(IM,2)=1
ISN 0030      144 CONTINUE
            C
ISN 0031      146 IF(DABS(EM(IM,2))-1.D-8) 150,150,190
            C
            C     REAL EIGENVALUES, DISTINCT AND/OR MULTIPLE.
            C
ISN 0032      150 IE=0
ISN 0033          CO 170 I=1,NS
ISN 0034          CO 16C J=1,NS
ISN 0035      160 A1(I,J)=-A(I,J)
ISN 0036          A1(I,I)=A1(I,I)&EM(IM,1)
ISN 0037      170 B1(I,1)=B(I,IM)
ISN 0038          IF(IMULT(IM,2) .EQ. 0) GO TO 174
ISN 0040          MULT=IMULT(IM,1)
ISN 0041          CO 172 I=1,NS
ISN 0042      172 B1(I,1)=B1(I,1)-T(MULT,I)
ISN 0043      174 CONTINUE
ISN 0044          CALL LINEQS(1,NS,1,A1,B1,X,A1,B1,SV,SVR,IER,D,1.E-10,MD,1)
ISN 0045          IF(IER) 180,280,180
ISN 0046      180 EM(IM,1)=EM(IM,1)-1.DO
ISN 0047          GO TC 141
            C
            C     COMPLEX PAIR OF EIGENVALUES, DISTINCT AND/OR MULTIPLE.
            C
ISN 0048      190 IE=1
ISN 0049          IM1=IM&1
ISN 0050          CO 21C J=1,NS
ISN 0051          CO 2CC I=1,NS
ISN 0052      200 A2(I,J)=-A(I,J)
ISN 0053      210 A2(J,J)=A2(J,J)&EM(IM,1)
ISN 0054          IF(IMULT(IM,2) .EC. 1) GO TO 232
ISN 0056          DO 23C I=1,NS
ISN 0057          B1(I,1)=C.DO
ISN 0058          DO 22C J=1,NS
ISN 0059          B1(I,1)=B1(I,1)&A2(I,J)*B(J,IM)
ISN 0060          A1(I,J)=0.DO
ISN 0061          CO 220 K=1,NS
ISN 0062      220 A1(I,J)=A1(I,J)&A2(I,K)*A2(K,J)
ISN 0063          B1(I,1)=B1(I,1)&EM(IM,2)*B(I,IM1)
ISN 0064      230 A1(I,I)=A1(I,I)&EM(IM,2)*EM(IM,2)
ISN 0065          GO TC 238
ISN 0066      232 MULT=IMULT(IM,1)
ISN 0067          MULT1=MULT&1
ISN 0068          CO 236 I=1,NS
ISN 0069          B1(I,1)=0.DC
ISN 0070          DO 234 J=1,NS
ISN 0071          B1(I,1)=B1(I,1)&A2(I,J)*(B(J,IM)-T(MULT,J))
ISN 0072          A1(I,J)=C.DO
ISN 0073          CU 234 K=1,NS
ISN 0074      234 A1(I,J)=A1(I,J)&A2(I,K)*A2(K,J)
ISN 0075          B1(I,1)=B1(I,1)&EM(IM,2)*(B(I,IM1)-T(MULT1,I))
ISN 0076      236 A1(I,I)=A1(I,I)&EM(IM,2)*EM(IM,2)
ISN 0077      238 CONTINUE
ISN 0078          CALL LINEQS(1,NS,1,A1,B1,X,A1,B1,SV,SVR,IER,D,1.E-10,MD,1)
ISN 0079          IF(IER) 240,250,240
```

```
ISN 0080       240 EM(IM,1)=EM(IM,1)-1.D0
ISN 0081           EM(IM1,1)=EM(IM,1)
ISN 0082           GO TO 141
             C
             C     CONSTRUCTION OF  T(TRANS)  MATRIX
             C
ISN 0083       250 DO 270 I=1,NS
ISN 0084           T(IM1,I)=-B(I,IM)
ISN 0085           DO 260 J=1,NS
ISN 0086       260 T(IM1,I)=T(IM1,I)&A2(I,J)*X(J,1)
ISN 0087       270 T(IM1,I)=T(IM1,I)/EM(IM,2)
ISN 0088           IF(IMULT(IM,2) .EQ. 0) GO TO 280
ISN 0090           DO 272 I=1,NS
ISN 0091       272 T(IM1,I)=T(IM1,I)&T(MULT,I)/EM(IM,2)
ISN 0092       280 DO 290 I=1,NS
ISN 0093       290 T(IM,I)=X(I,1)
ISN 0094           IM=IM&1E
ISN 0095           IF(IM-K2) 140,300,300
             C
             C     CALCULATION OF  G = - C*T(-1)
             C
ISN 0096       300 DO 310 I=1,NC
ISN 0097           DO 310 J=1,NS
ISN 0098       310 A2(J,I)=0.D0
ISN 0099           K=1
ISN 0100           DO 320 I=1,NS
ISN 0101           A2(I,K)=1.D0
ISN 0102           K=K&1
ISN 0103           IF(K .GT. NC) K=1
ISN 0105       320 CONTINUE
ISN 0106           CALL LINEQS(1,NS,NC,T,A2,A1,G,B1,SV,SVR,IER,D,1.E-50,MD,MD)
ISN 0107           IF(IER) 350,330,350
ISN 0108       330 DO 340 I=1,NC
ISN 0109           DO 340 J=1,NS
ISN 0110       340 G(I,J)=-A1(J,I)
ISN 0111       350 RETURN
ISN 0112           END
```

APPENDIX B

Listing of the Pole Placement Computer Program
(including a subroutine to solve the matrix Riccati equation)

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                           SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
              C
              C      PROGRAM DETERMINES ALL-STATE FEEDBACK GAINS FOR ARBITRARY POLE-
              C          PLACEMENT.THE RESULTING GAINS ARE TAKEN TO INITIALIZE THE
              C          KLEINMAN ITERAIVE SCHEME FOR THE SOLUTION OF THE ALGEBRAIC
              C          MATRIX RICCATI EQUATION %..IMPORTANT.. FOR THE SOLUTION OF
              C          THE RICCATI EQUATION IT IS REQUIRED, THAT THE ARBITRARY POLES
              C          LIE IN THE LEFT HALF OF THE COMPLEX S-PLANE. ..<.
              C
              C      THE FOLLOWING PARAMETERS HAVE TO BE READ IN
              C          1.  NS,NC,NRICC,NCHECK             FORMAT%8I10<
              C               NS - ORDER OF SYSTEM MATRIX A
              C               NC - NUMBER OF SYSTEM INPUTS
              C               NRICC- #0 RICCATI EQUATICN WILL NOT BE SOLVED
              C                      #1 RICCATI EQUTTICN WILL BE SOLVED
              C               NCHECK - #0 NO CHECK OF COMPUTED GAINS, NO BACK SUBSTI-
              C                                  TUTION  OF COMPUTED RICCATI MATRIX.
              C                        #1 CHECK OF COMPUTEC GAINS AND BACK SUBSTITUTION
              C                                  OF COMPUTED RICCATI MATRIX.
              C          2.  A - %NS*NS< SYSTEM MATRIX, INPUT BY ROWS.
              C          3.  B - %NS*NC< INPUT MATRIX, INPUT BY ROWS.
              C          4.  EM - %NS*2< MATRIX OF DESIRED POLE LOCATION.
              C          5.  IFF NRICC#1  Q - %NS*NS< STATE WEIGHTING MATRIX
              C          6.  IFF NRICC#1  R - %NC*NC< CONTROL WEIGHTING MATRIX
              C          MATRIX INPUT FORMAT%5F15.6<
              C
              C      PROBLEMS CAN BE STACKED ONE AFTER THE OTHER. LAST CARD HAS TO BE
              C          BLANK.
              C
              C      PROGRAM DIMENSIONED FOR SYSTEMS UP TO ORDER 6 .
              C
  ISN 0002           DIMENSION A%6,6<,B%6,6<,EM%6,2<,G%6,6<,A1%6,6<,A2%6,6<,T%6,6<,
                    1B1%6,6<,X%6,1<,SV%6<,SVR%6<,Y%6,1<
  ISN 0003           REAL*8 AV%36<,AD%6,6<,SINV%6,6<,W%6,4<,XR%6<,XI%6<,VR%6<,VI%6<
  ISN 0004           REAL*8 AR%6<,RI%6<,B2%6,6<
  ISN 0005           REAL*8 R1(6,6),Q(6,6),R(6,6),TOL
  ISN 0006           REAL*8 Z%6,6<,ZG%6,6<
  ISN 0007           DIMENSION IANA%6<,IROW%6,2<
  ISN 0008           DOUBLE PRECISION DABS
  ISN 0009           DOUBLE PRECISION A,B,EM,G,A1,A2,T,B1,X,SV,SVR,Y
  ISN 0010         5 FORMAT%8I10<
  ISN 0011         7 FORMAT%6D20.8<
  ISN 0012        10 FORMAT%5F15.6<
  ISN 0013        15 FORMAT%4F20.8<
  ISN 0014        20 FORMAT%1H1,@  STATES # @,I3,4X,@  INPUTS # @,I3<
  ISN 0015        25 FORMAT%//,@  A - SYSTEM MATRIX@<
  ISN 0016        30 FORMAT%//,@  B - INPUT MATRIX@<
  ISN 0017        35 FORMAT%//,@  DESIRED EIGENVALUES CF %A - B*G<@<
              C   40 FORMAT%//,@  OBTAINED EIGENVALUES OF %A - B*G<@<
  ISN 0018        45 FORMAT%//,@  IER .NE. 0, THUS, %A,B< SEEMS TO BE MARGINALLY@,/,@
                    FCONTROLLABLE ONLY.@<
  ISN 0019        50 FORMAT%//,@  G - GAIN MATRIX@<
  ISN 0020        55 FORMAT%//,@  T - SOLUTION MATRIX@<
  ISN 0021        60 FORMAT%//,@  THE PAIR %A,B< IS UNCONTROLLABLE, NCON # 0 .@<
  ISN 0022        65 FORMAT%//,@  COMPUTED EIGENVALUES OF %A - B*G<.@<
  ISN 0023        70 FORMAT%//,@  VECTOR ALPHA%TRANS< %VECTOR  D # B*ALPHA<.@<
  ISN 0024        75 FORMAT%//,@  VECTOR  D%TRANS<.@<
  ISN 0025        80 FORMAT%//,@  DIAGONALIZED MATRIX  A, OR DIAGONALIZED MATRIX %A-B*Z
```

```
                 1G< IN THE CASE@,/,@  OF MULTIPLE EIGENVALUES AND/CR  COMMON OPEN-
                 2AND CLUSED-LOOP EIGENVALUES.@<
ISN 0026        85 FORMAT(//,'  MATRIX  SINV.')
ISN 0027        90 FORMAT%//,@  MATRIX  Q .@<
ISN 0028        91 FORMAT%//,@  MATRIX  R .@<
ISN 0029        92 FORMAT%//,@  MATRIX  R%INVERSE< .@<
ISN 0030        93 FORMAT&//,@  RICCATI MATRIX  P .@<
ISN 0031        94 FORMAT%//,@  RESIDUAL MATRIX. MATRIX IS ZERO, IF MATRIX P IS ACCUR
                4ATE.@<
ISN 0032        95 FORMAT(//,'  MATRIX  SINV*B .')
ISN 0033        96 FORMAT%1H1<
             C
ISN 0034           MD#6
ISN 0035           MD2#MD*MD
ISN 0036        99 READ%1,5< NS,NC,NRICC,NCHECK
ISN 0037           IF%NS .EQ. 0< GO TO 160
ISN 0039           WRITE%3,20< NS,NC
ISN 0040           READ%1,1C< %%A%I,J<,J#1,NS<,I#1,NS<
ISN 0041           WRITE%3,25<
ISN 0042           DO 1CO I#1,NS
ISN 0043           WRITE%3,7< %A%I,J<,J#1,NS<
ISN 0044       100 CONTINUE
ISN 0045           READ%1,10< %%B%I,J<,J#1,NC<,I#1,NS<
ISN 0046           WRITE%3,30<
ISN 0047           DO 101 I#1,NS
ISN 0048           WRITE%3,7< %B%I,J<,J#1,NC<
ISN 0049       101 CONTINUE
ISN 0050           READ%1,15< %%EM%I,J<,J#1,2<,I#1,NS<
ISN 0051           WRITE%3,35<
ISN 0052           DO 102 I#1,NS
ISN 0053       102 WRITE%3,15< %EM%I,J<,J#1,2<
ISN 0054           DO 103 I#1,NS
ISN 0055           DO 103 J#1,NS
ISN 0056           ZG%I,J<#0.DO
ISN 0057       103 Z%I,J<#A%I,J<
             C
             C     INITIALIZATION OF ARBITRARY FEEDBACK GAIN MATRIX  ZG  %A#A-B*ZG<.
             C
ISN 0058           L#0
ISN 0059           DO 105 I#1,NS
ISN 0060           L#L&1
ISN 0061           ZG%L,I<#I
ISN 0062           IF%L-NC< 105,104,104
ISN 0063       104 L#0
ISN 0064       105 CONTINUE
ISN 0065           NM#0
ISN 0066           NAG#C
             C
             C    COMPUTATION OF THE EIGENVALUES OF MATRIX  A .
             C
ISN 0067       106 CONTINUE
ISN 0068           CALL MVECT%A,AV,NS,MD,MD2<
ISN 0069           CALL HSBG%NS,AV,NS,MD2<
ISN 0070           CALL ATEIG(NS,AV,RR,RI,IANA,NS,MD,MD2)
             C
             C     CHECK FOR MULTIPLE EIGENVALUES. MAXIMALLY 3 TRIALS TO OBTAIN
             C         CISTINCT EIGENVALUES.
             C
```

```
ISN 0071          IF%NP .NE. 0< GO TO 1002
ISN 0073          IF%NS .EQ. 1< GO TO 1001
ISN 0075          NS1#NS-1
ISN 0076          DO 108 I#1,NS1
ISN 0077          I1#I&1
ISN 0078          DO 108 J#I1,NS
ISN 0079          IF%DABS%RR%I<-RR%J<<-1.D-8< 107,107,108
ISN 0080      107 IF%DABS%RI%I<-RI%J<<-1.D-8< 109,109,108
ISN 0081      108 CONTINUE
ISN 0082          GO TC 1001
ISN 0083      109 NAG#NAG&1
ISN 0084          IF%NAG-3< 1092,1092,1091
ISN 0085     1091 NAG#NAG-1
ISN 0086          GO TC 1001
ISN 0087     1092 DO 11C I#1,NS
ISN 0088          DO 11C J#1,NS
ISN 0089          DO 11C K#1,NC
ISN 0090      110 A%I,J<#A%I,J<-B%I,K<*ZG%K,J<
ISN 0091          GO TC 106
C
C         CHECK FOR COMMON CLOSED-LOOP AND OPEN-LOOP POLES. MAXIMALLY 3
C             TRIALS TO ELIMINATE COMMON POLES.
C
ISN 0092     1001 CONTINUE
ISN 0093          NM#1
ISN 0094          IF%NCHECK .EQ. 0< GO TO 1009
ISN 0096          NG#NAG&3
ISN 0097     1002 CONTINUE
ISN 0098          DO 1C04 I#1,NS
ISN 0099          DO 1C04 J#1,NS
ISN 0100          IF%DABS%RR%I<-EM%J,1<<-1.D-8< 1003,1003,1004
ISN 0101     1003 IF%DABS%RI%I<-EM%J,2<<-1.D-8< 1005,1005,1004
ISN 0102     1004 CONTINUE
ISN 0103          GO TC 1009
ISN 0104     1005 NAG#NAG&1
ISN 0105          IF%NAG-NG< 1007,1007,1006
ISN 0106     1006 NAG#NAG-1
ISN 0107          GO TC 1009
ISN 0108     1007 DO 1C08 I#1,NS
ISN 0109          DO 1C08 J#1,NS
ISN 0110          DO 1C08 K#1,NC
ISN 0111     1008 A%I,J<#A%I,J<-B%I,K<*ZG%K,J<
ISN 0112          GO TC 106
ISN 0113     1009 CONTINUE
ISN 0114          CALL MMULT%A,A,A1,NS,NS,NS,MD<
ISN 0115          CALL SIMTR2%A,A1,A2,B2,SINV,W,IROW,RR,RI,XR,XI,VR,VI,NS,MD,2<
ISN 0116          CALL MMULT%SINV,B,A1,NS,NS,NC,MD<
ISN 0117          WRITE(3,85)
ISN 0118          DO 1101 I#1,NS
ISN 0119     1101 WRITE%3,7< %SINV%I,J<,J#1,NS<
ISN 0120          WRITE(3,95)
ISN 0121          DO 1102 I#1,NS
ISN 0122     1102 WRITE%3,7< %A1%I,J<,J#1,NC<
ISN 0123          CALL SINGLE%A1,XR,RI,XI,IROW,NCON,NS,NC,MD<
ISN 0124          IF%NCON< 111,111,112
ISN 0125      111 WRITE%3,60<
ISN 0126          STOP
ISN 0127      112 CONTINUE
```

```
ISN 0128          WRITE%3,70<
ISN 0129          WRITE%3,7< %XI%I<,I#1,NC<
ISN 0130          WRITE%3,75<
ISN 0131          WRITE%3,7< %XR%I<,I#1,NS<
ISN 0132          DO 114 I#1,NS
ISN 0133          DO 113 J#1,NS
ISN 0134      113 AD%I,J<#0.D0
ISN 0135      114 AD%I,I<#AR%I<
ISN 0136          I#0
ISN 0137      115 I#I&1
ISN 0138          IF%DABS%RI%I<<-1.D-8< 117,117,116
ISN 0139      116 IP#I&1
ISN 0140          AD%I,IP<#RI%I<
ISN 0141          AD%IP,I<#-RI%I<
ISN 0142          I#IP
ISN 0143      117 IF%I-NS< 115,118,118
ISN 0144      118 CONTINUE
ISN 0145          WRITE%3,80<
ISN 0146          DO 119 I=1,NS
ISN 0147      119 WRITE(3,7) (AD(I,J),J=1,NS)
ISN 0148          DO 124 I#1,NS
ISN 0149          DO 122 J#1,NS
ISN 0150      122 B2%I,J<#0.D0
ISN 0151      124 B2%I,1<#XR%I<
ISN 0152          CALL ARBEIG%AD,B2,EM,G,T,A1,A2,B1,X,SV,SVR,NS,1,1,NS,IER,MD,IROW<
ISN 0153          IF%IER< 150,130,150
ISN 0154      130 CONTINUE
            C     WRITE%3,40<
            C     DO 140 I#1,NS
            C     WRITE%3,15< %EM%I,J<,J#1,2<
            C 140 CONTINUE
ISN 0155          DO 144 I=1,NC
ISN 0156          DO 142 J#1,NS
ISN 0157      142 B2%I,J<#0.D0
ISN 0158      144 B2%I,1<#XI%I<
ISN 0159          CALL MMULT%B2,G,A2,NC,1,NS,MD<
ISN 0160          CALL MMULT%A2,SINV,G,NC,NS,NS,MD<
ISN 0161          IF%NAG .EQ. 0< GO TO 147
ISN 0163          DO 145 I#1,NC
ISN 0164          DO 145 J#1,NS
ISN 0165      145 G%I,J<#G%I,J<&NAG*ZG%I,J<
ISN 0166          DO 146 I#1,NS
ISN 0167          DO 146 J#1,NS
ISN 0168      146 A%I,J<#Z%I,J<
ISN 0169      147 CONTINUE
ISN 0170          WRITE%3,50<
ISN 0171          DO 132 I#1,NC
ISN 0172          WRITE%3,7< %G%I,J<,J#1,NS<
ISN 0173      132 CONTINUE
ISN 0174          WRITE%3,55<
ISN 0175          DO 134 I#1,NS
ISN 0176          WRITE%3,7< %T%J,I<,J#1,NS<
ISN 0177      134 CONTINUE
            C
            C     CHECK - CALCULATE EIGENVALUES OF CLOSED-LOOP SYSTEM %A - B*G<
            C
ISN 0178          IF(NCHECK .EQ. 0) GO TO 139
ISN 0180          CALL MMULT%B,G,A1,NS,NC,NS,MD<
```

```
ISN 0181          DO 136 I#1,NS
ISN 0182          DO 136 J#1,NS
ISN 0183      136 A1%I,J<#-A1%I,J<&A%I,J<
ISN 0184          CALL MVECT%A1,AV,NS,MD,MD2<
ISN 0185          CALL HSBG%NS,AV,NS,MD2<
ISN 0186          CALL ATEIG%NS,AV,RR,RI,IANA,NS,MD,MD2<
ISN 0187          WRITE%3,65<
ISN 0188          DO 138 I#1,NS
ISN 0189          WRITE%3,15< RR%I<,RI%I<
ISN 0190      138 CONTINUE
            C
            C     SOLUTION OF ALGEBRAIC MATRIX RICCATI EQUATION.
            C
ISN 0191      139 CONTINUE
ISN 0192          IF%NRICC .EQ. 0< GO TO 155
ISN 0194          READ%1,10< %%Q%I,J<,J#1,NS<,I#1,NS<
ISN 0195          WRITE%3,90<
ISN 0196          DO 1140 I#1,NS
ISN 0197     1140 WRITE%3,7< %Q%I,J<,J#1,NS<
ISN 0198          READ%1,10< %%R%I,J<,J#1,NC<,I#1,NC<
ISN 0199          WRITE%3,91<
ISN 0200          DO 1141 I#1,NC
ISN 0201     1141 WRITE%3,7< %R%I,J<,J#1,NC<
ISN 0202          CALL SSRICC%A,B,Q,R,R1,G,T,B2,SINV,A1,A2,B1,IANA,IROW,W,1.D-4,50,
                 SNS,NC,MD,MD2,AV,AD,XR,XI,SV,SVR,RR,RI,X,Y<
ISN 0203          WRITE%3,92<
ISN 0204          DO 1142 I#1,NC
ISN 0205     1142 WRITE%3,7< %R1%I,J<,J#1,NC<
ISN 0206          WRITE%3,93<
ISN 0207          DO 1143 I#1,NS
ISN 0208     1143 WRITE%3,7< %B1%I,J<,J#1,NS<
            C
            C     CHECK OF SOLUTION BY BACKSUBSTITUTION INTO RICCATI EQUATION.
            C
ISN 0209          IF%NCHECK .EQ. 0< GO TO 155
ISN 0211          DO 1144 I#1,NS
ISN 0212          DO 1144 J#1,NS
ISN 0213          A1%I,J<#0.D0
ISN 0214          DO 1144 K#1,NS
ISN 0215     1144 A1%I,J<#A1%I,J<&B1%I,K<*A%K,J<
ISN 0216          DO 1145 I#1,NS
ISN 0217          DO 1145 J#1,NS
ISN 0218     1145 A2%I,J<#A1%I,J<&A1%J,I<&Q%I,J<
ISN 0219          DO 1146 I#1,NC
ISN 0220          DO 1146 J#1,NS
ISN 0221          T%I,J<#0.D0
ISN 0222          DO 1146 K#1,NS
ISN 0223     1146 T%I,J<#T%I,J<&B%K,I<*B1%K,J<
ISN 0224          DO 1147 I#1,NC
ISN 0225          DO 1147 J#1,NS
ISN 0226          A1%I,J<#0.D0
ISN 0227          DO 1147 K#1,NC
ISN 0228     1147 A1%I,J<#A1%I,J<&R1%I,K<*T%K,J<
ISN 0229          DO 1148 I#1,NS
ISN 0230          DO 1148 J#1,NS
ISN 0231          T%I,J<#0.D0
ISN 0232          DO 1148 K#1,NC
ISN 0233     1148 T%I,J<#T%I,J<&B%I,K<*A1%K,J<
```

```
      DO 1149 I#1,NS
      DO 1149 J#1,NS
      DO 1149 K#1,NS
 1149 A2%I,J<#A2%I,J<-B1%I,K<*T%K,J<
      WRITE%3,94<
      DO 1150 I#1,NS
 1150 WRITE%3,7< %A2%I,J<,J#1,NS<
      GO TC 99
  150 CONTINUE
      WRITE%3,45<
  155 GO TC 99
  160 CONTINUE
      WRITE%3,96<
      STOP
      END
```

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                    DATE  71.095/20.01.38

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              SUBROUTINE HSBG(N,A,IA,ND2)
                 C
                 C      CONVERTS A TO UPPER HESSENBERG FORM
                 C
  ISN 0003              DIMENSION A(ND2)
  ISN 0004              DOUBLE PRECISION A,PIV,T,S
  ISN 0005              DOUBLE PRECISION DABS
  ISN 0006              L=N
  ISN 0007              NIA=L*IA
  ISN 0008              LIA=NIA-IA
  ISN 0009           20 IF(L-3) 360,40,40
  ISN 0010           40 LIA=LIA-IA
  ISN 0011              L1=L-1
  ISN 0012              L2=L1-1
  ISN 0013              ISUB=LIA+L
  ISN 0014              IPIV=ISUB-IA
  ISN 0015              PIV=DABS(A(IPIV))
  ISN 0016              IF(L-3) 90,90,50
  ISN 0017           50 M=IPIV-IA
  ISN 0018              DO 80 I=L,M,IA
  ISN 0019              T=DABS(A(I))
  ISN 0020              IF(T-PIV) 80,80,60
  ISN 0021           60 IPIV=I
  ISN 0022              PIV=T
  ISN 0023           80 CONTINUE
  ISN 0024           90 IF(PIV) 100,320,100
  ISN 0025          100  IF(PIV-DABS(A(ISUB))) 180,180,120
  ISN 0026          120 M=IPIV-L
  ISN 0027              DO 140 I=1,L
  ISN 0028              J=M+I
  ISN 0029              T=A(J)
  ISN 0030              K=LIA+I
  ISN 0031              A(J)=A(K)
  ISN 0032          140 A(K)=T
  ISN 0033              M=L2-M/IA
  ISN 0034              DO 160 I=L1,NIA,IA
  ISN 0035              T=A(I)
  ISN 0036              J=I-M
  ISN 0037              A(I)=A(J)
  ISN 0038          160 A(J)=T
  ISN 0039          180 DO 200 I=L,LIA,IA
  ISN 0040          200 A(I)=A(I)/A(ISUB)
  ISN 0041              J=-IA
  ISN 0042              DO 240 I=1,L2
  ISN 0043              J=J+IA
  ISN 0044              LJ=L+J
  ISN 0045              DO 220 K=1,L1
  ISN 0046              KJ=K+J
  ISN 0047              KL=K+LIA
  ISN 0048          220 A(KJ)=A(KJ)-A(LJ)*A(KL)
  ISN 0049          240 CONTINUE
  ISN 0050              K=-IA
  ISN 0051              DO 300 I=1,N
  ISN 0052              K=K+IA
  ISN 0053              LK=K+L1

                                                                                      PAGE 002

  ISN 0054              S=A(LK)
  ISN 0055              LJ=L-IA
  ISN 0056              DO 280 J=1,L2
  ISN 0057              JK=K+J
  ISN 0058              LJ=LJ+IA
  ISN 0059          280 S=S+A(LJ)*A(JK)*1.0D0
  ISN 0060          300 A(LK)=S
  ISN 0061              DO 310 I=L,LIA,IA
  ISN 0062          310  A(I)=0.0D0
  ISN 0063          320 L=L1
  ISN 0064              GO TO 20
  ISN 0065          360 RETURN
  ISN 0066              END
```

LEVEL 18 (SEPT 69)   OS/360 FORTRAN H   DATE 71.096/22.27..

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              SUBROUTINE ARBEIG(A,B,EM,G,T,A1,A2,B1,X,SV,SVR,NS,NC,K1,K2,IER,MD,
                       1IMULT)
                 C
                 C
                 C      ARBITRARY PLACEMENT OF EIGENVALUES OF THE MATRIX  (A - B*G)
                 C          GAIN MATRIX G WILL ALWAYS BE REAL-VALUED.
                 C          PROGRAM HANDLES BOTH, DISTINCT AND/OR MULTIPLE EIGENVALUES.
                 C
                 C      IER = 0 - SUCCESSFUL, OTHERWISE( (A,B) IS UNCONTROLLABLE.
                 C
                 C      A - (NS * NS)  SYSTEM MATRIX
                 C      B - (NS * NC)  INPUT MATRIX
                 C      G = - C*T(-1)   - (NC *NS)  GAIN MATRIX
                 C      EM - (NS * 2)  MATRIX OF COMPLEX EIGENVALUES,  RE(EM) IN COL. 1,
                 C          IM(EM) IN COL. 2 . IF EM AND A HAVE COMMON EIGENVALUES,
                 C          EM(I,1) =EM(I,1) - 1 .
                 C          BOTH CONJUGATE COMPLEX EIGENVALUES MUST BE PLACED IN SUCCESS-
                 C          IVE ROWS IN THE EM - MATRIX, ALWAYS LIST THE COMPLEX EIGENVA-
                 C          LUE WITH POS. IM. PART FIRST.
                 C          MULTIPLE EIGENVALUES NEED NOT BE INPUTED IN SUCCESSIVE ROWS
                 C          OF THE  EM - MATRIX.
                 C
                 C      MATRICES A AND B ARE UNCHANGED BY THE SUBROUTINE.
                 C
                 C
  ISN 0003              DIMENSION A(MD,MD),B(MD,MD),EM(MD,2),G(MD,MD),A1(MD,MD),A2(MD,MD),
                       1T(MC,MC),B1(MD,MD),X(MD,1),SV(MD),SVR(MD),IMULT(MD,2)
                 C
  ISN 0004              DOUBLE PRECISION A1,A2,B,EM,G,T,A,B1,DABS,X
                 C
  ISN 0005              NC1=NC+1
  ISN 0006              IF(NC1 .GT. NS) GO TO 130
  ISN 0008              L=0
  ISN 0009              DO 120 J=NC1,NS
  ISN 0010              L=L+1
  ISN 0011              DO 100 I=1,NS
  ISN 0012          100 B(I,J)=B(I,L)
  ISN 0013              IF(L-NC) 120,110,110
  ISN 0014          110 L=0
  ISN 0015          120 CONTINUE
  ISN 0016          130 IM=K1-1
  ISN 0017              DO 132 I=1,NS
  ISN 0018              IMULT(I,1)=I
  ISN 0019          132 IMULT(I,2)=0
  ISN 0020          140 IM=IM+1
                 C
                 C      CHECK FOR MULTIPLE EIGENVALUES.
                 C          IF (EM(I,1),EM(I,2)) AND (EM(J,1),EM(J,2)) ARE EQUAL,
                 C          IMULT(J,1)=I , IMULT(J,2)=1 .
                 C
  ISN 0021          141 IF(IM .EQ. 1) GO TO 146
  ISN 0023              IMN=IM-1
  ISN 0024              DO 144 I=1,IMN
  ISN 0025              IF(DABS(EM(I,1)-EM(IM,1)) .LT. .1D-7 .AND. DABS(EM(I,2)-EM(IM,2))
                       1.LT. .1D-7) GO TO 142
```

```
ISN 0027          GO TC 144
ISN 0028      142 IMULT(IM,1)=I
ISN 0029          IMULT(IM,2)=1
ISN 0030      144 CONTINUE
            C
ISN 0031      146 IF(CABS(EM(IM,2))-1.D-8) 150,150,190
            C
            C     REAL EIGENVALUES, DISTINCT AND/OR MULTIPLE.
            C
ISN 0032      150 IE=0
ISN 0033          CO 170 I=1,NS
ISN 0034          CO 16C J=1,NS
ISN 0035      160 A1(I,J)=-A(I,J)
ISN 0036          A1(I,I)=A1(I,I)&EM(IM,1)
ISN 0037      170 B1(I,1)=B(I,IM)
ISN 0038          IF(IMULT(IM,2) .EQ. 0) GO TO 174
ISN 0040          MULT=IMULT(IM,1)
ISN 0041          CO 172 I=1,NS
ISN 0042      172 B1(I,1)=B1(I,1)-T(MULT,I)
ISN 0043      174 CONTINUE
ISN 0044          CALL LINEQS(1,NS,1,A1,B1,K,A1,B1,SV,SVR,IER,D,1.E-10,MD,1)
ISN 0045          IF(IER) 180,280,180
ISN 0046      180 EM(IM,1)=EM(IM,1)-1.DC
ISN 0047          GO TC 141
            C
            C     COMPLEX PAIR OF EIGENVALUES, DISTINCT AND/OR MULTIPLE.
            C
ISN 0048      190 IE=1
ISN 0049          IM1=IM&1
ISN 0050          CO 21C J=1,NS
ISN 0051          CO 2C0 I=1,NS
ISN 0052      200 A2(I,J)=-A(I,J)
ISN 0053      210 A2(J,J)=A2(J,J)&EM(IM,1)
ISN 0054          IF(IMULT(IM,2) .EC. 1) GO TO 232
ISN 0056          CO 23C I=1,NS
ISN 0057          B1(I,1)=C.D0
ISN 0058          CO 22C J=1,NS
ISN 0059          B1(I,1)=B1(I,1)&A2(I,J)*B(J,IM)
ISN 0060          A1(I,J)=0.D0
ISN 0061          CO 220 K=1,NS
ISN 0062      220 A1(I,J)=A1(I,J)&A2(I,K)*A2(K,J)
ISN 0063          B1(I,1)=B1(I,1)&EM(IM,2)*B(I,IM1)
ISN 0064      230 A1(I,I)=A1(I,I)&EM(IM,2)*EM(IM,2)
ISN 0065          GO TC 238
ISN 0066      232 MULT=IMULT(IM,1)
ISN 0067          MULT1=MULT&1
ISN 0068          CO 236 I=1,NS
ISN 0069          B1(I,1)=0.DC
ISN 0070          CO 234 J=1,NS
ISN 0071          B1(I,1)=B1(I,1)&A2(I,J)*(B(J,IM)-T(MULT,J))
ISN 0072          A1(I,J)=C.DC
ISN 0073          CU 234 K=1,NS
ISN 0074      234 A1(I,J)=A1(I,J)&A2(I,K)*A2(K,J)
ISN 0075          B1(I,1)=B1(I,1)&EM(IM,2)*(B(I,IM1)-T(MULT1,I))
ISN 0076      236 A1(I,I)=A1(I,I)&EM(IM,2)*EM(IM,2)
ISN 0077      238 CONTINUE
ISN 0078          CALL LINEQS(1,NS,1,A1,B1,X,A1,B1,SV,SVR,IER,D,1.E-10,MD,1)
ISN 0079          IF(IER) 24C,250,240
```

```
ISN 0080      240 EM(IM,1)=EM(IM,1)-1.DO
ISN 0081          EM(IM1,1)=EM(IM,1)
ISN 0082          GO TO 141
            C
            C     CONSTRUCTION OF  T(TRANS)  MATRIX
            C
ISN 0083      250 DO 270 I=1,NS
ISN 0084          T(IM1,I)=-B(I,IM)
ISN 0085          DO 260 J=1,NS
ISN 0086      260 T(IM1,I)=T(IM1,I)+A2(I,J)*X(J,1)
ISN 0087      270 T(IM1,I)=T(IM1,I)/EM(IM,2)
ISN 0088          IF(IMULT(IM,2) .EQ. 0) GO TO 280
ISN 0090          DO 272 I=1,NS
ISN 0091      272 T(IM1,I)=T(IM1,I)+T(MULT,I)/EM(IM,2)
ISN 0092      280 DO 290 I=1,NS
ISN 0093      290 T(IM,I)=X(I,1)
ISN 0094          IM=IM+IE
ISN 0095          IF(IM-K2) 140,300,300
            C
            C     CALCULATION OF  G = - C*T(-1)
            C
ISN 0096      300 DO 310 I=1,NC
ISN 0097          DO 310 J=1,NS
ISN 0098      310 A2(J,I)=0.D0
ISN 0099          K=1
ISN 0100          DO 320 I=1,NS
ISN 0101          A2(I,K)=1.DO
ISN 0102          K=K+1
ISN 0103          IF(K .GT. NC) K=1
ISN 0105      320 CONTINUE
ISN 0106          CALL LINEQS(1,NS,NC,T,A2,A1,G,B1,SV,SVR,IER,D,1.E-50,MD,ND)
ISN 0107          IF(IER) 350,330,350
ISN 0108      330 DO 340 I=1,NC
ISN 0109          DO 340 J=1,NS
ISN 0110      340 G(I,J)=-A1(J,I)
ISN 0111      350 RETURN
ISN 0112          END
```

```
         COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                            SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002                  SUBROUTINE LINEQS(IOP,N,MB,AA,BB,X,A,B,SV,SVR,IER,D,TOL,MD,ND)
                      C
                      C
                      C     LINEAR MATRIX EQUATION SOLVER. USES GAUSSIAN ELIMINATION WITH
                      C     FULL PIVOTAL CONDENSATION TO SOLVE  AA*X =BB FOR X, WHERE AA IS
                      C     (N * N), BB AND X ARE (N * MB)
                      C
                      C     IOP - OPERATION CODE
                      C         IOP = 1 - STANDARD SOLUTION - INPUTS AA, BB, SOLUTION IN X.
                      C             = 2 - MATRIX INVERSION - INPUT AA, SOLUTION X = AA(-1).
                      C                   BB NOT USED.
                      C             = 3 - NEW RIGHT HAND SIDE (BB) FOR EQUATIONS PREVIOUSLY
                      C                   SOLVED WITH SAME AA MATRIX. INPUT BB AND A, SV,
                      C                   SVR, FROM PREVIOUS RETURN. SOLUTION IN X.
                      C             = 4 - UPPER TRIANGULAR MATRIX INVERSION (NO REDUCTION).
                      C                   INPUT AA (MATRIX TO BE INVERTED). OUTPUT IS
                      C                   X = AA(-1). BB,SV,SVR NOT USED, AA UNCHANGED.
                      C
                      C     MD AND ND DEFINE SIZE OF ARRAYS IN PARAMETER LIST AS INDICATED BY
                      C         DIMENSION STATEMENT.
                      C     STORAGE - A,B,SV,SVR ARE STORAGE ARRAYS OF INDICATED DIMENSIONS.
                      C     AA AND BB ARE UNCHANGED BY SUBROUTINE.
                      C     D - DETERMINANT OF AA.
                      C     IER - ERROR CODE.
                      C         IER = 0 - SUCCESFUL SOLUTION.
                      C             = -1 - N IS .LE. 0 .
                      C             = K .GT. 0 - AA IS SINGULAR OF RANK (K-1).
                      C     AA MATRIX IS CONSIDERED TO BE SINGULAR IF A PIVOT LESS THAN
                      C         TOL*ABS(AAMAX) IS FOUND DURING THE ELIMINATION PROCESS.
                      C         AAMAX IS THE ELEMENT OF LARGEST MAGNITUDE IN THE AA MATRIX.
                      C     N SHOULD NOT EXCEED 100 WITHOUT INCREASING THE SIZE OF THE 'BUF'
                      C         ARRAY.
                      C     IN THE CALL TO LINEQS, THE ONLY MATRICES IN THE SET (AA,BB,X,A,B)
                      C         WHICH MUST BE DIFFERENT ARE A AND B. THAT IS, AA AND A, BB AND
                      C         B MAY BE THE SAME IF THERE IS NO DESIRE TO SAVE AA AND BB.
                      C         ALSO, X CAN BE THE SAME MATRIX AS EITHER A OR B, BUT IF X AND
                      C         A ARE COMMON, A SUBSEQUENT CALL TO LINEQS WITH A NEW BB MATRIX
                      C         (I.E., IOP=3) CANNOT BE MADE.
                      C
                      C
  ISN 0003                  DIMENSION AA(MD,MD),BB(MD,ND),X(MD,ND),A(MD,MD),B(MD,ND),SV(MD),
                           1          SVR(MD)
                      C
  ISN 0004                  DOUBLE PRECISION AA,BB,X,A,B,PIVOT,R,PF
  ISN 0005                  DOUBLE PRECISION BUF(100)
  ISN 0006                  DOUBLE PRECISION DABS
                      C
                      C
  ISN 0007                  EPS=0.
  ISN 0008                  N1=N-1
  ISN 0009                  IB=IOP-2
  ISN 0010                  IF(IB) 70,50,40
                      C
                      C     INVERSION - SET MB=N AND B=1
                      C
  ISN 0011               40 IF(IB-1) 70,70,50
  ISN 0012               50 MB=N
```

```
ISN 0013          DO 60 I=1,N
ISN 0014          DO 55 J=1,N
ISN 0015       55 B(J,I)=0.D0
ISN 0016       60 B(I,I)=1.D0
ISN 0017          IF(IB) 60,80,62
              C
              C   UPPER TRIANGULAR MATRIX INVERSION (NO ELIMINATION).
              C
ISN 0018       62 D=1.
ISN 0019          DO 64 I=1,N
ISN 0020          DO 63 J=1,N
ISN 0021       63 A(J,I)=AA(J,I)
ISN 0022       64 D=D*A(I,I)
ISN 0023          IF(DABS(D)-TOL) 66,66,200
ISN 0024       66 IER=1
ISN 0025          RETURN
ISN 0026       70 DO 75 I=1,MB
ISN 0027          DO 75 J=1,N
ISN 0028       75 B(J,I)=BB(J,I)
ISN 0029          IF(IB) 80,80,100
ISN 0030       80 DO 90 I=1,N
ISN 0031          DO 85 J=1,N
ISN 0032       85 A(J,I)=AA(J,I)
ISN 0033       90 SV(I)=I
ISN 0034          D=1.
ISN 0035      100 IF(N1) 101,150,102
ISN 0036      101 IER=-1
ISN 0037          RETURN
              C
              C
              C   ELIMINATION LOOP (THROUGH STATEMENT 126) .
              C
              C   SEARCH FOR LARGEST ELEMENT IN LOWER (NE * NE) BLOCK OF A (= PIVOT)
              C
ISN 0038      102 DO 126 NE=1,N1
ISN 0039          IF(IB) 103,103,110
ISN 0040      103 BF=DABS(A(NE,NE))
ISN 0041          PIVOT=A(NE,NE)
ISN 0042          NR=NE
ISN 0043          NC=NE
ISN 0044          DO 106 J=NE,N
ISN 0045          DO 106 I=NE,N
ISN 0046          IF(DABS(A(I,J))-BF) 106,106,104
ISN 0047      104 NR=I
ISN 0048          NC=J
ISN 0049          BF=DABS(A(I,J))
ISN 0050          PIVOT=A(I,J)
ISN 0051      106 CONTINUE
ISN 0052          IF(NE .EQ. 1) EPS=TOL*DABS(PIVOT)
ISN 0054          D=D*PIVOT
ISN 0055          SVR(NE)=NR
              C
              C   SINGULARITY CHECK
              C
ISN 0056          IF(DABS(PIVOT) - EPS) 108,108,110
ISN 0057      108 IER=NE
ISN 0058          D=0.
ISN 0059          RETURN
```

```
ISN 0060        110 NR=SVR(NE)
ISN 0061            IF(NR-NE) 117,117,111
                C
                C       ROW INTERCHANGE - A(NR,K) WITH A(NE,K) FOR K = NE TO N
                C                     - B(NR,K) WITH B(NE,K) FOR K = 1 TO MB
                C
ISN 0062        111 IF(IE) 112,112,115
ISN 0063        112 DO 114 K=NE,N
ISN 0064            BF=A(NR,K)
ISN 0065            A(NR,K)=A(NE,K)
ISN 0066        114 A(NE,K)=BF
ISN 0067        115 DO 116 K=1,MB
ISN 0068            BF=B(NR,K)
ISN 0069            B(NR,K)=B(NE,K)
ISN 0070        116 B(NE,K)=BF
ISN 0071        117 IF(IE) 1171,1171,122
ISN 0072       1171 IF(NC-NE) 122,122,118
                C
                C       COLUMN INTERCHANGE - A(K,NE) WITH A(K,NC) FOR K = 1 TO N
                C          NOTE - SV(I) IS THE ORIGINAL UNKNOWN VARIABLE NO. (I.E., COLUMN
                C          NUMBER) NOW OCCUPYING COLUMN I IN THE REDUCED ARRAY.
                C
ISN 0073        118 BF=SV(NC)
ISN 0074            SV(NC)=SV(NE)
ISN 0075            SV(NE)=BF
ISN 0076            DO 120 K=1,N
ISN 0077            BF=A(K,NC)
ISN 0078            A(K,NC)=A(K,NE)
ISN 0079        120 A(K,NE)=BF
                C
                C       REDUCTION LOOP - R = A(I,NE)/PIVOT = A(I,NE)/A(NE,NE)
                C                        A(I,J) = A(I,J) - R*A(NE,J) FOR I,J = NE&1 TO N
                C                        B(I,J) = B(I,J) - R*B(NE,J) FOR I=NE&1,N J=1,MB
                C          R IS STORED IN A(I,NE) (LOWER PART OF A MATRIX) FOR SUBSEQUENT
                C          CALLS WITH A NEW RIGHT HAND SIDE (BB MATRIX) - IOP = 3.
                C
ISN 0080        122 NE1=NE&1
ISN 0081            DO 126 I=NE1,N
ISN 0082            IF(IP) 1231,1231,123
ISN 0083        123 R=A(I,NE)
ISN 0084            GO TO 125
ISN 0085       1231 R=A(I,NE)/PIVOT
ISN 0086            A(I,NE)=R
ISN 0087            DO 124 J=NE1,N
ISN 0088        124 A(I,J)=A(I,J)-R*A(NE,J)
ISN 0089        125 DO 126 J=1,MB
ISN 0090        126 B(I,J)=B(I,J)-R*B(NE,J)
                C
                C       END OF ELIMINATION LOOP
                C
                C
                C       FINAL SINGULARITY CHECK
                C
ISN 0091        150 IF(DABS(A(N,N))-EPS) 152,152,170
ISN 0092        152 IER=N
ISN 0093            D=0.
ISN 0094            RETURN
ISN 0095        170 IF(IE .GT. 0) GO TO 200
```

```
ISN 0097          D=D*A(N,N)
               C
               C     BACK SUBSTITUTION AND SOLUTION
               C        X(I,K) = (B(I,K) - SUM(J=I+1 TO N) (A(I,J)*X(J,K)))/A(I,I)
               C        FOR I = N TO 1, AND EACH COLUMN OF B (K=1 TO MB).
               C
ISN 0098      200 DO 210 K=1,MB
ISN 0099          X(N,K)=B(N,K)/A(N,N)
ISN 0100          I=N
ISN 0101      202 I=I-1
ISN 0102          IF(I) 210,210,204
ISN 0103      204 M1=I+1
ISN 0104          BF=0.
ISN 0105          DO 206 J=M1,N
ISN 0106      206 BF=BF+A(I,J)*X(J,K)
ISN 0107          X(I,K)=(B(I,K)-BF)/A(I,I)
ISN 0108          GO TO 202
ISN 0109      210 CONTINUE
ISN 0110          IER=0
ISN 0111          IF(IR-2) 211,220,211
               C
               C     ROW EXCHANGE - PUT X VARIABLES INTO PROPER PLACE IN X MATRIX
               C        X(M,J) = X(I,J) WHERE M=SV(I), FOR J=1 TO MB .
               C
ISN 0112      211 DO 214 J=1,MB
ISN 0113          DO 212 I=1,N
ISN 0114      212 BUF(I)=X(I,J)
ISN 0115          DO 214 I=1,N
ISN 0116          K=SV(I)
ISN 0117      214 X(K,J)=BUF(I)
               C
ISN 0118      220 RETURN
ISN 0119          END
```

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                         DATE  71.096/22.24.

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002            SUBROUTINE ATEIG(M,A,RR,RI,IANA,IA,MD,MD2)
                C
                C     COMPUTES ROOTS OF UPPER HESSENBERG MATRIX A
                C
  ISN 0003            DIMENSION A(MD2),RR(MD),RI(MD),PRR(2),PRI(2),IANA(MD)
  ISN 0004            DOUBLE PRECISION E7,E6,E10,DELTA,PRR,PRI,PAN,PAN1,R,S,T,A,U,V,RR,
                     1                 RI,RMOD,EPS,D,G1,G2,G3,CAP,PSI1,PSI2,ALPHA,ETA
  ISN 0005            DOUBLE PRECISION DABS,DSQRT,DMAX1
  ISN 0006            INTEGER F,P1,O
  ISN 0007            E7=1.0D-8
  ISN 0008            E6=1.0D-6
  ISN 0009            E10=1.0D-10
  ISN 0010            DELTA=0.5D0
  ISN 0011            MAXIT=30
  ISN 0012            N=M
  ISN 0013         20 N1=N-1
  ISN 0014            IN=N1*IA
  ISN 0015            NN=IN&N
  ISN 0016            IF(N1) 30,1300,30
  ISN 0017         30 NP=N&1
  ISN 0018            IT=0
  ISN 0019            DO 40 I=1,2
  ISN 0020            PRR(I)=0.0D0
  ISN 0021       40   PRI(I)=0.0D0
  ISN 0022            PAN=0.0D0
  ISN 0023            PAN1=0.0D0
  ISN 0024            R=0.0D0
  ISN 0025            S=0.0D0
  ISN 0026            N2=N1-1
  ISN 0027            IN1=IN-IA
  ISN 0028            NN1=IN1&N
  ISN 0029            N1N=IN&N1
  ISN 0030            N1N1=IN1&N1
  ISN 0031         60 T=A(N1N1)-A(NN)
  ISN 0032            U=T*T
  ISN 0033            V=4.0D0*A(N1N)*A(NN1)
  ISN 0034            IF(DABS(V)-U*E7) 100,100,65
  ISN 0035         65 T=U&V
  ISN 0036            IF(DABS(T)-DMAX1(U,DABS(V))*E6) 67,67,68
  ISN 0037       67   T=0.0D0
  ISN 0038       68   U=(A(N1N1)&A(NN))/2.0D0
  ISN 0039            V=DSQRT(DABS(T))/2.0D0
  ISN 0040            IF(T)140,70,70
  ISN 0041         70 IF(U) 80,75,75
  ISN 0042         75 RR(N1)=U&V
  ISN 0043            RR(N)=U-V
  ISN 0044            GO TO 130
  ISN 0045         80 RR(N1)=U-V
  ISN 0046            RR(N)=U&V
  ISN 0047            GO TO 130
  ISN 0048        100 IF(T)120,110,110
  ISN 0049        110 RR(N1)=A(N1N1)
  ISN 0050            RR(N)=A(NN)
  ISN 0051            GO TO 130
  ISN 0052        120 RR(N1)=A(NN)
```

```
ISN 0053           RR(N)=A(NIN1)
ISN 0054      130  RI(N)=C.CDO
ISN 0055           RI(N1)=0.CDO
ISN 0056           RI(N1)=0.0
ISN 0057           GO TC 160
ISN 0058       140 RR(N1)=U
ISN 0059           RR(N)=U
ISN 0060           RI(N1)=V
ISN 0061           RI(N)=-V
ISN 0062       160 IF(N2)1280,1280,180
ISN 0063       180 N1N2=N1N1-IA
ISN 0064           RMOD=RR(N1)*RR(N1)&RI(N,)*RI(N1)
ISN 0065           EPS=E10*DSQRT(RMOD)
ISN 0066           IF(DABS(A(N1N2))-EPS) 1280,1280,240
ISN 0067      240  IF(DABS(A(NN1))-E10*DABS(A(NN))) 1300,1300,250
ISN 0068      250  IF(DABS(PAN1-A(N1N2))-DABS(A(N1N2))*E6) 1240,1240,260
ISN 0069      260  IF(DABS(PAN-A(NN1))-DABS(A(NN1))*E6) 1240,1240,300
ISN 0070       300 IF(IT-MAXIT) 320,1240,1240
ISN 0071       320 J=1
ISN 0072           DO 360 I=1,2
ISN 0073           K=NP-I
ISN 0074           IF(DABS(RR(K)-PRR(I))&DABS(RI(K)-PRI(I))-DELTA*(DABS(RR(K))
                 1      &DABS(RI(K)))) 340,360,360
ISN 0075       340 J=J&I
ISN 0076       360 CONTINUE
ISN 0077           GO TC (440,460,460,480),J
ISN 0078      440  R=0.CDC
ISN 0079           S=0.CDO
ISN 0080           GO TC 500
ISN 0081       460 J=N&2-J
ISN 0082           R=RR(J)*RR(J)
ISN 0083           S=RR(J)&RR(J)
ISN 0084           GO TC 500
ISN 0085       480 R=RR(N)*RR(N1)-RI(N)*RI(N1)
ISN 0086           S=RR(N)&RR(N1)
ISN 0087       500 PAN=A(NN1)
ISN 0088           PAN1=A(N1N2)
ISN 0089           DO 520 I=1,2
ISN 0090           K=NP-I
ISN 0091           PRR(I)=RR(K)
ISN 0092       520 PRI(I)=RI(K)
ISN 0093           P=N2
ISN 0094           IF(N-3)600,600,525
ISN 0095       525 IPI=N1N2
ISN 0096           DO 580 J=2,N2
ISN 0097           IPI=IPI-IA-1
ISN 0098           IF(DABS(A(IPI))-EPS) 600,600,530
ISN 0099       530 IPIP=IPI&IA
ISN 0100           IPIP2=IPIP&IA
ISN 0101           D=A(IPIP)*(A(IPIP)-S)&A(IPIP2)*A(IPIP&1)&R
ISN 0102           IF(D)540,560,540
ISN 0103      540  IF(DABS(A(IPI)*A(IPIP&1))*(DABS(A(IPIP)&A(IPIP2&1)-S)&DABS(A(IPIP2
                 1      &2)))-DABS(D)*EPS) 620,620,560
ISN 0104       560 P=N1-J
ISN 0105       580 CONTINUE
ISN 0106       600 Q=P
ISN 0107           GO TC 680
ISN 0108       620 P1=P-1
```

```
          Q=P1
          IF(P1-1)680,680,650
      650 DO 660 I=2,P1
          IPI=IPI-IA-1
          IF(DABS(A(IPI))-EPS) 680,680,660
      660 Q=Q-1
      680 II=(P-1)*IA&P
          DO 1220 I=P,N1
          III=II-IA
          IIP=II&IA
          IF(I-P)720,700,720
      700 IPI=III&1
          IPIP=IIP&1
          G1=A(II)*(A(II)-S)&A(IIP)*A(IPI)&R
          G2=A(IPI)*(A(IPIP)&A(II)-S)
          G3=A(IPI)*A(IPIP&1)
          A(IPI&1) =0.0D0
          GO TO 780
      720 G1=A(III)
          G2=A(III&1)
          IF(I-N2)740,740,760
      740 G3=A(III&2)
          GO TO 780
     760  G3=0.0D0
     780  CAP=DSQRT(G1*G1&G2*G2&G3*G3)
          IF(CAP)800,860,800
      800 IF(G1)820,840,840
      820 CAP=-CAP
      840 T=G1&CAP
          PSI1=G2/T
          PSI2=G3/T
          ALPHA=2.0D0/(1.0D0&PSI1*PSI1&PSI2*PSI2)
          GO TO 880
     860  ALPHA=2.0D0
          PSI1=0.0D0
          PSI2=0.0D0
      880 IF(I-Q)900,960,900
      900 IF(I-P)920,940,920
      920 A(III)=-CAP
          GO TO 960
      940 A(III)=-A(III)
      960 IJ=II
          DO 1040 J=I,N
          T=PSI1*A(IJ&1)
          IF(I-N1)980,1000,1000
      980 IP2J=IJ&2
          T=T&PSI2*A(IP2J)
     1000 ETA=ALPHA*(T&A(IJ))
          A(IJ)=A(IJ)-ETA
          A(IJ&1)=A(IJ&1)-PSI1*ETA
          IF(I-N1)1020,1040,1040
     1020 A(IP2J)=A(IP2J)-PSI2*ETA
     1040 IJ=IJ&IA
          IF(I-N1)1080,1060,1060
     1060 K=N
          GO TO 1100
     1080 K=I&2
     1100 IP=IIP-I
```

```
      CO 118C J=0,K
      JIP=IP&J
      JI=JIP-IA
      T=PSI1*A(JIP)
      IF(I-N1)112C,1140,114C
 1120 JIP2=JIP&IA
      T=T&PSI2*A(JIP2)
 1140 ETA=ALPHA*(T&A(JI))
      A(JI)=A(JI)-ETA
      A(JIP)=A(JIP)-ETA*PSI1
      IF(I-N1)116C,1180,1180
 1160 A(JIP2)=A(JIP2)-ETA*PSI2
 1180 CONTINUE
      IF(I-N2)12C0,1220,1220
 12C0 JI=II&3
      JIP=JI&IA
      JIP2=JIP&IA
      ETA=ALPHA*PSI2*A(JIP2)
      A(JI)=-ETA
      A(JIP)=-ETA*PSI1
      A(JIP2)=A(JIP2)-ETA*PSI2
 1220 II=IIP&1
      IT=IT&1
      GO TC 60
 1240 IF(DABS(A(NN1))-CABS(A(N1N2))) 13C0,1280,1280
 1280 IANA(N)=0
      IANA(N1)=2
      N=N2
      IF(N2)14C0,1400,2C
 1300 RR(N)=A(NN)
      RI(N)=C.0C0
      IANA(N)=1
      IF(N1)14C0,1400,1320
 1320 N=N1
      GO TC 20
 14C0 RETURN
      END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                          SOURCE,ECD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              SUBROUTINE EIGVEC%IVC,  A, B, W, IROW, XR, XI, VR, VI, ROOTRE,
                      1    ROOTIE, NE, NMAX, T2, SWI, COUNTE, ERR,MMM<
                C          SUBROUTINE TO FIND THE EIGENVECTORS OF A NON-SYMMETRIC MATRIX
                C          BY A MODIFIED WILKINSONOS INVERSE ITERATION METHOD.
                C          CONTROL IVC CODE IS
                C               1  FIND ONLY THE REGULAR EIGENVECTORS    %A X # LAMBDA X<
                C               2  FIND ONLY THE TRANSPOSED EIGENVECTORS %AT V # LAMBDA V<
                C               3  FIND BOTH TYPES OF EIGENVECTORS.
  ISN 0003              DIMENSION A%NMAX,NMAX<,B%NMAX,NMAX<,W%NMAX,4<,XR%NMAX<,XI%NMAX<,
                      1        VR%NMAX<,VI%NMAX<,IROW%NMAX,2<
  ISN 0004              DOUBLE PRECISION ROOTR,ROOTI,ROOTRE,ROOTIE,TEMP,TEMP2,AMAX,C1,C2,
                      1            SWI,W,XR,XI,VR,VI,B,ZERO,DCERR,A
  ISN 0005              DOUBLE PRECISION DABS,DSIGN,DSQRT,DMAX1
  ISN 0006              INTEGER COUNT, COUNTE, T2
  ISN 0007              IO1#1
  ISN 0008              IO3#3
  ISN 0009              ROOTR # ROOTRE
  ISN 0010              ROOTI # ROOTIE
  ISN 0011              N # NE
  ISN 0012              MM # MMM - 1
  ISN 0013              N1 # N - 1
  ISN 0014              NP1 # N & 1
  ISN 0015              IVC1 # IVC - 1
  ISN 0016              IVC2 # IVC1 - 1
  ISN 0017              COUNT # 1
  ISN 0018              DO 400 I#1,N
  ISN 0019              W%I,1<#0.0D0
  ISN 0020              XR%I<#0.0D0
  ISN 0021         400  CONTINUE
  ISN 0022              CLIM # 1.0E-4
  ISN 0023              IF%ROOTI< 1, 60, 1
                C
                C          COMPLEX EIGENVALUE.
                C
  ISN 0024            1 TEMP # - ROOTR - ROOTR
  ISN 0025              ISW # 2
  ISN 0026              TEMP2#ROOTR*ROOTR&ROOTI*ROOTI
  ISN 0027              JJ # 300
  ISN 0028              DO 606 I # 1, N
  ISN 0029              IF%T2< 600, 603, 600
  ISN 0030          600 DO 602 J # 1, N
  ISN 0031              JJ # JJ & 1
  ISN 0032              IF%JJ - 251< 602, 601, 601
  ISN 0033          601 JJ # 1
  ISN 0034              READ %T2< %WELL,1<, LL # 1,250<
  ISN 0035          602 B%I,J< # A%I,J<*TEMP & W%JJ,1<
  ISN 0036              GO TO 605
  ISN 0037          603 DO 604 J # 1, N
  ISN 0038          604 B%I,J< # A%I,J<*TEMP & B%I,J<
  ISN 0039          605 B%I,I< # B%I,I< & TEMP2
  ISN 0040          606 A%I,I< # A%I,I< - ROOTR
  ISN 0041              IF%T2 .NE. 0< REWIND T2
  ISN 0043              GO TO 700
  ISN 0044          607 IF%ICC< 622, 608, 622
                C
```

```
              C          MATRIX SINGULAR.                                                 ESYI 460
              C                                                                           ESYI 470
ISN 0045        622 IF%IVC2< 623, 625, 623                                                ESYI 480
ISN 0046        623 DO 624 LL # 1, N                                                      ESYI 490
ISN 0047            W%LL,2<#0.0D0
ISN 0048        624 XI%LL<#0.0D0
ISN 0049            IF%IVC1< 625, 514, 625                                                ESYI 510
ISN 0050        625 DO 626 LL # 1, N                                                      ESYI 520
ISN 0051            W%LL,4<#0.0D0
ISN 0052        626 VI%LL<#0.0D0
ISN 0053            GO TO 511                                                             ESYI 540
              C                                                                           ESYI 550
              C          MATRIX NOT SINGULAR.                                             ESYI 560
              C                                                                           ESYI 570
ISN 0054        608 DO 609 LL # 1, N                                                      ESYI 580
ISN 0055            W%LL,1<#1.0D0
ISN 0056            W%LL,2<#1.0D0
ISN 0057            W%LL,3<#1.0D0
ISN 0058        609 W%LL,4<#1.0D0
ISN 0059        699 IF%IVC2< 610, 612, 610                                                ESYI 600
ISN 0060        610 DO 611 I # 1, N                                                       ESYI 610
ISN 0061            I2 # IROW%I,2<                                                        ESYI 620
ISN 0062            XI%I2< # W%I,1<*ROOTI                                                 ES8  63
ISN 0063            DO 611 J # 1, N                                                       ESYI 640
ISN 0064        611 XI%I2< # XI%I2< & A%I,J<*W%J,2<                                       ESYI 650
'SN 0065            IF%IVC1< 612, 500, 612                                                ESYI 660
ISN 0066        612 DO 613 I # 1, N                                                       ESYI 670
ISN 0067            VI%I< # W%I,3<*ROOTI                                                  ESYI 680
ISN 0068            DO 613 J # 1,N                                                        ESYI 690
ISN 0069        613 VI%I< # VI%I< & A%J,I<*W%J,4<                                         ESYI 700
 SN 0070            GO TO 499                                                             ESYI 710
 SN 0071        615 CERR # 0.0                                                            ESYI 720
 SN 0072            DCERR#0.0D0
ISN 0073            IF%IVC2< 616, 619, 616                                                ESYI 730
ISN 0074        616 DO 618 I # 1, N                                                       ESYI 740
ISN 0075            XR%I< # -W%I,2<                                                       ESYI 750
ISN 0076            DO 617 J # 1, N                                                       ESYI 760
ISN 0077        617 XR%I< # XR%I< & A%I,J<*XI%J<                                          ESYI 770
ISN 0078        618 XR%I< # XR%I</ROOTI                                                   ESYI 780
ISN 0079            IF%IVC1< 619, 633, 619                                                ESYI 790
ISN 0080        619 DO 621 I # 1, N                                                       ESYI 800
ISN 0081            VR%I< # -W%I,4<                                                       ESYI 810
ISN 0082            DO 620 J # 1, N                                                       ESYI 820
ISN 0083        620 VR%I< # VR%I< & A%J,I<*VI%J<                                          ESYI 830
ISN 0084        621 VR%I< # VR%I</ROOTI                                                   ESYI 840
              C                                                                           ESYI 850
              C          SEARCH VECTORS FOR LARGEST ELEMENT AND NORMALIZE.                ESYI 860
              C                                                                           ESYI 870
ISN 0085        627 AMAX#0.0D0
ISN 0086            DO 629 L # 1, N                                                       ESYI 890
ISN 0087            TEMP # VR%L<**2 & VI%L<**2                                            ESYI 900
ISN 0088            IF%TEMP - AMAX< 629, 629, 628                                         ESYI 910
ISN 0089        628 AMAX # TEMP                                                           ESYI 920
ISN 0090            I2 # L                                                                ESYI 930
ISN 0091        629 CONTINUE                                                              ESYI 940
ISN 0092            C1 # VR%I2</AMAX                                                      ESYI 950
ISN 0093            C2 # -VI%I2</AMAX                                                     ESYI 960
ISN 0094            DO 630 L # 1, N                                                       ESYI 970
```

```
ISN 0095          TEMP # VI%L<                                              ESV1 980
ISN 0096          VI%L< # VR%L<*C2 & TEMP*C1                                ESV1 990
ISN 0097      630 VR%L< # VR%L<*C1 - TEMP*C2                                ESV11C00
ISN 0098          IF%COUNT .EQ. 1< GO TO 632                                ESV11010
ISN 0100          DO 631 LL # 1, N                                          ESV11020
ISN 0101     631  DCERR#DMAX1%DCERR,DABS%VR%LL<-W%LL,3<<,DABS%VI%LL<-W%LL,4<<<
ISN 0102      632 IF%INC2< 633, 638, 633                                    ESV11040
ISN 0103     633  AMAX#C.0D0
ISN 0104          DO 635 L # 1, N                                           ESV11060
ISN 0105          TEMP # XR%L<**2 & XI%L<**2                                ESV11070
ISN 0106          IF%TEMP - AMAX< 635, 635, 634                             ESV11080
ISN 0107      634 AMAX # TEMP                                               ESV11090
ISN 0108          I2 # L                                                    ESV11100
ISN 0109      635 CONTINUE                                                  ESV11110
ISN 0110          C1 # XR%I2</AMAX                                          ESV11120
ISN 0111          C2 # -XI%I2</AMAX                                         ESV11130
ISN 0112          DO 636 L # 1, N                                           ESV11140
ISN 0113          TEMP # XI%L<                                              ESV11150
ISN 0114          XI%L< # XR%L<*C2 & TEMP*C1                                ESV11160
ISN 0115      636 XR%L< # XR%L<*C1 - TEMP*C2                                ESV11170
ISN 0116          IF%COUNT .EQ. 1< GO TO 646                                ESV11180
ISN 0118          DO 637 LL # 1, N                                          ESV11190
ISN 0119     637  DCERR#DMAX1%DCERR,DABS%XR%LL<-W%LL,1<<,DABS%XI%LL<-W%LL,2<<<
               C                                                            ESV11210
               C        TEST FOR CONVERGENCE.                               ESV11220
               C                                                            ESV11230
ISN 0120      638 IF%COUNT .EQ. 1< GO TO 646                                ESV11240
ISN 0122          CERR#DCERR
ISN 0123          IF%CERR .GE. 1.0E-4< GO TO 639                            ESV11250
ISN 0125          IF%CERR .GE. CLIM< GO TO 648                              ESV11260
ISN 0127          CLIM # CERR                                               ESV11270
ISN 0128          IF%CLIM .LE. 1.0E-8< GO TO 648                            ESV11280
ISN 0130      639 IF%COUNT .GE. 15< GO TO 68                                ESV11290
ISN 0132      647 COUNT # COUNT & 1                                         ESV11300
ISN 0133          IF%RCOI1< 642, 673, 642                                   ESV11310
ISN 0134      642 IF%INC2< 640, 644, 640                                    ESV11320
ISN 0135      640 DO 641 LL # 1, N                                          ESV11330
ISN 0136          W%LL,1< # XR%LL<                                          ESV11340
ISN 0137      641 W%LL,2< # XI%LL<                                          ESV11350
ISN 0138          IF%INC1< 644, 610, 644                                    ESV11360
ISN 0139      644 DO 645 LL # 1, N                                          ESV11370
ISN 0140          W%LL,3< # VR%LL<                                          ESV11380
ISN 0141      645 W%LL,4< # VI%LL<                                          ESV11390
ISN 0142          GO TO 699                                                 ESV11400
ISN 0143      646 CERR # 0.0                                                ESV11410
ISN 0144          DCERR#0.0D0
ISN 0145          IF%ICC< 648, 647, 648                                     ESV11420
ISN 0146      648 ERR # CERR                                                ESV11430
ISN 0147          COUNTE # COUNT                                            ESV11440
ISN 0148          IF%RCOI1< 667, 668, 667                                   ESV11450
ISN 0149      667 DO 649 I # 1, N                                           ESV11460
ISN 0150      649 A%I,I< # A%I,I< & RCOTR                                   ESV11470
ISN 0151          RETURN                                                    ESV11480
ISN 0152       68 PRINT 101, RCOTR, RCOTI, CERR                             ESV11490
ISN 0153          GO TO 648                                                 ESV11500
               C                                                            ESV11510
               C        REAL EIGENVECTORS.                                  ESV11520
               C                                                            ESV11530
```

```
ISN 0154          60 ISN # 1                                                     ESV11540
ISN 0155             DO 651 I # 1, N                                             ESV11550
ISN 0156             DO 65C J # 1, N                                             ESV11560
ISN 0157         650 B%I,J< # A%I,J<                                             ESV11570
ISN 0158         651 B%I,I< # B%I,I< - RCOTR                                     ESV11580
ISN 0159             GO TC 700                                                   ESV11590
ISN 0160         652 IF%ICC< 680, 685, 680                                       ESV11600
               C                                                                 ESV11610
               C          SINGULAR MATRIX.                                       ESV11620
               C                                                                 ESV11630
ISN 0161         680 IF%IVC2< 681, 683, 681                                      ESV11640
ISN 0162         681 DO 682 L # 1, N                                             ESV11650
ISN 0163        682  XI%L<#0.ODO
ISN 0164             IF%IVC1< 683, 514, 683                                      ESV11670
ISN 0165         683 DO 684 L # 1, N                                             ESV11680
ISN 0166        684  VI%L<#0.ODO
ISN 0167             GO TC 511                                                   ESV11700
               C                                                                 ESV11710
               C          MATRIX NOT SINGULAR.                                   ESV11720
               C                                                                 ESV11730
ISN 0168         685 IF%IVC2< 653, 656, 653                                      ESV11740
ISN 0169         653 DO 654 L # 1, N                                             ESV11750
ISN 0170        654  XI%L<#1.ODO
ISN 0171             IF%IVC1< 656, 500, 656                                      ESV11770
ISN 0172         656 DO 657 L # 1, N                                             ESV  78
ISN 0173        657  VI%L<#1.ODO
ISN 0174             GO TC 499                                                   ESV11800
               C                                                                 ESV11810
               C          NORMALIZE REAL VECTORS.                                ESV11820
               C                                                                 ESV11830
ISN 0175         655 CERR # 0.0                                                  ESV11840
ISN 0176             DCERR#0.ODO
ISN 0177             IF%IVC2< 658, 662, 658                                      ESV11850
ISN 0178        658  C1#C.CDO
ISN 0179             C2#C.CDO
ISN 0180             DO 66C L # 1, N                                             ESV11870
ISN 0181             TEMP#DABS%XI%L<<
ISN 0182             IF%TEMP - C1< 660, 660, 659                                 ESV11890
ISN 0183         659 C1 # TEMP                                                   ESV11900
ISN 0184             C2 # XI%L<                                                  ESV11910
ISN 0185         660 CONTINUE                                                    ESV11920
ISN 0186             DO 661 L # 1, N                                             ESV11930
ISN 0187             XI%L< # XI%L</C2                                            ESV11940
ISN 0188             DCERR#DMAX1%DCERR,DABS%XI%L<-XR%L<<<
ISN 0189         661 XR%L< # XI%L<                                               ESV11960
ISN 0190             IF%IVC1< 662, 638, 662                                      ESV11970
ISN 0191        662  C2#C.CDO
ISN 0192             C1#C.ODO
ISN 0193             DO 664 L # 1, N                                             ESV11990
ISN 0194             TEMP#DABS%VI%L<<
ISN 0195             IF%TEMP - C1< 664, 664, 663                                 ESV12010
ISN 0196         663 C1 # TEMP                                                   ESV12020
ISN 0197             C2 # VI%I<                                                  ESV12030
ISN 0198         664 CONTINUE                                                    ESV12040
ISN 0199             DO 665 LL # 1, N                                            ESV12050
ISN 0200             VI%LL< # VI%LL</C2                                          ESV12060
ISN 0201             DCERR#DMAX1%DCERR,DABS%VI%LL<-W%LL,1<<<
ISN 0202             W%LL,1<#VI%LL<
```

```
ISN 0203        665 VR%LL<A#%LL,I<                                   
ISN 0204            GO TC 638                                        ESV12090
ISN 0205        668 IF%IVC2< 669, 671, 669                           ESV12100
ISN 0206        669 DO 670 L # 1, N                                  ESV12110
ISN 0207       670  V%I%L<#0.0D0
ISN 0208            IF%IVC1< 671, 70, 671                            ESV12130
ISN 0209        671 DO 672 L # 1, N                                  ESV12140
ISN 0210       672  VI%L<#0.0D0
ISN 0211         70 RETURN                                           ESV12160
ISN 0212        673 IF%IVC2< 674, 502, 674                           ESV12170
ISN 0213        674 DO 675 I # 1, N                                  ESV12180
ISN 0214            I2 # IKO#%I,2<                                   ESV12190
ISN 0215        675 XI%I2< # XR%I<                                   ESV12200
ISN 0216            GO TC 500                                        ESV12210
               C                                                     ESV12220
               C         BACK SUBSTITUTION SECTION.                  ESV12230
               C                                                     ESV12240
ISN 0217        499 IF%IVC2< 500, 502, 500                           ESV12250
ISN 0218        500 DO 501 I # 2, N                                  ESV12260
ISN 0219            I1 # I - 1                                       ESV12270
ISN 0220            DO 501 J # 1, I1                                 ESV12280
ISN 0221        501 XI%I< # XI%I< - B%I,J<*XI%J<                     ESV12290
ISN 0222        511 IF%IVC1< 502, 514, 502                           ESV12300
ISN 0223        502 DO 510 I # 1, N                                  ESV12310
ISN 0224            I1 # I - 1                                       ESV12320
ISN 0225            IF%I1< 503, 505, 503                             ESV12330
ISN 0226        503 DO 504 J # 1, I1                                 ESV12340
ISN 0227        504 VI%I< # VI%I< - B%J,I<*VI%J<                     ESV12350
ISN 0228            IF%ICC< 505, 506, 505                            ESV1236C
 SN 0229        505 IF%B%I,I<< 506, 507, 506                         ESV12370
 SN 0230        506 VI%I< # VI%I</B%I,I<                             ESV12380
.SN 0231            GO TC 510                                        ESV12390
ISN 0232        507 IF%VI%I<< 508, 509, 508                          ESV12400
ISN 0233        508 VI%I< # VI%I<*1.0E&15                            ESV12410
ISN 0234            GO TC 510                                        ESV12420
ISN 0235        509 VI%I< # 1.0                                      ESV12430
ISN 0236        510 CONTINUE                                         ESV12440
ISN 0237            IF%IVC2< 514, 525, 514                           ESV12450
ISN 0238        514 DO 522 I # 1, N                                  ESV12460
ISN 0239            IK # NP1 - I                                     ESV12470
ISN 0240            IF%I - 1< 515, 517, 515                          ESV12480
ISN 0241        515 I2 # IK & 1                                      ESV12490
ISN 0242            DO 516 J # I2, N                                 E"V12500
ISN 0243        516 XI%IK< # XI%IK< - B%IK,J<*XI%J<                  ESV12510
ISN 0244            IF%ICC< 517, 518, 517                            ESV12520
ISN 0245        517 IF%B%IK,IK<< 518, 519, 518                       ESV12530
ISN 0246        518 XI%IK< # XI%IK</B%IK,IK<                         ESV12540
ISN 0247            GO TC 522                                        ESV12550
ISN 0248        519 IF%XI%IK<< 520, 521, 520                         ESV12560
ISN 0249        520 XI%IK< # XI%IK<*1.0E&15                          ESV12570
ISN 0250            GO TC 522                                        ESV12580
ISN 0251       521  XI%IK<#1.0D0
ISN 0252        522 CONTINUE                                         ESV12600
ISN 0253            IF%IVC1< 525, 529, 525                           ESV12610
ISN 0254        525 DO 526 I # 2, N                                  ESV12620
ISN 0255            IK # NP1 - I                                     ESV12630
ISN 0256            I2 # IK & 1                                      ESV12640
ISN 0257            DO 526 J # I2, N                                 ESV12650
```

```
ISN 0258        526 VI%IR< # VI%IR< - B%J,IR<*VI%J<                ESV12660
ISN 0259            DO 527 L # 1, N                                ESV12670
ISN 0260            I2 # IROW%L,I<                                 ESV12680
ISN 0261        527 VH%I2< # VI%L<                                 ESV12690
ISN 0262            DO 528 L # 1, N                                ESV12700
ISN 0263        528 VI%L< # VH%L<                                  ESV12710
ISN 0264        529 IF%REC%I< 615, 655, 615                        ESV12720
              C                                                    ESV12730
              C         FACTOR MATRIX.                             ESV12740
              C                                                    ESV12750
ISN 0265        700 ICC # 0                                        ESV12760
ISN 0266            SWI#1.0D72
ISN 0267            DO 701 LL # 1, N                               ESV12780
ISN 0268        701 IROW%LL,I< # LL                                ESV12790
ISN 0269            DO 708 K # 1, N1                               ESV12800
ISN 0270            AMAX#DABS%B%K,K<<
ISN 0271            IMAX # K                                       ESV12820
ISN 0272            K1 # K & 1                                     ESV12830
ISN 0273            DO 702 I # K1, N                               ESV12840
ISN 0274            IF%AMAX.GT.DABS%B%I,K<<< GO TO 702
ISN 0276            AMAX#DABS%B%I,K<<
ISN 0277            IMAX # I                                       ESV12870
ISN 0278        702 CONTINUE                                       ESV12880
ISN 0279            IF%AMAX .LT. SWI< SWI # AMAX                   ESV12890
ISN 0281            IF%AMAX.GE.1.0D-25< GO TO 723
ISN 0283            B%K,K<#0.0D0
ISN 0284            ICC # ICC & 1                                  ESV12920
ISN 0285            GO TO 708                                      ESV12930
ISN 0286        723 IF%IMAX .EQ. K< GO TO 704                      ESV12940
ISN 0288            DO 703 J # 1, N                                ESV12950
ISN 0289            AMAX # B%K,J<                                  ESV12960
ISN 0290            B%K,J< # B%IMAX,J<                             ESV12970
ISN 0291        703 B%IMAX,J< # AMAX                               ESV12980
ISN 0292            I2 # IROW%K,I<                                 ESV12990
ISN 0293            IROW%K,I< # IROW%IMAX,I<                       ESV13000
ISN 0294            IROW%IMAX,I< # I2                              ESV13010
ISN 0295        704 DO 707 I # K1, N                               ESV13020
ISN 0296            IF%B%I,K<< 705, 707, 705                       ESV13030
ISN 0297        705 B%I,K< # B%I,K</B%K,K<                         ESV13040
ISN 0298            DO 706 J # K1, N                               ESV13050
ISN 0299        706 B%I,J< # B%I,J< - B%K,J<*B%I,K<                ESV13060
ISN 0300        707 CONTINUE                                       ESV13070
ISN 0301        708 CONTINUE                                       ESV13080
ISN 0302            AMAX#DABS%B%N,N<<
ISN 0303            IF%AMAX-1.0D-25< 712,712,713
ISN 0304        712 B%N,N<#0.0D0
ISN 0305            SWI#0.0D0
ISN 0306            ICC # ICC & 1                                  ESV13120
ISN 0307            GO TO 709                                      ESV13130
ISN 0308        713 IF%AMAX .LT. SWI< SWI # AMAX                   ESV13140
ISN 0310        709 IF%ICC .LE. ISW< GO TO 710                     ESV13150
ISN 0312            IF%K%< 1050,1050,1051
ISN 0313       1050 WRITE%103,102< ICC
ISN 0314            COUNTE # 0                                     ESV13180
ISN 0315            RETURN                                         ESV13190
ISN 0316       1051 WRITE%103,1052< ICC
ISN 0317        710 DO 711 LL # 1, N                               ESV13210
ISN 0318            I2 # IROW%LL,I<                                ESV13220
```

```
ISN 0319       711 IROW9I2,2< # LL                                               ESY13230
ISN 0320           IF*RECII< 607, 652, 607                                       ESY13240
ISN 0321      1052 FORMAT#///23H ****** WARNING ******     ,a   SUBROUTINE EIGVEC HAS ESY13250
                  1FOUND AN EIGENVALUE OF APPARENT MULTIPLICITY#,                ESY13260
                  1                                   I4,/23X,a   COMPUTATION OF EIESY13270
                  2GENVECIORTS< CONTINUES AT USER S CPTIONa//<                   ESY13280
ISN 0322       1C1 FORMAT#38HMORE THAN 15 LCOPS FOR EIGENVECTOR OF,2E12.4,       ESY13290
                  2 ' 14H DIFFERENCE OF,E12.4<                                   ESY13300
ISN 0323       1C2 FORMAT#16H0****WARNING**** , I4, 71H ZEROS ON DIAGONAL OF FACTOREDESY13310
                  1 MATRIX.  CHECK FCR MULTIPLE EIGENVALUES./20X,                ESY13320
                  2a SUBROUTINE EIGVEC WILL NOT PERFORM COMPUTATION FOR THIS EIGENVECESY13330
                  3TOR a//<                                                      ESY13340
ISN 0324           END
```

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                                   DATE  71.106/19.50.

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                           SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              SUBROUTINE SINGLE%SB,D,RI,GG,IMULT,NCON,NS,NC,MD<
                C
                C       PROGRAM CONVERTS MULTI-INPUT SYSTEM INTO PSEUDU SINGLE-INPUT
                C          SYSTEM
                C
  ISN 0003              REAL*8 SB%MD,MD<,D%MD<,RI%MD<,GG%MD<,P1,PIV,DABS,GSE
  ISN 0004              DIMENSION IMULT%MD,2<
                C
                C       PROGRAM CHECKS CONTROLLABILITY OF %A,B<
                C
  ISN 0005              NP#0
  ISN 0006              GSE#0.D0
  ISN 0007              DO 100 J#1,NC
  ISN 0008              DO 100 I#1,NS
  ISN 0009          100 GSE#GSE&DABS%SB%I,J<<
  ISN 0010              GSE#GSE/%NS*NC<
  ISN 0011              DO 140 I#1,NS
  ISN 0012              NCON#0
  ISN 0013              IMULT%I,2<#0
  ISN 0014              DO 110 J#1,NC
  ISN 0015          110 IF%%DABS%SB%I,J<<-GSE*1.D-8< .GT. 0.D0< NCON#NCON&1
  ISN 0017              IF%DABS%RI%I<<-1.D-8< 130,130,120
  ISN 0018          120 NP#NP&1
  ISN 0019              IF%NCON .EQ. 0< IMULT%I,2<#1
  ISN 0021              IF%NP-1< 140,140,125
  ISN 0022          125 I1#I-1
  ISN 0023              IF%%IMULT%I,2<&IMULT%I1,2<< .EQ. 2< GO TO 330
  ISN 0025              NP#0
  ISN 0026              GO TO 140
  ISN 0027          130 IF%NCON .EQ. 0< GO TO 330
  ISN 0029          140 CONTINUE
                C
                C       COMPUTATION OF SINGLE-INPUT VECTOR  D # SB*G .
                C
  ISN 0030              DO 170 I#1,NS
  ISN 0031              D%I<#0.D0
  ISN 0032              DO 170 J#1,NC
  ISN 0033          170 D%I<#D%I<&SB%I,J<
  ISN 0034              DO 180 I#1,NC
  ISN 0035          180 GG%I<#1.D0
  ISN 0036              IF%NC .EQ. 1< GO TO 325
                C
                C       TEST WHETHER  D  RENDERS %L,D< CONTROLLABLE
                C
  ISN 0038              N1#1
  ISN 0039          185 NP#0
  ISN 0040              I#N1-1
  ISN 0041          188 I#I&1
  ISN 0042              NCON#0
  ISN 0043              N1#I
  ISN 0044              IF%DABS%D%I<< .GT. GSE*1.D-8< NCON#1
  ISN 0046              IF%DABS%RI%I<<-1.D-8< 210,210,190
  ISN 0047          190 NP#NP&1
  ISN 0048              IF%NCON .EQ. 0< IMULT%I,2<#1
  ISN 0050              IF%NP-1< 220,220,200
```

```
ISN 0051        200 I1#I-1
ISN 0052            IF%%IMULT%I,2<&IMULT%I1,2<< .EQ. 2< GO TO 230
ISN 0054            NP#0
ISN 0055            GO TO 220
ISN 0056        210 IF%NCON .EQ. 0< GO TO 230
ISN 0058        220 IF%I-NS< 188,225,225
ISN 0059        225 GO TO 325
               C
               C     FIND NON-ZERO ELEMENT IN ROW N1 OF MATRIX  SB .
               C
ISN 0060        230 PIV#SB%N1,1<
ISN 0061            MSB#1
ISN 0062            DO 250 I#2,NC
ISN 0063            IF%DABS%PIV<-DABS%SB%N1,I<<< 240,250,250
ISN 0064        240 PIV#SB%N1,I<
ISN 0065            MSB#I
ISN 0066        250 CONTINUE
               C
               C     FIND ELEMENT OF LARGEST MAGNITUDE, PIV, IN COL.-NO. MSB OF MATRIX
               C         SB. FIND NON-ZERO ELEMENT OF SMALLEST MAGNITUDE, P1, IN VEC. D
               C
ISN 0067        260 DO 270 I#1,NS
ISN 0068            N2#I
ISN 0069            IF%DABS%D%I<<-GSE*1.D-8< 280,270,280
ISN 0070        270 CONTINUE
ISN 0071        280 P1#D%N2<
ISN 0072            DO 290 I#1,NS
ISN 0073            IF%DABS%PIV< .LT. DABS%SB%I,MSB<<< PIV#SB%I,MSB<
ISN 0075            IF%DABS%D%I<< .LT. GSE*1.D-8< GO TO 290
ISN 0077            IF%DABS%P1< .LT. DABS%D%I<<< P1#D%I<
ISN 0079        290 CONTINUE
ISN 0080            P1#DABS%PIV/P1<&1.D-8
ISN 0081            N2#P1&1
ISN 0082            P1#N2
ISN 0083            DO 300 I#1,NS
ISN 0084        300 D%I<#P1*D%I<&SB%I,MSB<
ISN 0085            DO 310 I#1,NC
ISN 0086        310 GG%I<#P1*GG%I<
ISN 0087            GG%MSB<#GG%MSB<&1.D0
ISN 0088            IF%N1-NS< 320,325,325
ISN 0089        320 N1#N1&1
ISN 0090            GO TO 185
ISN 0091        325 NCON#1
ISN 0092            GO TO 340
ISN 0093        330 NCON#0
ISN 0094        340 RETURN
ISN 0095            END
```

```
LEVEL 18 ( SEPT 69 )                         OS/360  FORTRAN H                                    DATE  71.106/19.51.54

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002               SUBROUTINE SIMTR2%A,AA,AA1,S,SINV,W,IROW,RR,RI,XR,XI,VR,VI,NS,MD,
                        SIVC<
                   C
                   C     COMPUTES SIMILARITY TRANSFORMATION MATRIX  SINV  FOR MATRIX  A
                   C         WITH SIMPLE EIGENVALUES. YIELDS REAL-VALUED TRANSFORMATION
                   C         MATRIX.
                   C
  ISN 0003               REAL*8 A%MD,MD<,AA%MD,MD<,AA1%MD,MD<,SINV%MD,MD<,RR%MD<,RI%MD<
  ISN 0004               REAL*8 XR%MD<,XI%MD<,VR%MD<,VI%MD<,DABS,DSQRT
  ISN 0005               REAL*8 W%MD,4<,S%MD,MD<,SW1
  ISN 0006               DIMENSION IROW%MD,2<
  ISN 0007            10 FORMAT(//T3 'EIGENVECTOR ERROR MESSAGE')
  ISN 0008            20 FORMAT(T3,'SW1=',E10.4,10X,'ITER=',I5,10X,'DIF=',E10.4)
  ISN 0009               K=0
  ISN 0010           100 CONTINUE
  ISN 0011               DO 110 J=1,NS
  ISN 0012               DO 110 I=1,NS
  ISN 0013           110 AA1(I,J)=AA(I,J)
  ISN 0014               K=K+1
  ISN 0015               CALL EIGVEC%IVC,A,AA1,W,IROW,XR,XI,VR,VI,RR%K<,RI%K<,NS,MD,0,SW1,
                        EITER,DIF,2<
  ISN 0016               IF(ITER .LT. 15) GO TO 111
  ISN 0018               WRITE%3,10<
  ISN 0019               WRITE%3,20< SW1,ITER,DIF
  ISN 0020           111 CONTINUE
  ISN 0021               IF%DABS%RI%K<< .GT. 1.D-8< GO TO 130
                   C
                   C     COL. AND/OR ROW EIGENVECTORS CORRESPONDING TO A REAL EIGENVALUE
                   C
  ISN 0023               W(1,1)=0.D0
  ISN 0024               DO 120 I=1,NS
  ISN 0025               W(1,1)=W(1,1)+DABS(VR(I))
  ISN 0026           120 SINV%K,I<=VR%I<
  ISN 0027               IF%IVC-2< 126,126,122
  ISN 0028           122 W(1,3)=0.D0
  ISN 0029               DO 124 I=1,NS
  ISN 0030               SINV(K,I)=SINV(K,I)/W(1,1)
  ISN 0031               W(1,3)=W(1,3)+SINV(K,I)*XR(I)
  ISN 0032           124 S%I,K<=XR%I<
  ISN 0033               DO 123 I=1,NS
  ISN 0034           123 S(I,K)=S(I,K)/W(1,3)
  ISN 0035           126 IF%K-NS< 100,150,150
                   C
                   C     COMPLEX COL. AND/OR ROW EIGENVECTORS ARE CONVERTED TO A SET OF TWO
                   C         REAL-VALUED TRANSFORMATION VECTORS
                   C
  ISN 0036           130 K1=K+1
  ISN 0037               W(1,1)=0.D0
  ISN 0038               W(1,2)=0.D0
  ISN 0039               DO 140 I=1,NS
  ISN 0040               W(1,1)=W(1,1)+DABS(VR(I))
  ISN 0041               W(1,2)=W(1,2)+DABS(VI(I))
  ISN 0042               SINV%K,I<=2.D0*VR%I<
  ISN 0043           140 SINV%K1,I<=-2.D0*VI%I<
  ISN 0044               IF%IVC-2< 136,136,132
```

```
ISN 0045     132 IF(W(I,1) .LT. W(I,2)) W(I,1)=W(I,2)
ISN 0047         W(I,3)=0.D0
ISN 0048         W(I,4)=0.D0
ISN 0049         DO 134 I=1,NS
ISN 0050         SINV(K,I)=SINV(K,I)/(2.D0*W(1,1))
ISN 0051         SINV(K1,I)=SINV(K1,I)/(2.D0*W(1,1))
ISN 0052         W(1,3)=W(1,3)+.5D0*SINV(K,I)*XR(I)+.5D0*SINV(K1,I)*XI(I)
ISN 0053     134 W(1,4)=W(1,4)+.5D0*SINV(K,I)*XI(I)-.5D0*SINV(K1,I)*XR(I)
ISN 0054         W(1,1)=W(1,3)*W(1,3)+W(1,4)*W(1,4)
ISN 0055         DO 135 I=1,NS
ISN 0056         S(I,K)=(XR(I)*W(1,3)+XI(I)*W(1,4))/W(1,1)
ISN 0057     135 S(I,K1)=(XI(I)*W(1,3)-XR(I)*W(1,4))/W(1,1)
ISN 0058     136 K=K1
ISN 0059         IF(K-NS) 100,150,150
ISN 0060     150 RETURN
ISN 0061         END
```

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                    DATE  71.106/19.43.2

         COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                            SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002         SUBROUTINE MVECTR(A,AV,NS,MD,MD2)
            C
            C    CONVERTS MATRIX A INTO VECTOR AV
            C
ISN 0003         REAL*8 A(MD,MD),AV(MD2)
ISN 0004         DO 10 J=1,NS
ISN 0005         DO 10 I=1,NS
ISN 0006         K=(J-1)*NS+I
ISN 0007      10 AV(K)=A(I,J)
ISN 0008         RETURN
ISN 0009         END
```

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                    DATE  71.106/19.49.2

         COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                            SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002         SUBROUTINE MMULT(A,B,C,NS,NC,NB,MD)
            C
            C    COMPUTES MATRIX PRODUCT  C = A*B
            C
ISN 0003         REAL*8 A(MD,MD),B(MD,MD),C(MD,MD)
ISN 0004         DO 10 I=1,NS
ISN 0005         DO 10 J=1,NB
ISN 0006         C(I,J)=0.D0
ISN 0007         DO 10 K=1,NC
ISN 0008      10 C(I,J)=C(I,J)+A(I,K)*B(K,J)
ISN 0009         RETURN
ISN 0010         END
```

```
LEVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                    DATE  71.095/20.11.26

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              SUBROUTINE SSRICC(A,B,C,R,R1,P,P1,X,Y,AUX1,AUX2,AUX3,IANA,IROW,
                       NW,TOL,MAXIT,NS,NC,MC,MC2,AV,AC,XR,XI,SV,SVR,RR,RI,X1,Y1)
                 C
                 C         SOLVES STEADY-STATE MATRIX RICCATI EQUATION
                 C             A(T)*P + P*A - P*B*R(-1)*B(T)*P + C = 0
                 C             WHERE  A IS A (NS*NS) MATRIX,
                 C                    B IS A (NS*NC) MATRIX,
                 C                    C IS A (NS*NS) P.S.D. MATRIX,
                 C                    R IS A (NC*NC) P.D. MATRIX AND
                 C                    P IS A (NS*NS) P.D.  MATRIX.
                 C
                 C         RICCATI EQUATION IS SOLVED VIA KLEINMAN'S SUCCESSIVE APPROXIMATION
                 C             METHOD. KLEINMAN'S ITERATIVE EQUATION IS SOLVED VIA EIGEN-
                 C             SYSTEM APPROACH.
                 C
                 C         A,B,C,R WILL NOT BE CHANGED BY SUBROUTINE. R1=R(-1) WILL BE
                 C             COMPUTED IN SUBROUTINE.
                 C         IT IS ASSUMED, THAT (A - B*R(-1)*B(T)*P) HAS A COMPLETE SET OF
                 C             EIGENVECTORS (I.E. IS SIMILAR TO A DIAGONAL MATRIX).
                 C
                 C         MATRIX  AUX3  WILL CONTAIN THE FINAL SOLUTION, THE RICCATI MATRIX.
                 C
  ISN 0003               REAL*8 A(MD,MD),B(MC,MC),C(MC,MC),R(MC,MC),R1(MC,MC),P(MD,MD),
                        1       P1(MD,MD),X(MC,MC),Y(MC,MC),RR(MC),RI(MC),SV(MC),SVR(MD),
                        2       AV(MD2),AUX1(MC,MC),AUX2(MC,MC),AUX3(MC,MC),AC(MC,MC),
                        3       W(MD,4),XR(MC),XI(MC)
  ISN 0004               REAL*8 X1(MD,1),Y1(MC,1)
  ISN 0005               REAL*8 TOL,SW1,DEL,DABS
  ISN 0006               DIMENSION IANA(MC),IROW(MC,2)
  ISN 0007             5 FORMAT(5D20.8)
  ISN 0008            10 FORMAT(//,'  IER=-1 , CHECK ORDER OF MATRICES IN CALL STATEMNT FOR
                        1 LINEQS.')
  ISN 0009            15 FORMAT(//,'  IER=',I3,' , MATRIX SINGULAR (OF RANK IER). NOT POSSI
                        2BLE.. CHECK INPUT MATRICES.')
  ISN 0010            20 FORMAT(//,'  TOL .LE. ',E15.7,' WAS NOT ACHIEVED IN MAXIT=',I3,
                        3' ITERATIONS.')
  ISN 0011            25 FORMAT(//,'  GAIN TOLERANCE .LE. ',E15.7,'WAS ACHIEVED AFTER ',I3,
                        4' ITERATIONS.')
  ISN 0012            30 FORMAT(//,'  R(INV) CANNOT BE COMPUTED. CHECK R .')
  ISN 0013            40 FORMAT(//,2X,'CHECK  ',4I6)
  ISN 0014               CALL LINEQS(2,NC,NC,R,X,R1,P1,Y,SV,SVR,IER,0,1.E-10,MC,MD)
  ISN 0015               IF(IER) 500,90,500
  ISN 0016            90 ITER=0
  ISN 0017           100 ITER=ITER+1
  ISN 0018               DO 110 J=1,NS
  ISN 0019               DO 110 I=1,NS
  ISN 0020               AUX1(I,J)=A(I,J)
  ISN 0021               DO 110 K=1,NC
  ISN 0022           110 AUX1(I,J)=AUX1(I,J)-B(I,K)*P(K,J)
                 C
                 C       DETERMINE EIGENVALUES AND -VECTORS OF (A-B*C)
                 C
  ISN 0023               CALL MVECT(AUX1,AV,NS,MC,MC2)
  ISN 0024               CALL HSBG(NS,AV,NS,MC2)
  ISN 0025               CALL ATEIG(NS,AV,RR,RI,IANA,NS,MC,MC2)
```

```
      CALL MMULT(ALX1,ALX1,P1,NS,NS,NS,MD)
      CALL SIMTR2(ALX1,P1,AUX2,X,Y,W,IROW,RR,RI,XR,XI,SV,SVR,NS,MD,3)
      DO 130 J=1,NS
      DO 120 I=1,NS
  120 AD(I,J)=0.D0
  130 AD(J,J)=RR(J)
      I=0
  140 I=I+1
      IF(DABS(RI(I))-1.D-8) 160,160,150
  150 IP=I+1
      AD(I,IP)=-RI(I)
      AD(IP,I)=RI(I)
      I=IP
  160 IF(I-NS) 140,170,170
  170 CONTINUE
C
C     RIGHT HAND SIDE OF TRANSFORMED ITERATIVE EQUATION
C
      DO 180 J=1,NS
      DO 180 I=1,NC
      P1(I,J)=0.D0
      DO 180 K=1,NC
  180 P1(I,J)=P1(I,J)+R(I,K)*F(K,J)
      DO 190 J=1,NS
      DO 190 I=1,NS
      ALX2(I,J)=-C(I,J)
      DO 190 K=1,NC
  190 ALX2(I,J)=ALX2(I,J)-P(K,I)*P1(K,J)
      DO 200 J=1,NS
      DO 200 I=1,NS
      P1(I,J)=0.D0
      DO 200 K=1,NS
  200 P1(I,J)=P1(I,J)+ALX2(I,K)*X(K,J)
      DO 210 J=1,NS
      DO 210 I=1,NS
      ALX2(I,J)=0.D0
      DO 210 K=1,NS
  210 ALX2(I,J)=ALX2(I,J)+X(K,I)*P1(K,J)
C
C     SOLVE  (LAMBDA*I + L(T))*T(I) = D(I) FOR  T(I)
C
      KL=0
  220 KL=KL+1
      IF(DABS(RI(KL))-1.D-8) 230,230,280
C
C     REAL EIGENVALUES OF  (A-B*R(-1)*B(T)*P)
C
  230 DO 250 J=KL,NS
      KJ=J-KL+1
      DO 240 I=KL,NS
      KI=I-KL+1
  240 ALX1(KI,KJ)=AD(I,J)
  250 ALX1(KJ,KJ)=ALX1(KJ,KJ)+RR(KL)
      KN=NS-KL+1
      DO 260 I=1,KN
      KI=KL+I-1
  260 X1(I,1)=ALX2(KI,KL)
      CALL LINEQS(1,KN,1,ALX1,X1,Y1,AUX1,X1,SV,SVR,IER,0,1.E-10,MD,1)
```

```
ISN 0075          IF(IER) 480,265,490
ISN 0076      265 DO 270 I=1,KA
ISN 0077          KI=KL+I-1
ISN 0078          P1(KL,KI)=Y1(I,1)
ISN 0079      270 P1(KI,KL)=Y1(I,1)
ISN 0080          IF(KL-NS) 220,380,380
              C
              C   COMPLEX EIGENVALUES OF (A - B*R(-1)*B(T)*P)
              C
ISN 0081      280 DO 300 J=KL,NS
ISN 0082          KJ=J-KL+1
ISN 0083          DO 290 I=KL,NS
ISN 0084          KI=I-KL+1
ISN 0085      290 ALX1(KI,KJ)=AD(I,J)
ISN 0086      300 ALX1(KJ,KJ)=ALX1(KJ,KJ)+RR(KL)
ISN 0087          KA=NS-KL+1
ISN 0088          DO 310 J=1,KA
ISN 0089          DO 310 I=1,KA
ISN 0090          ALX3(I,J)=0.D0
ISN 0091          DO 310 K=1,KA
ISN 0092      310 ALX3(I,J)=ALX3(I,J)+ALX1(I,K)*AUX1(K,J)
ISN 0093          DO 320 I=1,KA
ISN 0094      320 ALX3(I,I)=ALX3(I,I)+RI(KL)*RI(KL)
ISN 0095          KL1=KL+1
ISN 0096          DO 330 I=1,KA
ISN 0097          KI=KL+I-1
ISN 0098          X1(I,1)=RI(KL)*ALX2(KI,KL1)
ISN 0099          DO 330 J=1,KA
ISN 0100          KJ=KL+J-1
ISN 0101      330 X1(I,1)=X1(I,1)+ALX1(I,J)*AUX2(KJ,KL)
ISN 0102          CALL LINEQS(1,KA,1,ALX3,X1,Y1,AUX3,X1,SV,SVR,IER,0,1.E-10,MD,1)
ISN 0103          IF(IER) 480,340,490
ISN 0104      340 DO 350 I=1,KA
ISN 0105          KI=KL+I-1
ISN 0106          P1(KL,KI)=Y1(I,1)
ISN 0107      350 P1(KI,KL)=Y1(I,1)
ISN 0108          DO 355 I=1,KA
ISN 0109          KI=KL+I-1
ISN 0110      355 P1(KI,KL1)=-ALX2(KI,KL)/RI(KL)
ISN 0111          DO 370 I=1,KA
ISN 0112          KI=KL+I-1
ISN 0113          DO 360 J=1,KA
ISN 0114      360 P1(KI,KL1)=P1(KI,KL1)+ALX1(I,J)*Y1(J,1)/RI(KL)
ISN 0115      370 P1(KL1,KI)=P1(KI,KL1)
ISN 0116          KL=KL1
ISN 0117          IF(KL-NS) 220,380,380
              C
              C   COMPUTATION OF THE  RICCATI MATRIX  P(TG1)
              C
ISN 0118      380 DO 390 I=1,NS
ISN 0119          DO 390 J=1,NS
ISN 0120          ALX1(I,J)=0.D0
ISN 0121          DO 390 K=1,NS
ISN 0122      390 ALX1(I,J)=ALX1(I,J)+P1(I,K)*Y(K,J)
ISN 0123          DO 400 I=1,NS
ISN 0124          DO 400 J=1,NS
ISN 0125          ALX3(I,J)=0.D0
ISN 0126          DO 400 K=1,NS
```

```
ISN 0127          400 AUX3(I,J)=AUX3(I,J)+EV(K,I)*AUX1(K,J)
              C
              C       COMPUTATION OF THE GAIN MATRIX  K(I+1)  AND COMPARISON WITH  K(I)
              C
ISN 0128              DO 410 I=1,NC
ISN 0129              DO 410 J=1,NS
ISN 0130              AUX1(I,J)=0.D0
ISN 0131              DO 410 K=1,NS
ISN 0132          410 AUX1(I,J)=AUX1(I,J)+EB(K,I)*AUX3(K,J)
ISN 0133              DO 420 I=1,NC
ISN 0134              DO 420 J=1,NS
ISN 0135              P1(I,J)=0.D0
ISN 0136              DO 420 K=1,NC
ISN 0137          420 P1(I,J)=P1(I,J)+ER1(I,K)*AUX1(K,J)
ISN 0138              DO 430 J=1,NS
ISN 0139              DO 430 I=1,NC
ISN 0140          430 AUX1(I,J)=P1(I,J)-P(I,J)
ISN 0141              DEL=0.D0
ISN 0142              DO 440 J=1,NS
ISN 0143              DO 440 I=1,NC
ISN 0144              IF(DABS(P1(I,J)) .GT. TOL) SV(I)=DABS(AUX1(I,J)/P1(I,J))
ISN 0146              IF(SV(I) .GT. DEL) DEL=SV(I)
ISN 0148          440 CONTINUE
ISN 0149              IF(DEL-TOL) 460,460,450
ISN 0150          450 IF(ITER-MAXIT) 462,470,470
ISN 0151          462 DO 464 J=1,NS
ISN 0152              DO 464 I=1,NC
ISN 0153          464 P(I,J)=P1(I,J)
ISN 0154              GO TO 100
ISN 0155          460 WRITE(3,25) DEL,ITER
ISN 0156              RETURN
ISN 0157          470 WRITE(3,20) TOL,MAXIT
ISN 0158              RETURN
ISN 0159          480 WRITE(3,10)
ISN 0160              RETURN
ISN 0161          490 WRITE(3,15) IER
ISN 0162              RETURN
ISN 0163          500 WRITE(3,30)
ISN 0164              RETURN
ISN 0165              END
```