DESIGN OF NUCLEAR REACTOR CONTROL SYSTEM WITH TIGHT FEEDBACK

A DISSERTATION
submitted in partial fulfilment
of the requirements for the award of the Degree
of
MASTER OF ENGINEERING
in
ELECTRICAL ENGINEERING
(System Engineering & Operations Research)

By

SURESH KUMAR SHARMA

DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITY OF ROORKEE
ROORKEE, U.P.
November, 1976

C E R T I F I C A T E

Certified that the dissertation entitled " DESIGN OF NUCLEAR REACTOR CONTROL SYSTEM WITH TIGHT FEEDBACK" which is being submitted by Sri Suresh Kumar Sharma in partial fulfillment for the award of Degree of Master of Engineering in " SYSTEM ENGINEERING AND OPERATIONS RESEARCH" of the University of Roorkee, Roorkee is a record of student's own work carried out by him under my supervision and guidance. The matter embodied in this dissertation has not been submitted for the award of any other degree or diploma.

This is further to certify that he has worked for a period of Eight months from March to Nov., 1976 for preparing dissertation for Master of Engineering Degree at this University.

Dated: November 20, 1976

Jayanta Pal.

(JAYANTA PAL)
LECTURER
Electrical Engineering Department
University of Roorkee, Roorkee

...

ACKNOWLEDGEMENT

The author wishes to express his deep sense of gratitude to Sri JAYANTA PAL, Lecturer in Electrical Engineering Department, University of Roorkee, Roorkee for initiating this topic and his constant encouragement and able guidance accorded to him at every stage of this work, without which it would have not been possible to bring the work upto this stage in the given time. Author is greatly indebted for having the privilege of being associated with him in the preparation of this present dissertation.

The author sincerely thanks Dr. T.S.M. Rao for providing all the facilities in connection with this dissertation work, during his tenure in the office as Professor and Head of the Electrical Engineering Department, University of Roorkee, Roorkee.

Thanks are due to Dr. M.N.K. Rao, Incharge Computer Centre, Structural Engineering Research Centre, Roorkee, for allowing the author to use IBM 1620 and to give him special time for the job as required, and to Sri M.K.Vasantha, Reader, Electrical Engineering Department, University of Roorkee, Roorkee, for his help. He also thanks Mr. R.S. Chaudhry (M.E. Final Year student, Power System) and S.P. Srivastav (M.E. Final Year Student, Power apparatus and Electric drives) for their co-operation and help.

Last but not least author thanks to operators of Computer Centre, for listing the results and Programs, and to every body who has in any way directly or indirectly or by inspirations influenced his work.

SYNOPSIS

The present work covers the design of Nuclear Reactor Control System with tight feedback, The design uses the modified form of the Sequential Return Difference (SRD) method. The SRD method is modified to reduce computational work and, hence, make the algorithm faster.

Sequential return difference method gives a satisfactory closed-loop linear system via a sequence of single loop designs. The stability of the resulting system may be checked by classical techniques such as Routh Hurqitz Criterion, Nyquist Criterion or Root locii. Here, as the number of equations to be checked for stability is very large. Routh Hurwitz criterion is utilized. In the Ist Chapter return difference and return ratio matrices are discussed, and how these are useful in the synthesis is given as a background for the SRD method. Second Chapter describes the SRD method. Modified transfer function matrix achieved from the SRD method is checked for step input. All the output are given in the end.

The SRD method is found to give good controller designs using only single-input single-output classical (well established) design techniques in the frequency domain. This powerful method can be fruitfully used to design complex systems if inter-active computer graphic terminals are available. In this work, however, only IBM 1620 with punched output facility was available and inordinately long CPU time (this computer is very slow-50 times slower than IBM 360) was involved.

Computer programs in FORTRAN II were developed for

(a) Sequential Return difference algorithm

(b) Routh Hurwitz criterion (**SUBROUTINE**)

(c) Ensuring tight feedback

(d) Nyquist criterion

(e) Step response with subroutines of Polynomial multiplication and polynomial summation

(f) Determining the Laplace inverse and generation of time response curves for three output.

CONTENTS

Contd.

Page

CHAPTER-I

INTRODUCTION

Design of Control Systems has seen rapid progress in recent years. With modern developments in design techniques, complexities in the controller structure are also increasing at the same rate. For this reason, designers now have started to look for design of multivariable systems using the old classical techniques. It has also been brought up to the practical stage by few designers. The present work is based upon the views of such control Engineers. The work of a control engineer is not to just formulate and analyse the control problem but ultimately he has to be involved in the design of total control systems which is a complex subject in itself.

A design problem has never a unique solution. A given set of specifications may be satisfied by a number of transfer functions. More-over leaving technical specifications, the requirements of weight, size and cost may introduce extra constraints. Here the considerations for weight, size and cost has not been taken into account. The problem has been tackled only from stability and performance point of view.

1.1. NEED OF THE CONTROL SYSTEM:

In a feedback control system, the requirement of a controller can in no case be avoided. Whenever the error signal between desired and actual output is received, one should be able to control the system to make the error to converge to zero eventually. In the case of a Nuclear Reactor on sudden removal of power, fluctuations will occur at the outputs such as voltage, frequency and power. These fluctuations have to be damped out in a reasonable time so that relevant safety regulations are adhered to. Therefore controlling the output of a reactor under required specifications becomes a necessity.

1.2. TOWARDS THE DESIGN OF CONTROL SYSTEM :

The last few decades has seen an explosion of knowledge in the field of control systems. Two distinct classifications may be made viz classical control methods and modern control methods. The classical methods, which allow more insight in the dynamics of the system in the form of root locus. Bode plots, Nyquist and Polar plots etc., however, suffer from the fact that they are basically suited to single-input single-output systems only, The modern control theory, on the other hand, gives powerful tools for the design of controllers for multivariable systems. Barring the new techniques of pole placement etc. modern control methods eg optimal control etc. suffer from the basic drawback of requirement of on line digital controller for control purpose plus the added disadvantage of trial-and-error schemes involved in selecting the weighting matrices.

With the advent of high speed digital computers, methods came up which utilize the classical well-known (well understood) techniques eg. Nyquist Bode etc. for the design of multivariable systems. The work of H.H.Rosenbrock and D.Q. Mayne falls under this class of classico-modern approach.

1.3. DESIGN CRITERIONS:

Besides providing a useful theoretical basis, ROSENBROCK described a useful technique based on the inverse Nyquist array, for designing linear multivariable systems, and a similar technique using the Nyquist array and classical methods are extended by MacFaralane. In this approach extensions are done by Dr. D.Q. Mayne. Here a series compensator is chosen to transform the plant transfer function $G_P(S)$, which is the $m \times m$ matrix transfer function of the system being controlled to $G(S) = G_P(S)\ G_C(S)$ where $G(S)$ is diagonally dominant. Then m single loop control problems are considered, choosing $K_i(S)$, $i = 1\text{-}m$.

The design criterions are assumed to be :

i) Performance

ii) Stability

iii) Security or integrity, the maintenance of stability in face of component failure

iv) Low interaction.

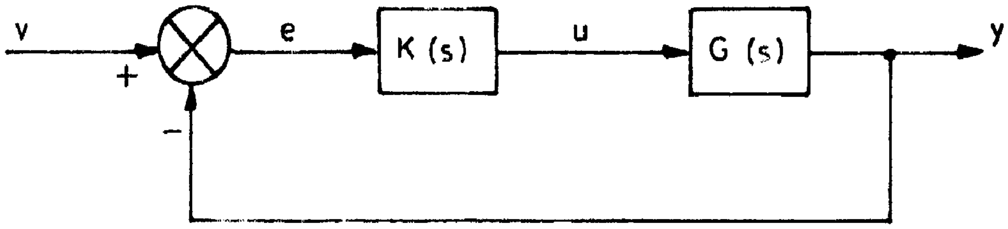


FIG.1.1 MULTIVARIABLE CONTROL SYSTEM.

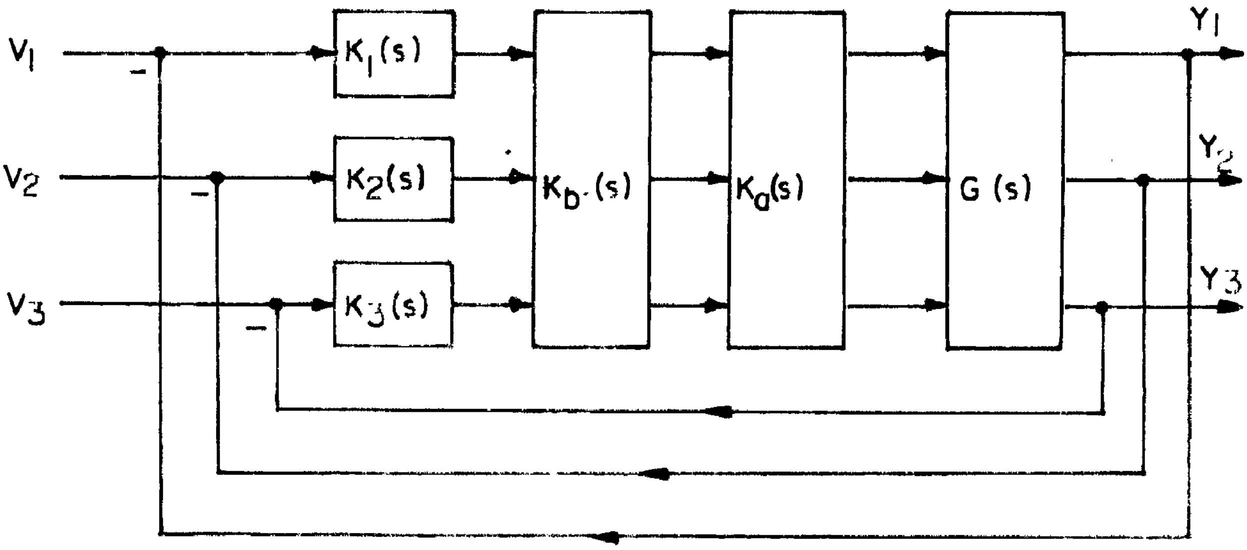


FIG.1.2 STRUCTURE OF THE CONTROLLER.

Stability and performance are basic criteria. Security is achieved in practice by a variety of means e.g. switching to alternative controller. If any component fails, stability may have to be achieved at the expense of performence. Hence a design method which ensures stability in the event of the failure of any specified combinations of N components, should also be flexible enough to include the case N =0. Last is interaction which at low frequencies is automatically reduced in high performance systems, and may not be important at high frequencies, so that, like security, it may not be an important factor in some designs. Thus in the present case the criterias for stability and performance have only been taken into consideration.

1.4. STRUCTURE OF THE CONTROLLER:

The plant has the mxm transfer function matrix G(S), and the controller is represented by an mxm matrix K(S). The object is to find a suitable matrix K(S) which will ensure that the closed loop system meets certain performance specifications.

It will be assumed that the elements of G(S) and K(S) are rational polynomial functions of S, and that neither G(S) nor K (S) is identically zero. It will also be assumed that all the zeros of K (S) are in the open left half plane, because the right half plane zeros in K (S) G(S) give rise to control difficulties, so that there will be no incentive to introduce them in K(S). Finally it is assumed that the plant from which G(S) arises is asymptotically stable before control is applied, and that K(S) has all its poles in the open left half plane.

Since the objective is to design a suitable controller K(S), it is desirable to know what structure is adequate to describe a general K(S). Any such K(S) can be written as a product.

$$K(S) = K_a K_b(S) K_c(S)$$

where three matrices K_a, $K_b(S)$ and $K_c(S)$ has the following properties.

The matrix K_a is a permutation matrix. It therefore represents a preliminary renumbering of the inputs to G(S), which usually will be done so that the new input i affects chiefly the output i.

The matrix $K_b(S)$ has determinant $K_b(S) = 1$ and represents a sequence of elementry column operations. Each such operation consists of adding, to column j of Q(S) operated on, a multiple ofdij (S) by column i. Here dij (S) is a rational polynomial having all its zeros in left half plane. And where $Q(S) = G(S) K(S)$, $K_b(S)$ is used only to make the plant transfer function diagonally dominant or it accomplishes a modification of the interaction in the plant.

The matrix $K_c(S)$ is diagonal and its nonzero entries have all their poles and zeros in the open left half plane. $K_c(S)$ may be written as

$$K_c(S) = \text{diag}(\ K_i\ \ (S)\ )$$

$K_c(S)$ represents m independent controllers. The m loops which contain the K_i (S) will be called the m principal loops. The importance of the decomposition of K(S) into K_a, K_b and K_c is that the successive application of K_a, K_b and K_c is sufficient to generate the most general K satisfying the conditions on K. The structure is given in fig.1.2.

In fig.1.2 K_1, K_2 and K_3 are 3 independent controllers. $K_b(S)$ is to make G(S) diagonally dominant and K_a is for renumbering the inputs.

1.5. SRD AND INA METHODS: A COMPARISON :

Inverse Nyquist Array method for the design of linear multivariable systems was given by H.H. ROSENBROCK in his pioneering paper (2). This method was perhaps the first to design the multivariable systems with classical techniques having very much satisfactory results. In this method, it is necessary to make the plant transfer function diagonally dominant. Sequential return difference method given by Dr. D.Q. Mayne has some modifications over the work of ROSENBROCK and has generated a new algorithm for the design of multivariable systems. SRD method has its supermacy in the sense that diagonally dominance is not necessarily needed. Thus for employing the SRD method one is not supposed to choose $K_b(S)$ shown in fig.2.1. K_a is dependent of number of loops closed. Thus SRD method needs only the proper selection of K_i (S).

The distinguishing feature of SRD method is that it calculates at the ith iteration, the exact modified transfer function of the system with the previous i-1 loops closed. If stability and disturbance attention are the sole design requirements, the multivariable controller can be synthesised by this method with good results. In SRD method compensator may have to be designed, if any diagonal element of the modified transfer function matrix is nonminimal phase i.e. it has zeros in the right half plane. Compensator is to cancel the zeros of right half plane, and the determinant of compensating matrix must be unity.

CHAPTER - II

RETURN DIFFERENCE AND RETURN RATIO MATRICES AND THEIR UTILITY IN THE DESIGN OF CONTROL SYSTEMS

The concepts of return difference and return ratio are shown to play a fundamental role in the analysis and synthesis of multivariable feedback control systems. Matrix transfer functions are regarded as operators over the field of rational functions in the complex variables. The eigen values of such operators are identified as characteristic transfer functions. The corresponding characteristic frequency responses provide a simple and natural link between classical single loop design techniques and multivariable system feedback theory. These concepts serve as a unifying thread in a coherent and systematic discussion of multivariable feedback system design techniques. Moreover these concepts have made the analysis and synthesis very simple.

2.1. RETURN DIFFERENCE MATRIX

From fig.2.1.

r(S) = mx1 matrix of reference input transforms

e(S) = mx1 matrix of error transforms

y(S) = mx1 matrix of plant output transforms

u(S) = rx1 matrix of plant input transforms

K(S) = rxm matrix of controller transfer functions

G(S) = mxr matrix of plant transfer functions

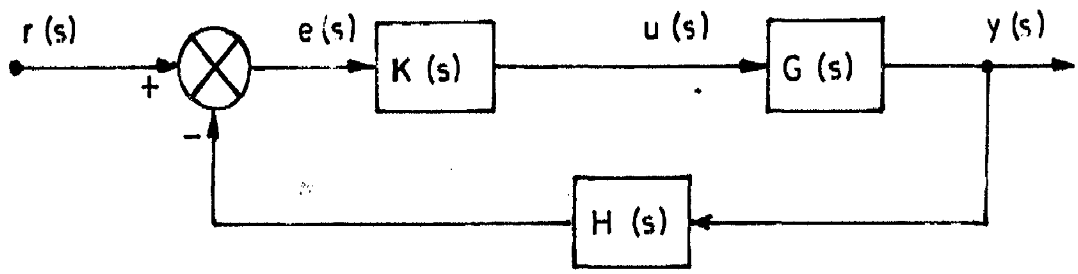


FIG. 2.1

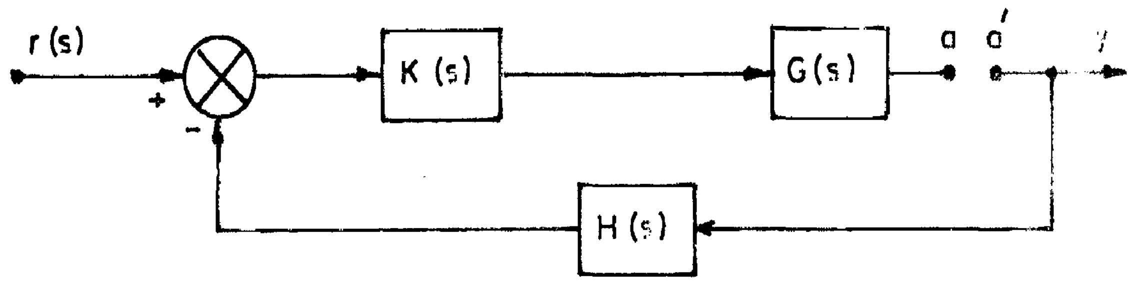


FIG. 2.2

K(S), G(S) and H(S) are matrices over the field of rational. functions in the complex variable S. The closed loop system transfer function matrix is given by

$$R(S) = (I_m + G(S)\ K(S)\ H(S))^{-1}\ G(S)\ K(S) \qquad \ldots (2.1)$$

Now suppose all the feedback loops are broken as shown in fig.2.2. and a signal transform vector a(S) is injected at point a. The transform of the signal returned at a' is then

$$- G(S)\ K(S)\ H(S)\, a\ (S)$$

and the difference between injected and returned signals is thus

$$\left\{ I_m + G(S)\ K(S)\ H(S)\right\} \alpha\ (S) = T\ (S)\, \alpha\ (S) \qquad \ldots(2.2)$$

where $$T(S) = \left\{ I_m + G(S)\ K(S)\ H(S)\right\} \qquad \ldots(2.3)$$

T(S) is defined as the system return difference matrix.

2.2. RETURN RATIO MATRIX :

The matrix

$$F(S) = G(S)\ K(S)\ H\ (S) \qquad \ldots(2.4)$$

is defined as the system return ratio matrix, so that we have

$$T(S) = Im + F(S) \qquad \ldots(2.5)$$

Here T(S) and F(S) both are natural generalisations of the equivalent scalar concept introduced by Bode. Using arguments given by ROSENBROCK, it can be shown that

Let

$$T(S) = \frac{\text{Closed-loop characteristic Polynomial}}{\text{Open-loop characteristic Polynomial}} \qquad \ldots(2.6)$$

The proof of (2.6) is given in the appendix of (4). This is the fundamental equation relating open and closed loop behaviour in multiple loop control systems.

2.3. STABILITY IN TERMS OF RETURN DIFFERENCE MATRIX FOR MULTIPLE LOOP SYSTEMS:

Assume that the system is open loop stable. The open loop characteristic polynomial will then have no zeros in the closed right half complex plane. Thus it follows from equation (2.6) that the closed loop characteristic polynomial will not vanish in the closed right half complex plane if, and only if, det T(S) does not vanish in the closed right half complex plane. Thus it is sufficient to check only T(S) instat of closed loop characteristic polynomial.

2.3.1. Nyquist Type of Criterion:

Let D be a contour in the complex plane consisting of the imaginary axis from -ja to + ja and a semicircle centred at the origin of radius a in the right half plane. Further, let a be large enough to ensure that every zero and pole of det GK(S) and det R(S) which is in the open right half plane lies within D.

Suppose D maps into a close curve T in the complex under the mapping det T(S).

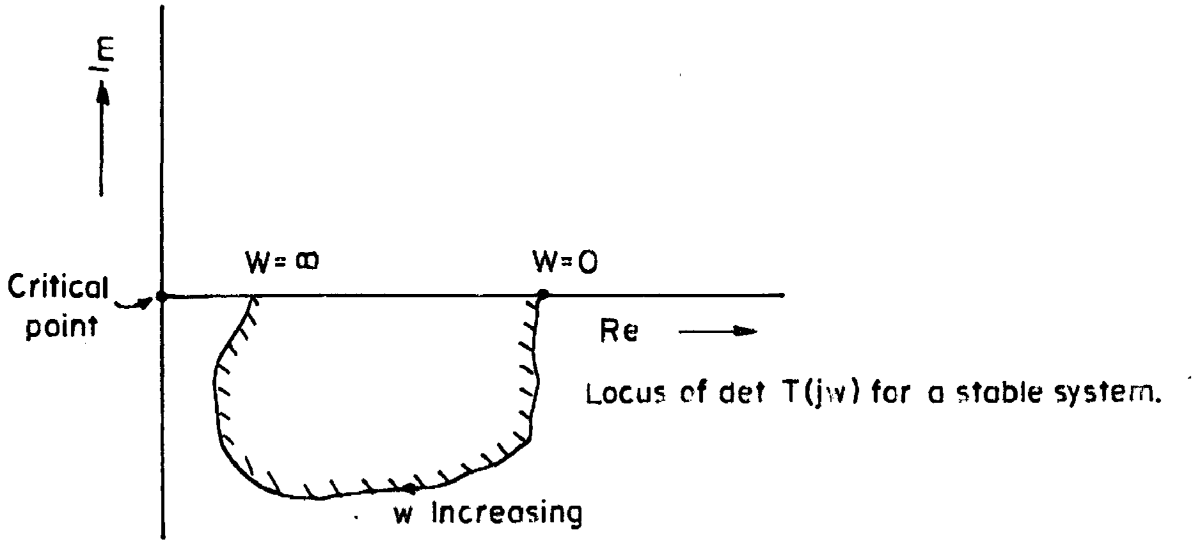


FIG.2.3 SIMPLE MULTIVARIABLE NYQUIST CRITERION

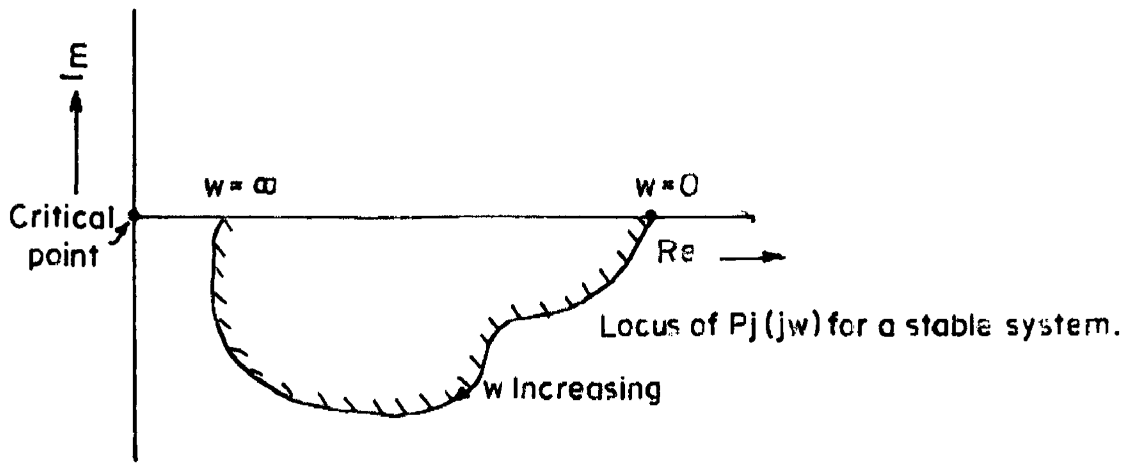


FIG.2.4 EXTENDED NYQUIST CRITERION FOR CHARACTERISTIC FREQUENCY RESPONSE.

Thus the system is closed loop stable if no point within D maps on the origin of the complex plane under the mapping det T(S).

Thus the system is closed loop stable if T does not enclose the origin of the complex plane. If det F(S) $\rightarrow$ 1 as S $\rightarrow \infty$, then, taking as arbitrarily large, we can conveniently refer to T as the locus det F (jw). This gives the multiple-loop Nyquist type of criterion for stability, shown in fig.2.3.

Let the eigen values of T(S) be $P_j(S)$. j=1 ...m we then have that

$$\det T(S) = \prod_{j=1}^{m} P_j(S) \qquad \dots \quad (2.7)$$

therefore, det T(S) will not vanish for any S enclosed by D if none of $P_j(S)$; j =1 ....m vanish for any S enclosed by D. Let D maps into j in the complex plane under $P_j(S)$; j = 1...m , then for a stable system T will not enclose the origin of the complex plane for j=1...m. Thus the system will be stable with all loops if none of enclose the origin of the complex plane for j=1...m.

Fundamental stability property of complex-plane loci of the return difference matrix eigen values can be stated as : The system is closed loop stable if all the eigen value loci P_j (jw) for j=1...m satisfy the Nyquist criterion as illustrated in fig.2.4.

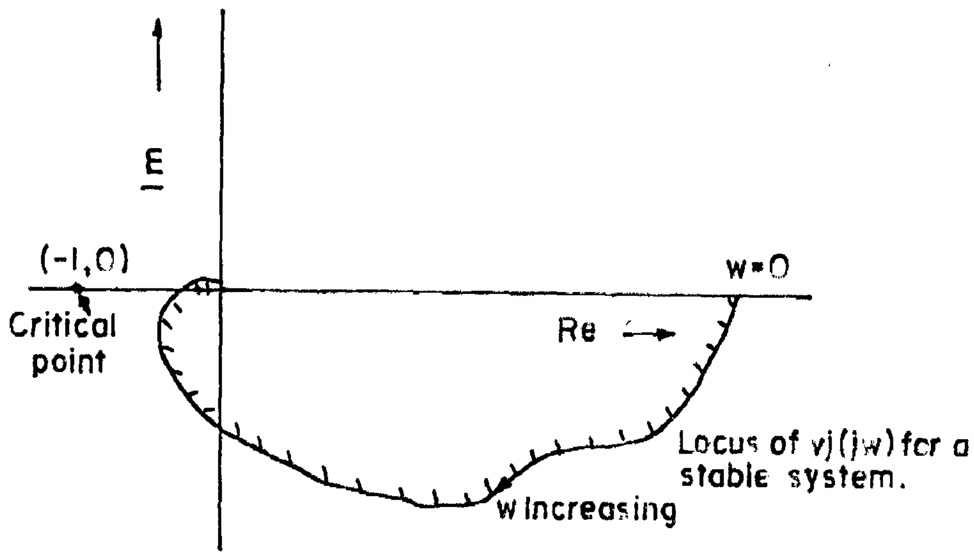


FIG.2.5 CHANGE IN CRITICAL POINT.

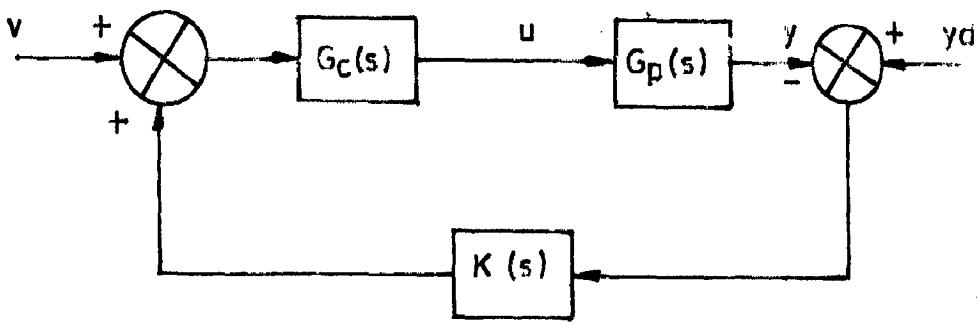


FIG. 3.1 MULTIVARIABLE CONTROL SYSTEM.

This criterion can equally well be stated in terms of the return ratio matrix. Since

$$T(S) = I_m + F(S)$$

The eigen values of T(S) and F(S) are simply related via the eigen value shift theorem. This shows that if $V_j(S) : j = 1 \ldots m$ are the eigen values of F(S), then

$$P_j = 1 + V_j\,(S) \qquad j = 1 \ldots m \qquad (2.8)$$

In terms of the return ratio matrix, therefore, we simply get a unit shift in the location of the critical point. The system is closed loop stable if all the eigen value loci $V_j\,(jw)$ for $j=1,\ldots m$ satisfy the Nyquist criterion as in fig.2.5.

The consideration of the behaviour of det T(S) along the imaginary axis in the complex plane shows that the old criterion for scalar systems is simply extended to the multiple loop case.

2.3.2. Routh Hurwitz Criterion :

In the case of return difference and return ratio matrix, the application of Routh Hurwitz Criterion is very simple. Here if scalar return difference is known before hands, the stability may be checked by the direct application of the criterion for each scalar return difference separately. On the other hands if scalar return ratio is known then the closed loop stability may be decided on the basis of

$$t_j(S) = 1 + f_j(S) \qquad \ldots (2.9)$$

Where t_j = Scalar return difference

f_j = Scalar return ratio

This is useful, when it is difficult to use the Nyquist type of criterion because of computation difficulties.

CHAPTER - III

SEQUENTIAL RETURN DIFFERENCE METHOD FOR THE DESIGN OF LINEAR MULTIVARIABLE SYSTEMS:

The classical frequency methods for designing single loop control systems have proved to be so useful that it is surprising that so little effort has been devoted to extending these techniques to multivariable systems. This may be due to the development of modern control theory, which though originally motivated by open-loop trajectory optimization problems, yielded useful and elegant results for linear multivariable control and filtering problems. The resultant controllers are complex, however, requiring a dynamic filter or observer of almost the same order of complexity as the plant or process being controlled. In order to reduce the complexity of controller ROSENBROCK(2) redried attention to the problem of extending classical procedures to multivariable problems. ROSENBROCK described a useful technique using the Inverse Nyquist Array, for designing multivariable systems. These methods are extended by MACFARALANE (4). Based on the work of (4) a method named as sequential return difference is given by Mayne (6).

Sequential return difference method gives a satisfactory closed- loop linear system via a sequence of single loop designs, in which classical techniques such as Nyquist diagram, root loci, Routh's Criterion etc. are employed.

Stability and performance are the basic criteria for the design. Security is achieved in practice by a variety of means e.g. switching to alternative controller. In the case of component failure, security may have to be achieved at the expense of performance. Interaction at low frequencies is automatically reduced in high performance systems, and may not be important at high frequencies so that, like security it may not be an important factor in some designs. Hence in the sequel, a basic design algorithm, consisting of a sequence of single loop designs, for achieving good performance and stability is described. In this case diagonal dominance is not necessarily required, though it can be employed if it is desired so that increased flexibility in the choice of the compensating matrix $G_c(S)$ is available. Diagonal dominance (2) however automatically provides security against arbitrary, output transducer failure (6) and also limits interaction.

3.1. ASSUMPTIONS AND PROOFS OF RESULTS TO BE USED IN GENERATING THE SRD ALGORITHM :

The system considered is shown in the fig.3.1 The process to be controlled has mxm transfer function $G_p(S)$. $G_c(S)$ is a mxm compensator matrix and :

$$G(S) = G_p(S)\ G_c(S) \qquad \ldots \qquad (3.1)$$

K(S) is a mxm diagonal matrix. Assumptions made are :

(A1) Neither G(S) nor K(S) are identically zero.

The matrix return difference T(S) is defined to be

$$T(S) = I_m + G(S)\,K(S) \quad \ldots (3.2)$$

The system has a state space representation :

$$\dot{x}(t) = AX(t) + Be(t) \quad \ldots (3.3)$$

$$y(t) = CX(t) \quad \ldots (3.4)$$

$$e(t) = y_d(t) - y(t) \quad \ldots (3.5)$$

i.e. $$\dot{X}(t) = (A - BC)\,X(t) + B\,Y_d(t) \quad \ldots (3.6)$$

The expressions

$$Wo(S) = SI - A \quad \ldots (3.7)$$

$$Wc(S) = SI - A + BC \quad \ldots (3.8)$$

are respectively the open loop and closed loop characteristic polynomials.

Let $$\bar{G}(S) = G(S)\;K(S) \quad \ldots (3.9)$$

the loop transfer function, and let R(S) denot the closed loop transfer function relating y(s) to Yd(S). Clearly

$$\bar{G}(S) = C\,(SI - A)^{-1}\,B \quad \ldots (3.10)$$

since

$$T(S)\;\bar{G}(S) = \bar{G}(S)\;T(S) \quad \ldots (3.11)$$

We have

$$R(S) = T^{-1}(S)\;\bar{G}(S) = \bar{G}(S)\;T^{-1}(S) \quad \ldots (3.12)$$

Wo(S) and Wc(S) are related by the following well known result (4).

$$T(S) = W_c(S)\;/\;W_o(S) \quad \ldots (3.13)$$

Now we make the further assumption.

(A2) The, process, with transfer function $G_p(S)$, is open loop asymptoticallystable. Note that the assumption of asymptotic stability is made for simplicity in presentation.

(A3) It is also assumed that $G_c(S)$ and K(S) have poles and zeros in the left half plane only and $|G_c(S)| = 1$ then $W_o(S)$ has zeros in the open left half plane only.

From equation (3.13) it follows that closed loop system is asymptotically stable if and only if the locus Tm of T (jw) does not encircle or pass through the origin. And for reducing interaction, high gain loops are required so as

$$T^{-1}(jw) \rightarrow 1 \quad \text{for} \quad w \in X \qquad ...(3.14)$$

For convenience some extra terms appropriate to the condition when the first j loops are closed and the remaining open, i.e. $K_j(S) = 0$.

$j = j+1 \ldots m$, are defined

$$K_i(S) = \text{diag}(k_1(S), \ldots\ldots k_i(S), 0, 0\ldots) \qquad (3.15)$$

$$T_i(S) = I_m + G(S) \quad K_i(S) \qquad (3.16)$$

$$G_i(S) = T^{-1}(S) \quad G(S) \qquad (3.17)$$

for $=0..m$ clearly $T_0(S) = I_m$, $G_o(S) = G(S)$

$G_i(S)$ is the transfer function relating y to v in fig.3.1. when loop j+1 ...m are open, the scalar return difference

$$t_i(S) = 1 + k_i(S) \quad g_{ij}(S) \qquad (3.18)$$

where $g_{ij}(S)$ is the ijth element of $G^k(S)$.

Let $g_{i\cdot}$, $g_{\cdot i}$ denote respectively, the ith row and ith column of G. Let S_o denote the open loop state space representation (A, B,C) of $G_p(S)\, G_c(S)\, K(S)$ given in equation (3.3) and (3.4). Let S_m denote the closed loop state space representation (A-BC, B,C) given in equations (3.3), (3.4) and (3.5). for i= 1...m, let S_i denote the state space representation ($A-B_i$ C, B_i, C) corresponding to the situation when the first j loops are closed and the remaining are open. Let W_c (S) = SI - A + B C denote the characteristic polynomial of S_i. Now T_i maps D into T_i and t_s maps D into Y_s. Then

$$T_i\,(S) = W_c^i\,(S) \;/\; W_o\,(S)$$

and let N_i, n_i denote respectively the no. of ne encirclements of the origin by T_i and Y_i :

$$N_i = \sum_{j=1}^{i} n_j \tag{3.19}$$

This is from the theorem (2), which is

For i = 1 ...m

$$T_i(S) = \prod_{j=1}^{i} t_j(S) \tag{3.20}$$

Now since W_o (S) has no right half plane roots, the from eq.(3.19) N_i is equal to the number of roots of W_c (S) in the right half plane. Thus the stability theorems are valid for scalar return difference and Scalar return ratios also. The proof (3.20) is in (6).

3.2. SEQUENTIAL RETURN DIFFERENCE ALGORITHM:

From the last discussion a design procedure, currently used in practice, would be to choose k_1 so that $t_1 = 1 + k_1 g_{11}$ is satisfactory, calculate G^1, choose k_2 so that $t_2 = 1 + k_2 g_{22}$ is satisfactory. Calculate G^2 etc. However this procedure ignores the fact that even if $G_p(S)$ has no right half plane zeros, g_{ii}, i=1...m, as obtained above may right half plane zeros. The role of G_c, where $|G_c(S)|=1$ in the basic algorithm which has the objectives; performance and asymptotic stability. These zeros of right half plane give " design difficulty " appropriately to the various loops. The next is a simple sequential method for calculating G and t_i, i =1.....m, which is SRD algorithm. The following algorithm generates G, t_i, i = 1....m.

(i) Set G (S) = G(S)

Set i = 1.

Choose k_1 (S)

(ii) Set $t_i(S) = 1 + k_j(S)\, g_{ii}^{i-1}(S)$

(iii) If i = n, stop otherwise

Set $\overline{K_i}(S) = K_i(S) / t_i(S)$ (3.21)

Set $G(S) = G^{i-1}(S) - \overline{K_i}(S)\, g_{\cdot i-1}^{i-1}(S)\, g_{i\cdot}^{i-1}(S)$ (3.22)

Set = i = i + 1

(iv) GO TO (ii)

Proof. For equation (3.22)

$$T_i(S) = T_{i-1}(S) + K_i(S)\, g_{\cdot i}(S)\, g_{i\cdot}(S)$$
$$= T_{i-1}(S) \quad I_m + K_i(S)\, g_{\cdot i}^{i-1}(S)\, g_{i\cdot}^{i-1}(S)$$

Hence using a well known identity to invert the term in brackets (6).

$$T^{-i}(S)\, G(S) = \left[I_m - \overline{K_i}(S)\, g_{\cdot i}^{i-1}(S)\, g_{i\cdot}^{i-1}(S) \right] G^{i-1}(S) \quad (3.23)$$

$$G(S) = T^{-1}\, G(S)$$

Thus $G(S) = G^{i-1}(S) - \overline{K_i}(S)\, g_{\cdot i}^{i-1}(S)\, g_{i\cdot}^{i-1}(S)$

3.3. MODIFICATIONS MADE IN THE SRD ALGORITHM:

In the last step of sequential return difference algorithm G (S) is calculated. G (S) is the modified transfer function when first i loops are closed. If i is unity i.e. only one loop is closed, then only first diagonal element of transfer function G(S) is going to be change. And if i is two, then only second diagonal element will change, and same will be the case for other values of i. So for reducing the computational complexities, only diagonal elements of the transfer function G(S), which are modified are calculated. After this is done, another modification is that for closing each loop, n values of controller transfer function are assumed and corresponding scalar return differences and modified diagonal elements are calculated. If i =1 ....n, the scalar return difference is taken as t (i, J), and in the same way the other variables. Thus the modified algorithm becomes:

The following algorithm generates $t_{i,j}^{i}(S)$, $G_{i,j}(S)$, $i=1...m$, $J = 1...n$

(i) Set $G_{i-1,J}(S) = G(S)$

Set $i = 1$

(ii) Choose $k_{i,j}(S)$ $J = 1 \ldots n$

Set $t_{i,J}(S) = 1 + 1_{i,j}(S) \cdot g_{ij}(S)$, $J = 1 \ldots n$

(iii) Set $\overline{K_{i,j}}(S) = k_{i,j}(S) / t_{i,j}(S)$, $j = 1 \ldots n$

Set $g^{i}_{ii}(S) = g_{ii} - \overline{K_{i,j}}(S)\, g_{i}(S)\, g_{i}(S)$, $J = 1 \ldots m$

Set $G_{i,j}(S) = G(S)$ with $g_{ii}(S)$ as $g_{ii}(S)$,

$j = 1 \ldots n$

(iv) If $i = m$ stop otherwise

Set $i = i + 1$

(v) GO TO (ii)

For the development of computer program the elements of the transfer function G(S) are named as $G_{11}N$ (I), $G_{11}D(I)$, $G_{12}N(I)$, $G_{12}D$ (I) .... etc. This is done because the elements of the transfer function are polynomials in S, so their coefficient arrays in descending order are taken for computations. Controller transfer functions are also polynomials in S, thus it becomes necessary to read them as KN (L, I) and KD (L, I), L stands for the number of values to be taken into account for computations, I is for the number of coefficient in descending order for a particular value of controller transfer function. N and D are for numerator and denominator.

CHAPTER - IV

DESIGN OF REACTOR CONTROLLER

The state equations of the Nuclear reactor are given. For using sequential return difference algorithm, the given state equations are first converted into the matrix transfer function. Matrix transfer function is used in the algorithm described in 3.3. Thus the values of scalar return differences and the modified matrix transfer functions are achieved. Scalar return differences are used to check the stability. Out of the stable set of systems under consideration, few are checked for transient responses to step inputs.

4.1. TRANSFER FUNCTION OF THE NUCLEAR REACTOR :

The state equations for the nuclear reactor are given as :

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X}_3 \end{bmatrix} = \begin{bmatrix} -0.188 & 0.0 & 0.227 \\ 0.0 & 0.0 & 1.0 \\ -2.138 & -0.587 & -0.550 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix}$$

State vectors, X_1, X_2 and X_3 are :

X_1 = E.m.f.

X_2 = rate of change of rotor angle

X_3 = rate of change of angular velocity

u_1, u_2 and u_3 are step inputs.

in the above representation

$$A = \begin{bmatrix} -0.188 & 0.0 & 0.227 \\ 0.0 & 0.0 & 1.0 \\ -2.138 & -0.587 & -0.550 \end{bmatrix} \quad B = \begin{bmatrix} 1.0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Plant transfer function $G_p(S) = (SI-A)^{-1} B$

$G(S) = G_p(S) \ G_c(S)$

Let $G_c(S) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Then, G(S) is calculated as :

$G_{11}N = S^2+0.550S+0.587$, $G_{11}D = S^3+0.738S^2+1.1742S+0.111$

$G_{12}N = -0.133$ $G_{12}D = S^3+0.738S^2+1.1742S+0.111$

$G_{13}N = 0.227\ S$ $G_{13}D = S^3+0.738S^2+1.1742S+0.111$

$G_{21}N = 2.138$ $G_{21}D = S^3+0.738S^2+1.1742S+0.111$

$G_{22}N = S^2+0.738S+0.5872$ $G_{22}D = S^3+0.738S^2+1.1742S+0.111$

$G_{23}N = S+0.188$ $G_{23}D = S^3+0.738S^2+1.1742S+0.111$

$G_{31}N = 2.138$ $G_{31}D = S^3+0.738S^2+1.1742S+0.111$

$G_{32}N = -0.587(S+0.188)$ $G_{32}D = S^3+0.738S^2+1.1742S+0.111$

$G_{33}N = S(S+0.1888)$ $G_{33}D = S^3+0.738S^2+1.1742S+0.111$

4.2. CONTROLLER TRANSFER FUNCTIONS ASSUMED AND RESULTING SYSTEMS:

For controller transfer functions, five values for each feed back loop have been assumed. The table is given below :

Table 4.1.

Loops	1	2	3	4	5
K_1	5/S	70	20	10	5
K_2	"	"	"	"	"
K_3	"	"	"	"	"

These values are given in the form KN(L,I) and KD(L,I), I=1..2, L = 1...15. The computer program for sequential return difference algorithm is given in Appendix A. Thus using above fifteen values we may arrive at many controllers, as three values are to be chosen for one controller. These controllers have to be checked for stability.

4.3. FINDING OUT THE STABILITY OF RESULTING SYSTEMS USING SCALAR RETURN DIFFERENCE:

From the results of computer program given in Appendix-A, the values of scalar return differences are found. These scalar return differences are used to check the stability by Routh Hurwitz criterion. The Sub-Routine for Routh's criterion is given in Appendix-A. For ensuring the tight feedback, a computer program is given in Appendix-B. In checking the feedback, again scalar return difference has been utilized which has been achieved from the results of sequential return difference algorithm.

Table 4.2

Sl. No.	A. WHEN FIRST FEEDBACK LOOP IS CLOSED		
	CONTROLLER TRANSFER FUNCTION	STABILITY	NATURE OF FEEDBACK
1.	5/S	UNSTABLE	TIGHT
2.	70	STABLE	TIGHT
3.	20	STABLE	TIGHT
4.	10	STABLE	TIGHT
5.	5	STABLE	TIGHT
Sl. No.	B. WHEN SECOND LOOP IS CLOSED		
	SAME	SAME	SAME
1 to 5	SAME	SAME	SAME
Sl. No.	C. WHEN THIRD LOOP IS CLOSED		
	SAME	SAME	SAME
1 to 5	SAME	SAME	SAME

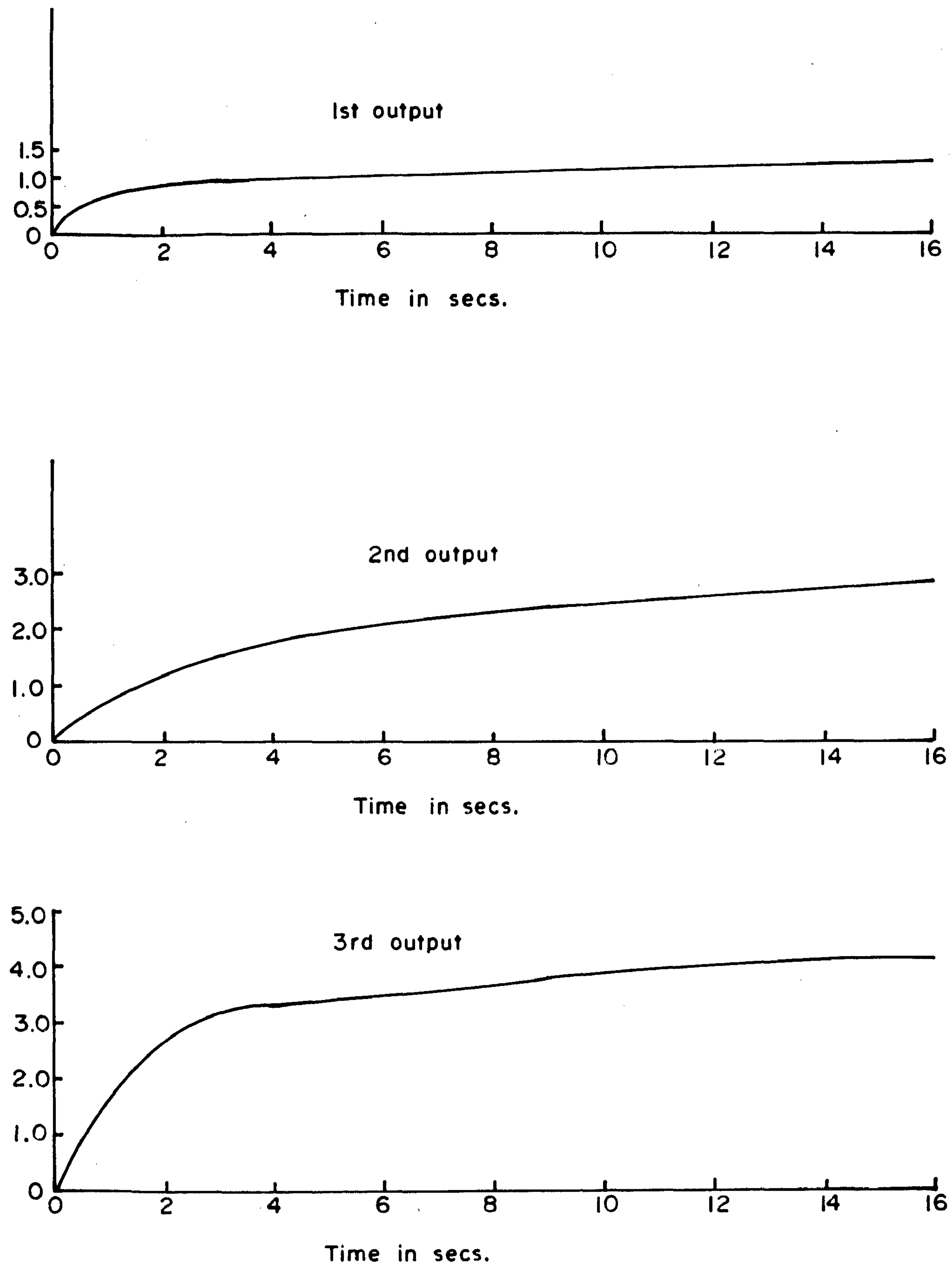
From Table 4.2. it is clear that out of fifteen chosen values of controllers transfer functions, the system is stable only for twelve values. Thus we are getting four controller transfer functions for each feedback loop for a stable system.

4.4. PERFORMANCE

Having found out the values of controller transfer functions for which the system is stable, it becomes necessary to check which one gives good performance and with which feedback loop. Here, as in the present problem the order of the modified elements of modified transfer function matrix rises too much. Because of this rise in the order, it is very difficult to check the transient response of all the systems with modified transfer functions. So, here the Nyquist plot of all the systems are given. Thus relative stability is giving the measure of better performance. Program for Nyquist criterion is given in Appendix-C.

4.4.1. Step Response Calculations :

The values of controller transfer functions for each feedback loop giving best performance out of the choosen values of controller transfer functions for which system is stable, are K (1,2), K (2,2) and K (3,3). The respective modified matrix transfer functions may be achieved from the results of the program given in Appendix-A. These becomes the data for the program of step response given in Appendix-D. As the result of step response, three polynomials of around thirty three order numerator and denominator are obtained. For comparison in transient



IG. 5.7 TIME RESPONSES WITHOUT FEEDBACK.

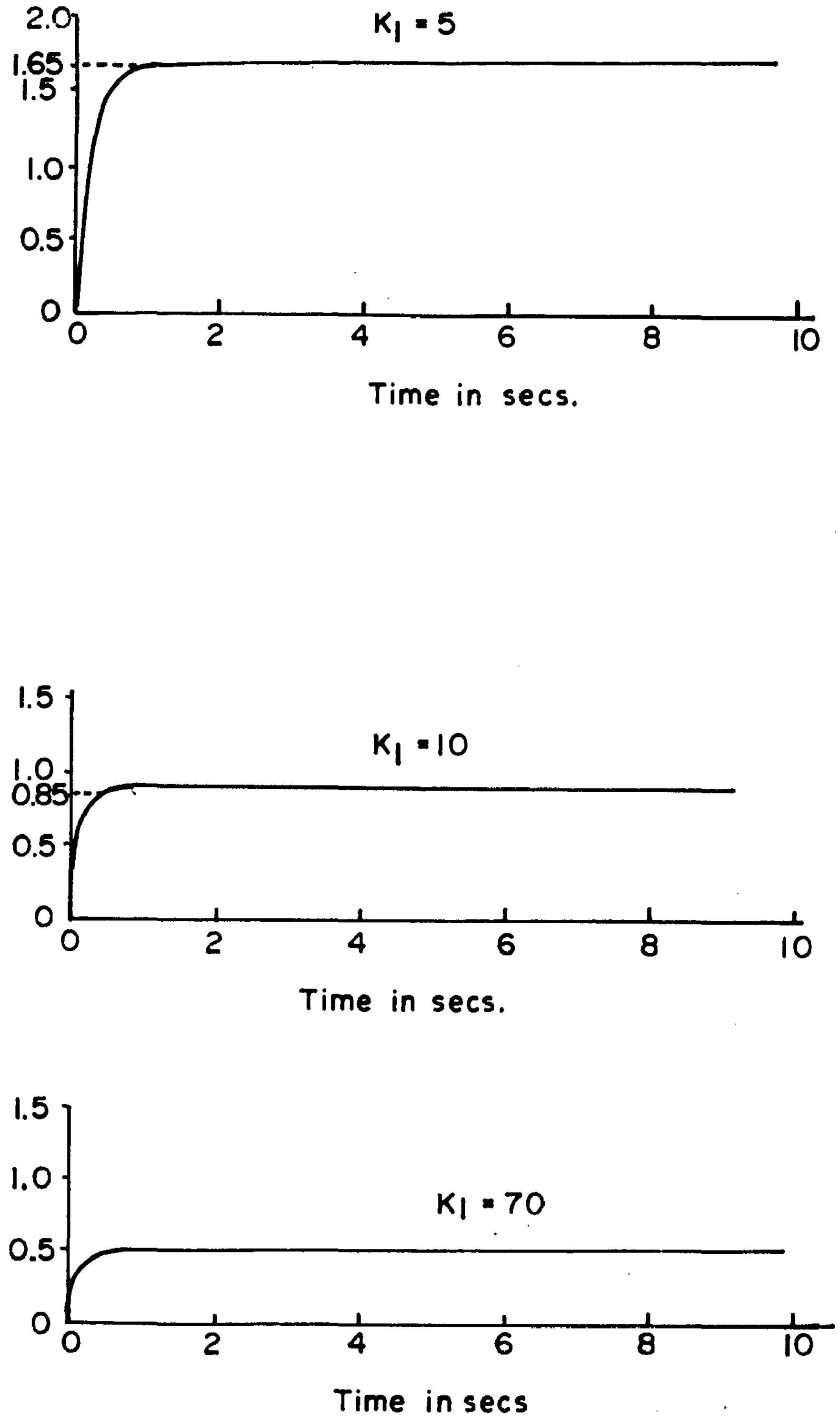


FIG.5.4 TIME RESPONSES FOR FIRST LOOP CLOSED.

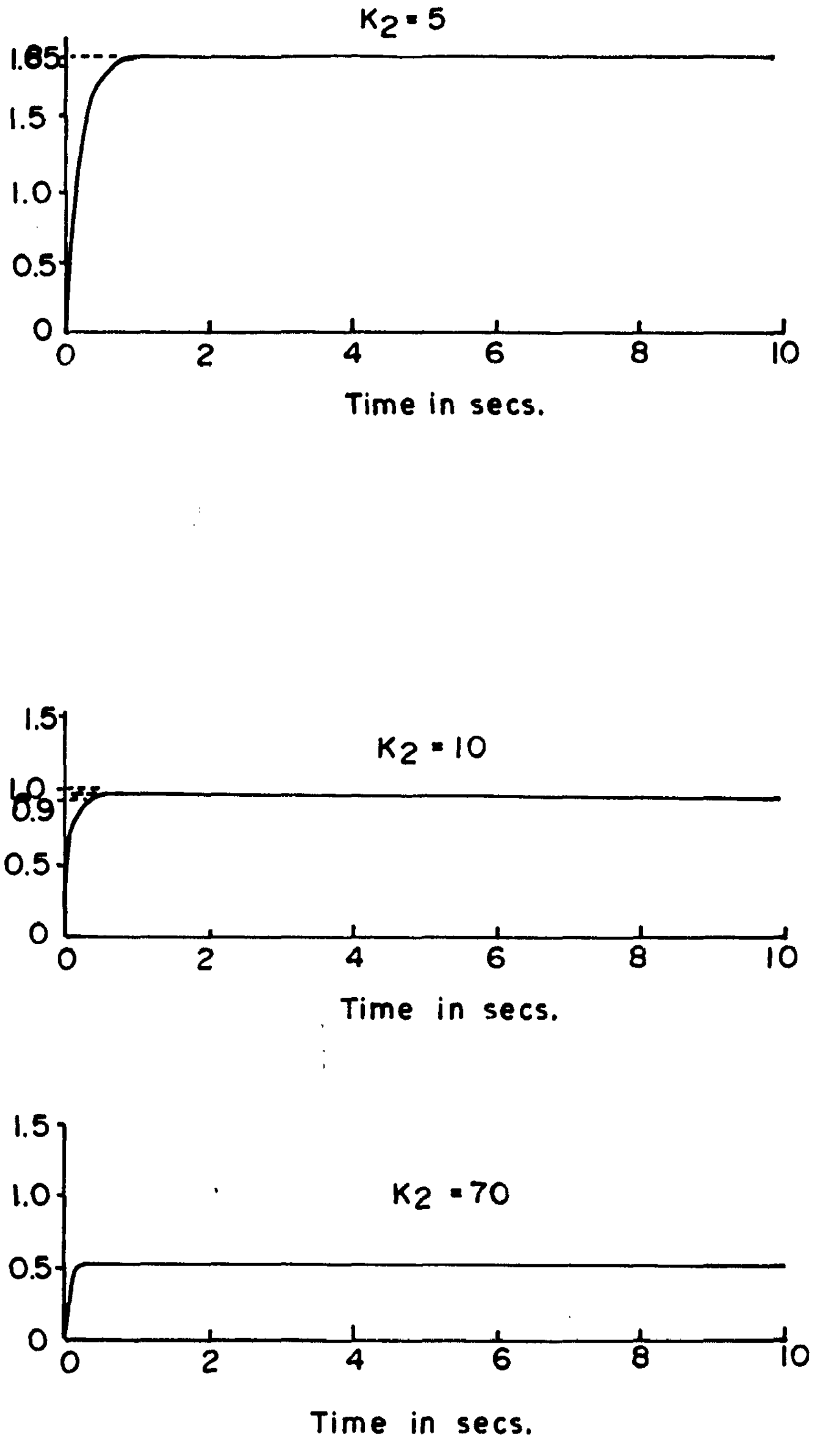


FIG.5.5 TIME RESPONSES FOR SECOND LOOP CLOSED.

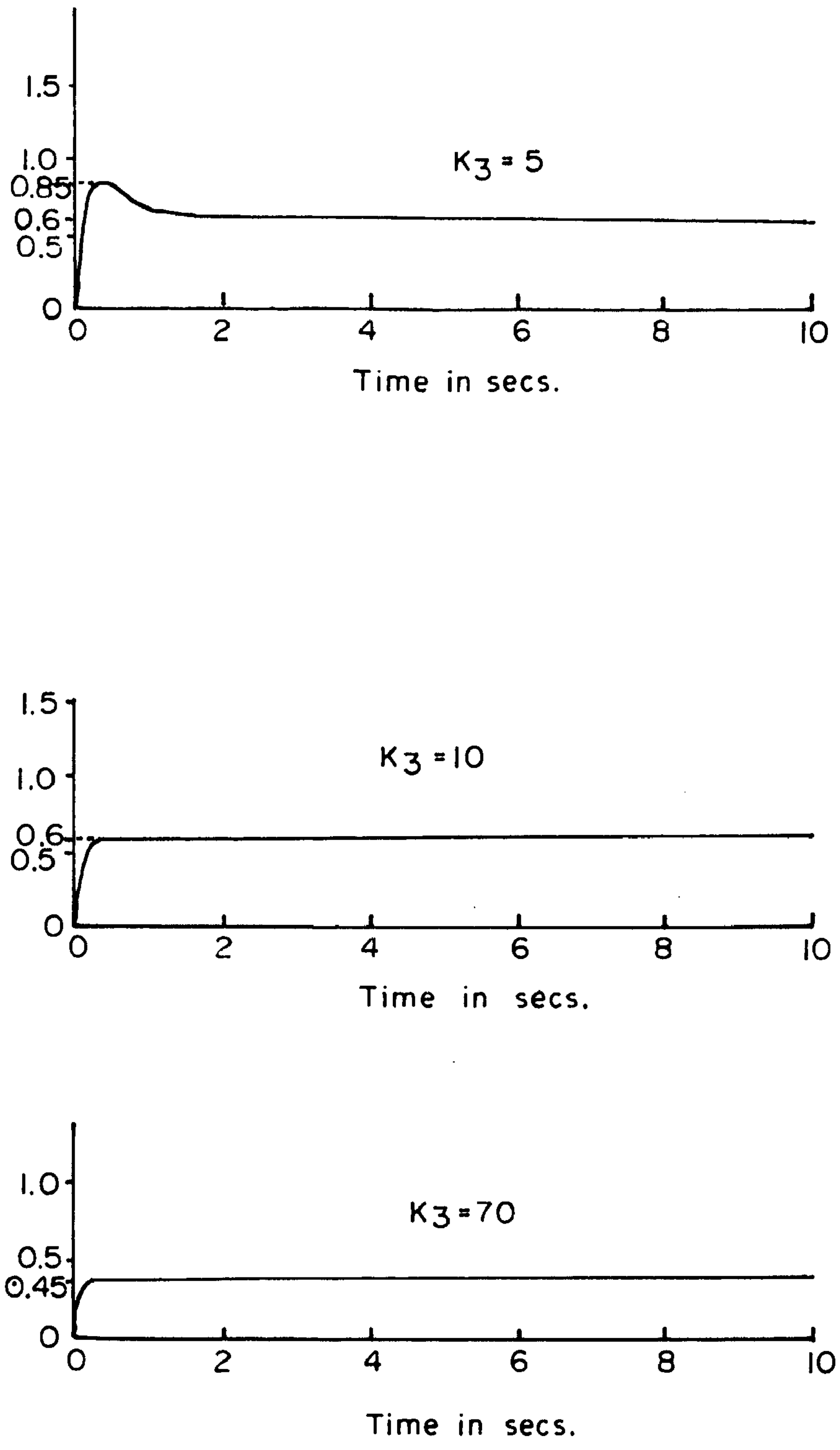
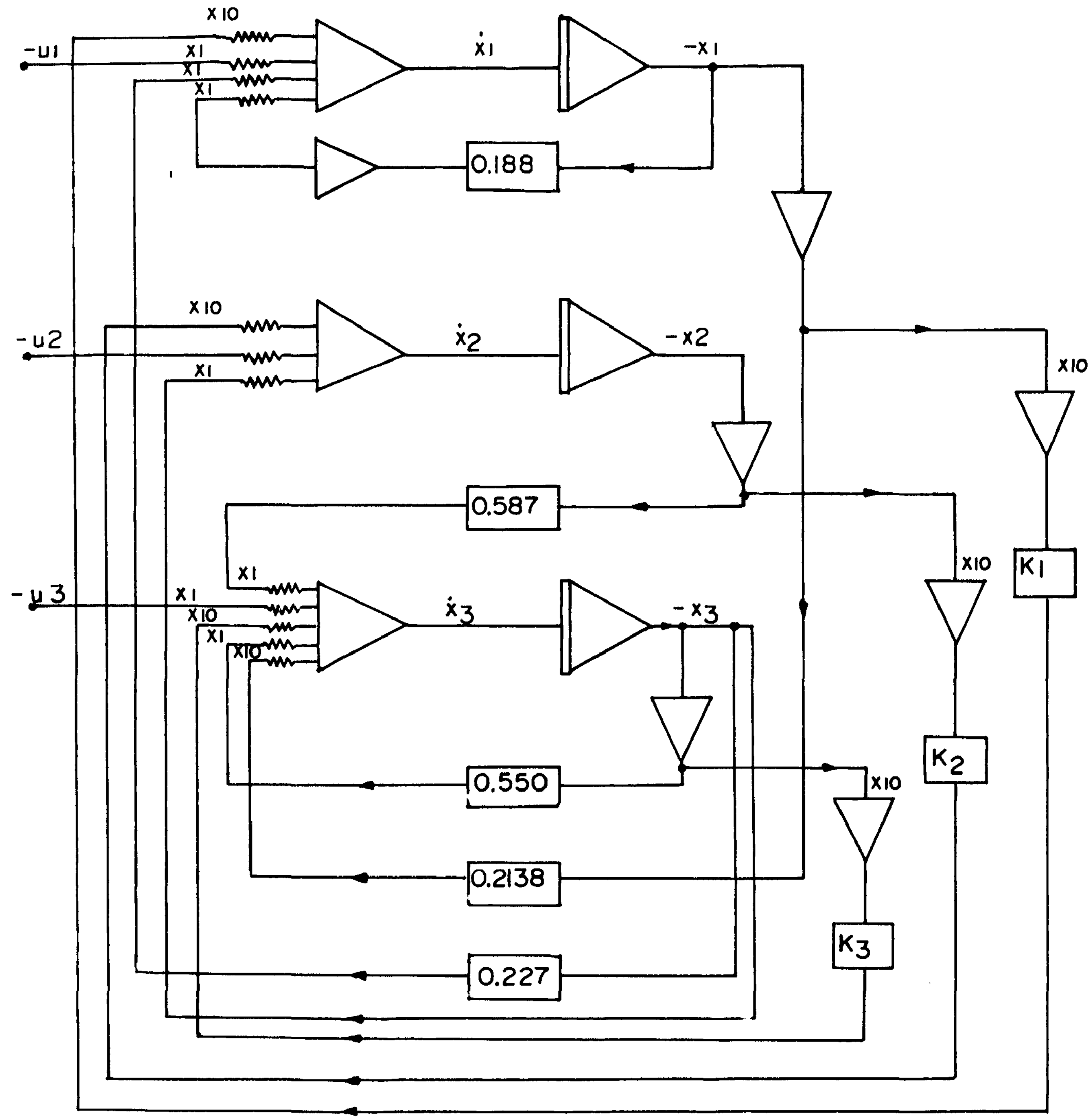


FIG.5.6 TIME RESPONSES FOR THIRD LOOP CLOSED.



-u1
-u2
-u3
X10
X1
$\dot{x}_1$
$-x_1$
$\dot{x}_2$
$-x_2$
$\dot{x}_3$
$-x_3$
0.188
0.587
0.550
0.2138
0.227
K_1
K_2
K_3

FIG.5.8

responses, the controller transfer functions K(1,3), K(2,3) and K(3,3) are also considered for step responses.

4.4.2. Partial Fractions of the Step Response :

Before going to partial fractions, all the roots of the denominator of each response are determined. The program for roots is given in Appendix-E. These roots and the numerators of the responses are used in the program for partial fractions given in Appendix-F.

4.4.3. Transient Response :

The roots of the denominator of each response and their respective residues are the data for the program of "Laplace inverse and time response generation. This program is developed to give three sets of transient responses and each set giving two curves (for two values of controller gain) for one output. The open-loop response of the system with step-input is calculated and is shown in fig.5.7.

The closed-loop response of the plant is found for different controller settings, with the help of Analog Computer AC 20. The reason for using the Analog Computer is the unusual high time required by the digital computer in finding the roots of the high-order polynomials that are generated in the SRD algorithm.

As seen from the closed-loop response from Figs.5.4,5.5,5.6, the response is very fast with no overshoot (compared with the sluggish response of the open-loop system). It must be, however, emphasised that the closed loop response has large steady-state error. This is due to using very simple controller dynamics i.e. a single gain term only. With a single gain term in the feed back path such high values of steady-state errors are expected. To do away with this error in the response obviously one has to include further terms added in the feedback like integral terms and derivative terms. This was not tried in the present work only due to non-availability of a fast digital computer.

The closed-loop response was also found using the computer program developed. However, these results show high magnitudes (of the order of 10^3) for a step input. The reason is that on the application of SRD algorithm, the order of the modified diagonal elements rises upto 27. As a result some zeros of the modified diagonal elements happen to be in the right half plane and makes the element non-minimum phase. This is a disadvantage which is inherent with this algorithm.

These right half plane zeros of the modified diagonal elements may be cancelled by proper selection of compensating matrix $G_c(S)$. The value $G_c(S)$ should present case, the selection of $G_c(S)$ becomes difficult. Here, the order of modified diagonal elements is very high, and a number of zeros are in the right half plane. For cancelation of these zeros, the structure of compensating matrix will also be complex. A method to systematically determine $G_c(S)$ is given in 4.5.

4.5. A NEW GENERALIZED ALGORITHM FOR DESIGNING THE COMPENSATOR- $G_c(S)$:

For designing the compensator first one has to know the number of zeros coming in the right half plane, of the diagonal elements of modified transfer function matrix. In the present work the order of the numerator and denominator polynomials is about 27. IBM 1620 takes 25 to 40 minutes for finding out the real and imaginary roots of such a polynomial under the accuracy of 0.01. First consider the case of a three input three output system. If first diagonal element has the zeros A and B in right half plane, second diagonal element has the zeros C and D in right half plane and the third diagonal element has the zeros E and F in right half plane; then compensator has to be designed such that, it should have the poles at the locations of zeros in the respective diagonal element. The necessary condition is that the $G_c(S) = 1$. The possible compensator for the present problem w hat the author could think of is

$$G_c(S) = \begin{bmatrix} \frac{1}{(S-A)(S-B)} & 1 & 0 \\ 0 & \frac{1}{(S-C)(S-D)} & 1 \\ -1 & \frac{1}{(S-C)(S-D)(S-E)(S-F)} & \frac{1}{(S-E)(S-F)} \end{bmatrix}$$

then $G_c(S) = 1$

This is the case of three input three output system and with two zeros of each diagonal element in the right half plane. But in actual cases the problem may be some what different. Thus, there should be a generalized method for the design of compensator. A possible generalized matrix for the design of compensator is presented below :

Let A1(I) = No.of pole locations for first diagonal element of the compensator

AN(I) = No. of pole locations for Nth diagonal element

$$X(J) = \prod_{I=1}^{IZ(i)} (S-A\,(J,I))$$

Where

J =1 ... N

N is the number of inputs and outputs

IZ(i) is the number of zeros in the right half plane for j+h diagonal element

Thus

$$X(1) = \prod_{I=1}^{IZ(1)} (S-A(1,I))$$

$$X(2) = \prod_{I=1}^{IZ(2)} (S-A(2,I))$$

$$\vdots$$

$$X(N) = \prod_{I=1}^{IZ(N)} (S-A(N,I))$$

Now compensator may be given as

$$G_c(S) = \begin{bmatrix} \frac{1}{X(1)} & 1 & 0 & \cdots\cdots\cdots 0 & 0 \\ 0 & \frac{1}{X(2)} & 1 & 0 \cdots\cdots\cdots 0 & 0 \\ 0 & 0 & \frac{1}{X(3)} & 1 \quad 0 \cdots\cdots 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \frac{1}{X(N-1)} & 1 \\ -1 & 0 & 0 & \frac{1}{X(N-1).X(N)} & \frac{1}{X(N)} \end{bmatrix}$$

$G_c(S) = 1$

mathematically

$$G_c(j,j) = 1/X(j), \quad j = 1,N$$

$$G_c(j,j+1) = 1 \quad , \quad j = 1,N-1$$

$$G_c(N,1) = -1$$

$$G_c(N,N-1) = 1/X(N).\ X(N-1)$$

Thus, obtained compensating matrix is to be multiplied with modified transfer function matrix to provide phase minimality to the diagonal elements of the modified transfer function matrix.

But in the present case the order of the numerator and denominator is very high, and computer involved in the work being IBM 1620 which is very slow. Thus lot of computer time at this computer will be wasted only in checking the right half plane zeros. After that again finding the roots of the denominators of compensated diagonal elements and finding difficult to get the transient responses after compensation with this computer. The method is very useful after employing the algebraic operations for reducing the order of the polynomials which is at research level upto now. So performance is decided on the basis of relative stability by NYQUIST.

CHAPTER- V

RESULTS AND DISCUSSION

The results of sequential return difference algorithm and of other programs are given in the end. For the values of controller transfer functions with which the system is stable, Nyquist plots are given. Three sets of plots are for three output of the reactor, and each set is having four curves for different values of controller transfer functions. The best value of controller transfer function from each set of four curves is selected on the basis of relative stability.

These zeros of modified diagonal element coming in the right half plane introduce " design difficulty " which comes across only at the time of checking the transient response.

5.1. BRIEF INTRODUCTION TO COMPUTER PROGRAMS

First program is for sequential return difference method where it is made to read the original transfer function matrix element wise i.e. first it will read numerator of first element then denominator of the same and proceeds to the next element of first row, and same is for the remaining rows. Controller transfer functions are read in the same way. Subroutine of Routh Hurwitz criterion is called in the program to check the stability. Subroutine takes the return difference polynomial in the array of the coefficients of a polynomial in descending order. If some starting coefficients of the array are zero, then arrangement to remove them and to reduce the order accordingly is made in the subroutine. After algebric operations employment, the program will work with little modifications according to the operations.

Ensuring the tight feedback is second program where numerators and denominators of scalar return difference may be given in arrays A(I) and B (I). Arrangement for removal of initial zeros in the arrays is made here also.

Nyquist plot program also excepts the arrays as in the program of tight feedback. Here if some wants to get polar value i.e. r and Q, he can have by getting punched XM and XA.

For step response, first the modified diagonal elements will be supplied, then the remaining elements of the transfer function matrix. Subroutine PMULT is for polynomial multiplication and subroutine PSUM is for polynomial summation. This program gives in first iteration numerator, denominator for all the three outputs. And again in the second iteration gives all the things for second set. Note that this program reads matrix transfer function elements except the diagonal elements in another way. Such as for any element it will one coefficient from the numerator coefficients array and then one from denominator coefficients array and so on.

In the case of finding the roots, Bairstraw method has been employed. Here also you **give** the polynomial array in descending order.

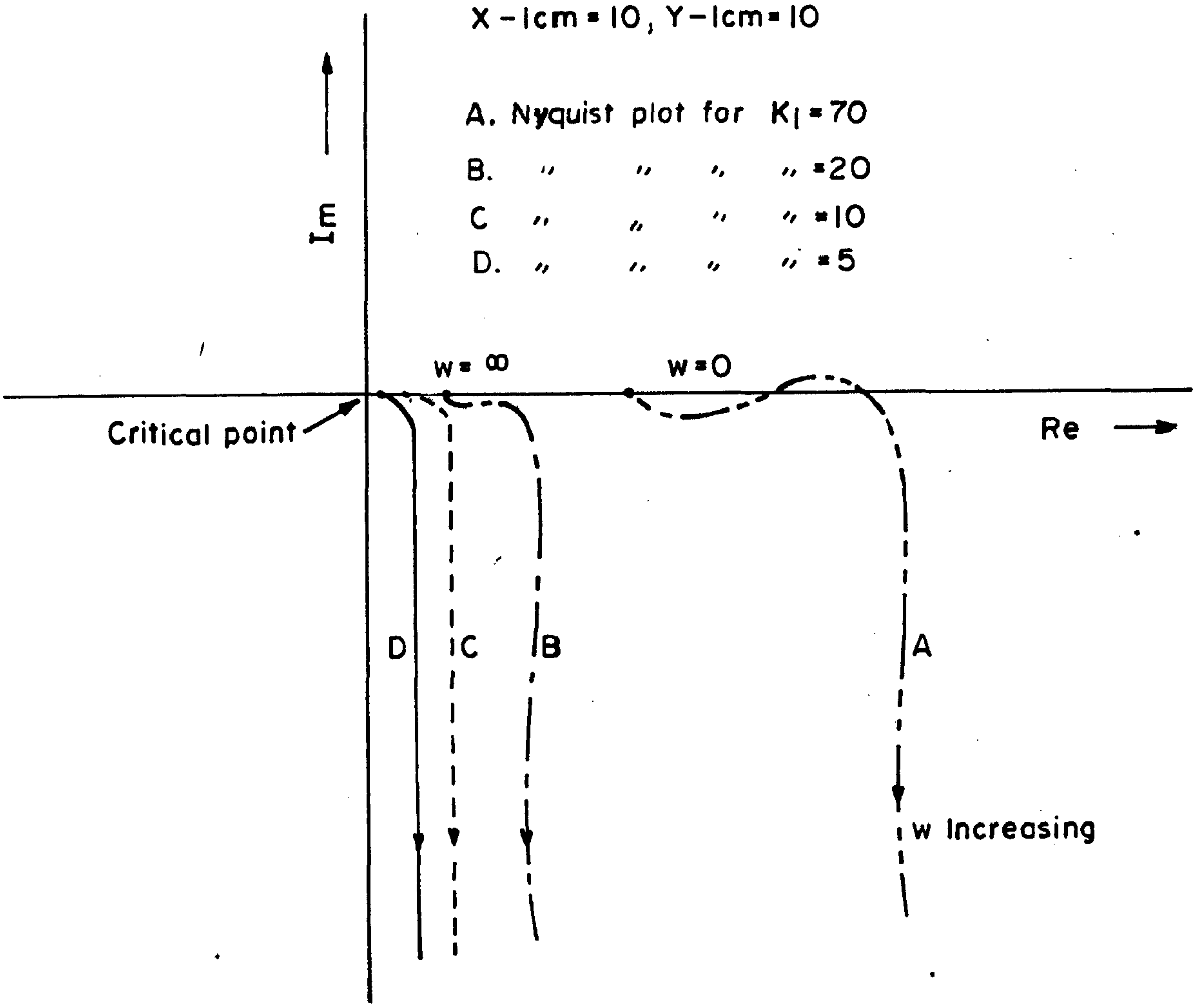


FIG.5.1 NYQUIST PLOTS FOR DIFFERENT VALUES OF CONTROLLER TRANSFER FUNCTIONS WHEN Ist LOOP IS CLOSED.

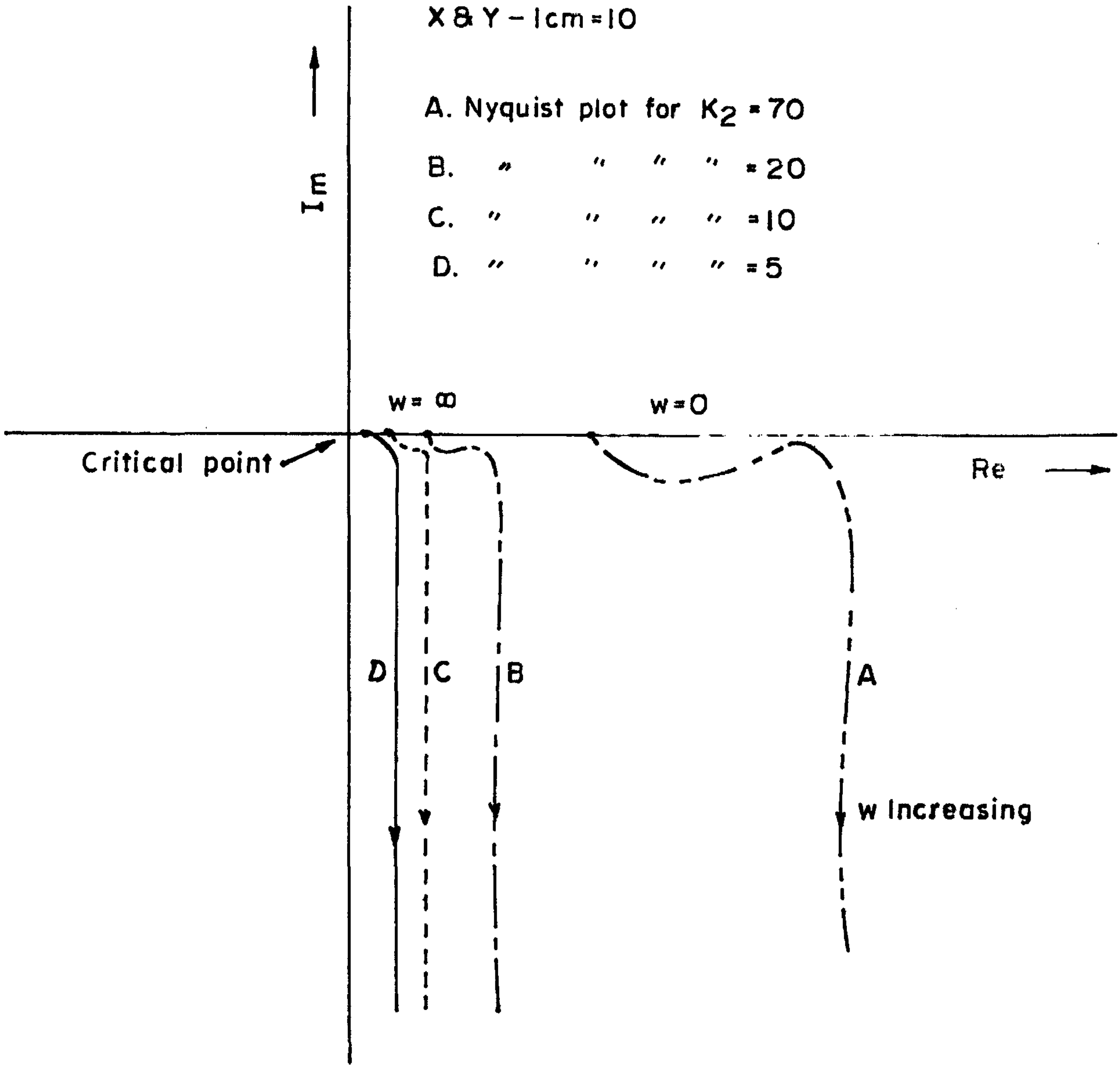


FIG.5.2 NYQUIST PLOT FOR DIFFERENT VALUES OF CONTROLLER TRANSFER FUNCTIONS WHEN 2nd LOOP IS CLOSED.

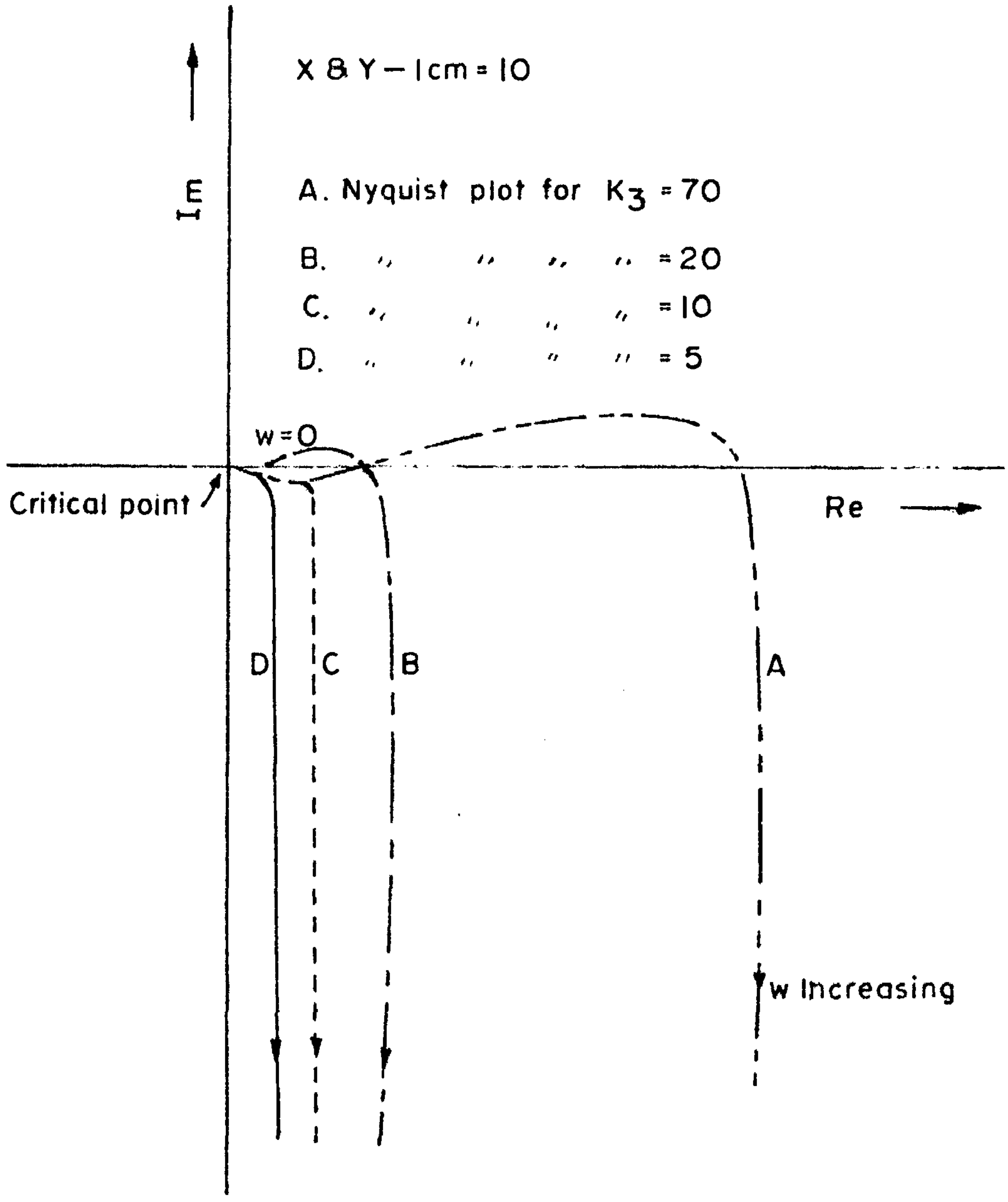


FIG.5.3 NYQUIST PLOTS FOR DIFFERENT VALUES OF CONTROLLER TRANSFER FUNCTIONS WHEN 3rd LOOP IS CLOSED.

Partial fractions may be obtained by providing the numerator polynomial array, array of real roots, array of imaginary roots and root multiplicy array. Here provision is made to prepare the data for the program of transients response. Thus in result alternate cards will be the data for the program of transient response.

Transient response program reads A(I), B(I), C(I), D(I) which are real part of root, real part of residue at this root, imaginary part of the root and imaginary part of the residue. E(I), F(I), G(I), H(I), are the same things for next polynomial of the same output. This program will give the transient response of the output with which the above two polynomials are associated. If you want transient responses for first output, give the value of L = - 1, for second output L = 0, and for third output L = 1.

5.2. CONTROLLER PROPOSED FOR THE REACTOR:

On the basis of results of sequential return difference algorithm, Nyquist criterion and after ensuring the tight feedback, three values of controller transfer functions K_1, K_2 and K_3 for all the three feedback loops are 70,70 and 70 respectively. These controller transfer functions provide three tight feedback loops and a stable system for better performance out of the chosen controller transfer functions. The Nyquist plots for all the three controller transfer functions are given in fig.5.1,5.2 and 5.3 respectively.

5.3. SCOPE OF FURTHER WORK :

In the present work, the order of Polynomials of the diagonal elements in the modified transfer function matrix rises too much. Some work has also been done to reduce the order by algebraic operations (3). Further work is possible in many branches of this Classico-Modern technique, such as finding the optimal sequence of loop closure considering the probability of failure of all the loops. Considerations of input, output transducer failure may be included as a part of the present algorithm. Inclusion of checking the phase minimality of modified diagonal elements and the design of compensator accordingly may give the compact and self sufficient algorithm. Further, the algorithm may be extended to optimal design as a iterative method. After all the above inclusions, the method requires the use of very fast computers with interactive graphic displays. Further work needs to be done in implementing the algorithm given in 4.5. PID type of controller transfer functions may also be incorporated.

CONCLUSION

In the present problem, the sequential return difference method has been applied which gives rise to very high order of numerator and denominator polynomials of the diagonal elements of modified transfer function matrix. This high order has given computational difficulties. Because of the slow computer used, at this stage it was very difficult to develop one more program for compensator and then to achieve the time response. Actually after some algebraic operations for reducing the order of polynomial, this method utilizing the compensator also will be very much useful, and the computational time may be reduced tremendously. This method is also capable of including the cases of transducer failure, which will increase the flexibility of the design method.

Interactive graphic display systems, if available along with a moderately fast computer, would make the design procedure feasible for higher order industrial systems.

Reliability consideration may also be included. The feedback loop which is most probable to transducer failure should be closed in the last. However, much work needs to be done for finding the optimal sequence of loop closure.

Had the computational time been small, all the programs could have been used as the subroutines of the first program. This way the design procedure would have become very simple. One would have required only to change K_1, K_2 and K_3, and would have obtained the responses.

APPENDIX - A

```
*******************************************************************************

 S.K.SHARMA   SEQUENTIAL RETURN DIFF. ALGOR9THM                         SRD 000
 FIRST DATA CARD CONAINS THE ORDERS                                     SRD 000
 AFTER FIRST NEXT NINE CARDS ARE TO READ THE TRANSF. FUNCT. MAT.        SRD 000
 ELEVENTH TO TWENTY FIRST DATA CARDS ARE FOR CONTROLLOR K1.             SRD 000
 TWENTY FIRST TO THIRTIETH DATA CARDS ARE FOR CONTROLLOR K2.            SRD 000
 THIRTY FIRST TO FORTIETH DATA CARDS ARE FOR CONTROLLOR K3.             SRD 000
 READ1,N,M,N1,M1                                                        SRD 000
 FORMAT(4I10)                                                           SRD 000
 M IS THE ORDER OF THE DENOMINAT. OF THE ELEMENT OF TRANS. FUNCT. MV,)  001
 N IS THE ORDER OF THE NUM. OF THE ELEMENT OF TRANS. FUNCT. MAT.        SRD 001
 N1 IS THE ORDER OF THE NUM. OFTHE  CONTROLLOR                          SRD 001
 M1 IS THE ORDER OF THE DENOMINAT. OF THE CONTROLLOR                    SRD 001
 DIMENSION  G11N(4),G11D(4),G12N(4),G13N(4),G13D(4),G12D(4),            SRD 001
1 G21N(4),G21D(4),G22N(4),G22D(4),G23N(4),G23D(4),G31N(4),G31D(4)       SRD 001
 DIMENSION RD(27),GN(27),GD(27),SPN(27),SPD(27),ZN2(27)                 SRD 001
 DIMENSION ZD1(27),ANZN(27),ANZD(27),RN1(27),RN2(27),RN(27)             SRD 001
 DIMENSION XN(27),XD(27),YN(27),YD(27),ZN(27),ZD(27),ZN1(27)            SRD 001
 DIMENSION G32N(4),G32D(4),G33N(4),G33D(4)                              SRD 001
 DIMENSION  KN(60,2),KD(60,2)    ,SM(27)
 IF(N-M) 988,988,989                                                    SRD 002
 IR=M+1                                                                 SRD 002
 GO TO 990                                                              SRD 002
 IR=N+1                                                                 SRD 002
 READ2,(G11N(I),I=1,IR),(G11D(I),I=1,IR)                                SRD 002
 READ2,(G12N(I),I=1,IR),(G12D(I),I=1,IR)                                SRD 002
 READ2,(G13N(I),I=1,IR),(G13D(I),I=1,IR)                                SRD 002
 READ2,(G21N(I),I=1,IR),(G21D(I),I=1,IR)                                SRD 002
 READ2,(G22N(I),I=1,IR),(G22D(I),I=1,IR)                                SRD 002
 READ2,(G23N(I),I=1,IR),(G23D(I),I=1,IR)                                SRD 003
 READ2,(G31N(I),I=1,IR),(G31D(I),I=1,IR)                                SRD 003
 READ2,(G32N(I),I=1,IR),(G32D(I),I=1,IR)                                SRD 003
 READ2,(G33N(I),I=1,IR),(G33D(I),I=1,IR)                                SRD 003
 IF(N1-M1) 991,991,992                                                  SRD 003
 IT=M1+1                                                                SRD 003
 GO TO 994                                                              SRD 003
 IT=N1+1                                                                SRD 003
 DO 987 L=1,15                                                          RHC 000
 READ2,(KN(L,I),I=1,IT),(KD(L,I),I=1,IT)                                SRD 003
 FORMAT (8F10.3)                                                        SRD 004
 IXN=N+1                                                                SRD 004
 IXD=M+1                                                                SRD 004
 IYN=N1+1                                                               SRD 004
 IYD=M1+1                                                               SRD 004
 DO 110 L=1,15                                                          SRD 068
 IF(L-6) 112,113,113                                                    SRD 069
 DO 114 I=1,IXN                                                         SRD 004
 G11N(I)=G22N(I)                                                        SRD 004
 G11D(I)=G22D(I)                                                        SRD 005
 G31N(I)=G32N(I)                                                        SRD 005
 G31D(I)=G32D(I)                                                        SRD 005
 G13N(I)=G23N(I)                                                        SRD 005
```

```
  G13D(I)=G23D(I)                                   SRD 005
  IF(L-11) 112,116,116                              SRD 069
  DO 117 I=1,IXN                                    SRD 005
  G11N(I)=G33N(I)                                   SRD 005
  G11D(I)=G33D(I)                                   SRD 005
  G21N(I)=G23N(I)                                   SRD 006
  G21D(I)=G23D(I)                                   SRD 006
  G12N(I)=G32N(I)                                   SRD 006
  G12D(I)=G32D(I)                                   SRD 006
  CONTINUE                                          SRD 006
  DO 15 I=1,IXN                                     SRD 006
  XN(I)=G11N(I)                                     SRD 006
5 XD(I)=G11D(I)                                     SRD 006
  DO 16 I=1,IYN                                     SRD 006
  YN(I)=KN(L,I)                                     SRD 006
  YD(I)=KD(L,I)                                     SRD 007
  FINDING THE SCALAR RETURNE DIFFERENCE             SRD 007
  IF (IXN*IYN) 4,4,5                                SRD 007
  IZN=0                                             SRD 007
  GO TO 9                                           SRD 007
  IZN =IXN+IYN                                      SRD 007
  DO 6 I=1,IZN                                      SRD 007
  ZN(I)=0.0                                         SRD 007
  DO 8 I=1,IXN                                      SRD 007
  DO 8 J=1,IYN                                      SRD 007
  K=I+J-1                                           SRD 008
  ZN(K)=XN(I)*YN(J)+ZN(K)                           SRD 008
  CONTINUE                                          SRD 008
  IZN=IZN-2                                         SRD 008
  IF(IXD*IYD) 10,10,11                              SRD 008
  IZD=0                                             SRD 008
  GO TO 14                                          SRD 008
  IZD=IXD+IYD                                       SRD 008
  DO12I=1,IZD                                       SRD 008
  ZD(I)=0.0                                         SRD 008
  DO 13 I=1,IXD                                     SRD 009
  DO 13 J=1,IYD                                     SRD 009
  K=I+J-1                                           SRD 009
  ZD(K)=XD(I)*YD(J)+ZD(K)                           SRD 009
    CONTINUE                                        SRD 009
  IZD=IZD-2                                         SRD 009
  MXN=IZN+1                                         SRD 009
  MYD=IZD+1                                         SRD 009
  ND=MXN                                            SRD 009
  IF (MXN -MYD) 17,18,18                            SRD 009
  ND=MYD                                            SRD 010
  IF (ND) 19,19,20                                  SRD 010
  DO 21 I=1,ND                                      SRD 010
  IF (I-MXN)22,22,23                                SRD 010
  IF(I-MYD)24,24,25                                 SRD 010
  ZN1(I)=ZN(I)+ZD(I)                                SRD 010
  GO TO 21                                          SRD 010
  ZN1(I)=ZD(I)                                      SRD 010
  GO TO 21                                          SRD 010
  ZN1(I)=ZN (I)                                     SRD 010
```

```
  CONTINUE                                                                      SRD 011
  IZN1=ND-1                                                                     SRD 01
  MD=IZD+1                                                                      SRD 011
  PUNCH 940                                                                     SRD 011
  FORMAT(79 (1H*))                                                              SRD 069
  PUNCH 928                                                                     SRD 011
  FORMAT (2X,62HORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DSRD 011
 1INOMINATOR)                                                                   SRD 011
  PUNCH 150,IZN1,IZD                                                            SRD 011
  FORMAT (2I10)                                                                 SRD 011
  PUNCH 929                                                                     SRD 012
  FORMAT (2X,22 (1H-))                                                          SRD 012
  PUNCH 930                                                                     SRD 012
  FORMAT (2X,63HSCALAR RETURN DIFFERENCE NUMERATOR AND DINOMINATOR RSRD 012
 1ESPECTIVELY)                                                                  SRD 012
  PUNCH 151,(ZN1(I), I=1,ND),(ZD(K),K=1,MD)                                     SRD 012
  FORMAT (8F10.3)                                                               SRD 012
  PUNCH 931                                                                     SRD 012
  FORMAT (2X,70 (1H*))                                                          SRD 012
  MSN=IZN1                                                                      SRD 069
  DO 801 I=1,IZN1                                                               SRD 069
  SM(I)=ZN1(I)                                                                  SRD 069
  CALL IROUTH(ZN1,IZN1)                                                         SRD C12
  PUNCH 827                                                                     SRD 069
  FORMAT (79 (1H*))                                                             SRD 069
  IZN1=MSN                                                                      SRD 070
  DO 802 I=1,IZN1                                                               SRD 070
2 ZN1(I)=SM(I)                                                                  SRD 070
  FINDING THE MODIFIED TRANSFER FUNCTION MATRIX                                 SRD 012
  MXN=IYN                                                                       SRD 013
  MYN=IZD+1                                                                     SRD 013
  IF(MXN*MYN) 27,27,28                                                          SRD 013
  MZN=0                                                                         SRD 013
  GO TO 31                                                                      SRD 013
  MZN=MXN+MYN                                                                   SRD 013
  DO 29 I=1,MZN                                                                 SRD 013
  ZN1(I)=0.0                                                                    SRD 013
  DO 30 I=1,MXN                                                                 SRD 013
  DO 30 J=1,MYN                                                                 SRD 013
  K=I+J-1                                                                       SRD 014
  ZN1(K)=ZN1(K)+YN(I)*ZD(J)                                                     SRD 014
  CONTINUE                                                                      SRD 014
  MZN=MZN-2                                                                     SRD 014
  MXD=IYD                                                                       SRD 014
  MYD =IZN +1                                                                   SRD 014
  IF (MXD*MYD)32,32,33                                                          SRD 014
  MZD=0                                                                         SRD 014
  GO TO 36                                                                      SRD 014
  MZD=MXD+MYD                                                                   SRD 014
  DO 34 I=1,MZD                                                                 SRD 015
  ZD1(I)=0.0                                                                    SRD 015
  DO 35 I=1,MYD                                                                 SRD 015
  DO 35 J=1,MXD                                                                 SRD 015
  K=I+J-1                                                                       SRD 015
  ZD1(K)=ZD1(K)+YD(J)*ZN(I)                                                     SRD 015
  CONTINUE                                                                      SRD 015
```

```
MZD =MZD-2                                                  SRD 01
DO 39 I=1,IXN                                               SRD 01
YN(I)=G11N(I)                                               SRD 01
DO 48 I=1,IXD                                               SRD 01
YD(I)= G11D(I)                                              SRD 01
LYN=N+1                                                     SRD 01
LYD=M+1                                                     SRD 01
LIZN=2                                                      SRD 01
LIZD=2                                                      SRD 01
DO 157 I=1,LIZN                                             SRD 01
ANZN(I)=0.0                                                 SRD 01
DO 158 I=1,LIZD                                             SRD 01
ANZD(I)=1.0                                                 SRD 01
IS=1                                                        SRD 01
IF(IXN*LYN)40,40,41                                         SRD 01
IZN1=0                                                      SRD 01
GO TO 44                                                    SRD 01
IZN1=IXN+LYN                                                SRD 01
DO 42 I=1,IZN1                                              SRD 01
ZN(I)=0.0                                                   SRD 01
DO43 I=1,IXN                                                SRD 01
DO 43 J=1,LYN                                               SRD 01
K=I+J-1                                                     SRD 01
ZN(K)=XN(I)*YN(J)+ZN(K)                                     SRD 01
CONTINUE                                                    SRD 01
IZN1=IZN1-2                                                 SRD 01
IF (IXD*LYD) 45,45,46                                       SRD 01
IZD1=0                                                      SRD 01
GO TO 51                                                    SRD 01
IZD1=IXD+LYD                                                SRD 01
DO 49 I=1,IZD1                                              SRD 01
ZD(I)=0.0                                                   SRD 01
DO 50 I=1,IXD                                               SRD 01
DO 50 J=1,LYD                                               SRD 01
K=I+J-1                                                     SRD 01
ZD(K)=XD(I)*YD(J)+ZD(K)                                     SRD 01
CONTINUE                                                    SRD 01
IZD1=IZD1-2                                                 SRD 01
IZN1=IZN1+1                                                 SRD 01
IZD1=IZD1+1                                                 SRD 01
IF (LIZD*IZN1) 120,120,121                                  SRD 01
IRN1=0                                                      SRD 01
GO TO 124                                                   SRD 01
 IRN1=LIZD+IZN1                                             SRD 02
DO 122 I=1,IRN1                                             SRD 02
RN1(I)=0.0                                                  SRD 02
DO 123 I=1,LIZD                                             SRD 02
DO 123 J=1,IZN1                                             SRD 02
K=I+J-1                                                     SRD 02
RN1(K)=RN1(K)+ANZD(I)*ZN(J)                                 SRD 02
CONTINUE                                                    SRD 02
IRN1=IRN1-2                                                 SRD 02
IF(LIZN*IZD1) 125,125,126                                   SRD 02
IRN2=0                                                      SRD 02
GO TO 129                                                   SRD 02
IRN2=LIZN+IZD1                                              SRD 02
```

```
      DO 127 I=1,IRN2
127   RN2(I)=0.0
      DO 128 I=1,LIZN
      DO 128 J=1,IZD1
      K=I+J-1
      RN2(K)=RN2(K)+ANZN(I)*ZD(J)
128   CONTINUE
      IRN2=IRN2-2
129   KXN=IRN1+1
      KYN=IRN2+1
      ND=KXN
      IF(KXN-KYN)130,131,131
130   ND=KYN
131   IF(ND)132,132,133
133   DO134 I=1,ND
      IF(I-KXN)135,135,136
135   IF(I-KYN)137,137,138
137   RN(I)=RN1(I)+RN2(I)
      GOTO134
136   RN(I)=RN2(I)
      GOTO134
138   RN(I)=RN1(I)
134   CONTINUE
132   IRN=ND-1
      IF(LIZD*IZD1)139,139,140
139   IRD=0
      GOTO143
140   IRD=LIZD+IZD1
      DO141 I=1,IRD
141   RD(I)=0.0
      DO142I=1,LIZD
      DO142J=1,IZD1
      K=I+J-1
      RD(K)=RD(K)+ANZD(I)*ZD(J)
142   CONTINUE
      IRD=IRD-2
143   IMN=IRN+1
      IMD=IRD+1
      DO52 I=1,IMN
52    ANZN(I)=RN(I)
      DO 53 I=1,IMD
53    ANZD(I)=RD(I)
      LIZN=IMN
      LIZD=IMD
      IS=IS+1
      IF(IS-2)55,56,55
56    DO57 I=1,IXN
57    XN(I)=G12N(I)
      DO61I=1,IXD
61    XD(I)=G12D(I)
      DO 59 I=1,LYN
59    YN(I)=G21N(I)
      DO 60 I=1,LYD
60    YD(I)=G21D(I)
      GO TO 58
55    IF(IS-3)62,63,62
```

```
63    DO 64 I=1,IXN                                  SRD 026
64    XN(I)=G13N(I)                                  SRD 027
      DO 65 I=1,IXD                                  SRD 027
65    XD(I)=G13D(I)                                  SRD 027
      DO 66 I=1,LYN                                  SRD 027
66    YN(I)=G31N(I)                                  SRD 027
      DO 67 I=1,LYD                                  SRD 027
 67   YD(I)=G31D(I)                                  SRD 027
      GO TO 58                                       SRD 027
 62   DO 68 I=1,IMN                                  SRD 027
      GN(I)=ANZN(I)                                  SRD 027
68    XN(I)=GN(I)                                    SRD 028
      DO 69 I=1,IMD                                  SRD 028
      GD(I)=ANZD(I)                                  SRD 028
69    XD(I)=GD(I)                                    SRD 028
      MZN=MZN+1                                      SRD 028
      MZD=MZD+1                                      SRD 028
      DO 70 I=1,MZN                                  SRD 028
70    YN(I)=ZN1(I)                                   SRD 028
      DO 71 I=1,MZD                                  SRD 028
71    YD(I)=ZD1(I)                                   SRD 028
      IF(IMN*MZN) 72,72,73                           SRD 029
72    ISZ=0                                          SRD 029
      GO TO 76                                       SRD 029
73    ISZ =IMN+MZN                                   SRD 029
      DO74I=1,ISZ                                    SRD 029
74    SPN(I)=0.0                                     SRD 029
      DO 75 I=1,IMN                                  SRD 029
      DO 75 J=1,MZN                                  SRD 029
      K=I+J-1                                        SRD 029
      SPN(K)=SPN(K)+XN(I)*YN(J)                      SRD 029
75    CONTINUE                                       SRD 030
      ISZ=ISZ-2                                      SRD 030
76    IF(IMD*MZD) 77,77,78                           SRD 030
77    ITZ=0                                          SRD 030
      GO TO 81                                       SRD 030
78    ITZ=IMD+MZD                                    SRD 030
      DO 79 I=1,ITZ                                  SRD 030
79    SPD(I)=0.0                                     SRD 030
      DO 80 I=1,IMD                                  SRD 030
      DO 80 J=1,MZD                                  SRD 030
      K=I+J-1                                        SRD 031
      SPD(K)=SPD(K)+XD(I)*YD(J)                      SRD 031
80    CONTINUE                                       SRD 031
      ITZ=ITZ-2                                      SRD 031
81    DO82I=1,IXN                                    SRD 031
      XN(I)=G11N(I)                                  SRD 031
82    XD(I)=G11D(I)                                  SRD 031
      IZ3=ISZ+1                                      SRD 031
      IZ4=ITZ+1                                      SRD 031
      DO 84 I=1,IZ3                                  SRD 031
84    YN(I)=SPN(I)                                   SRD 032
      DO 85 I=1,IZ4                                  SRD 032
85    YD(I)=SPD(I)                                   SRD 032
      IF(IXN*IZ4) 86,86,87                           SRD 032
86    IZN1=0                                         SRD 032
      GO TO 90                                       SRD 032
```

```
87     IZN1=IXN+IZ4
       DO 88 I=1,IZN1
88     ZN1(I)=0.0
       DO 89 I=1,IXN
       DO 89 J=1,IZ4
       K=I+J-1
       ZN1(K)=ZN1(K)+XN(I)*YD(J)
89     CONTINUE
       IZN1=IZN1-2
90     IF (IXN*IZ3) 91,91,92
91     IZN2=0
       GO TO 95
92     IZN2=IXN+IZ3
       DO 93 I=1,IZN2
93     ZN2(I)=0.0
       DO 94 I=1,IXN
       DO 94 J=1,IZ3
       K=I+J-1
       ZN2(K)=ZN2(K)+XD(I)*YN(J)
94     CONTINUE
       IZN2=IZN2-2
95     IX=IZN1+1
       IY=IZN2+1
       ND=IX
       IF(IX-IY) 96,97,97
96     ND=IY
97     IF (ND) 98,98,99
99     DO 100 I=1,ND
       IF(I-IX) 101,101,102
101    IF(I-IY) 103,103,104
103    ZN(I)=ZN1(I)-ZN2(I)
       GO TO 100
102    ZN(I)=-ZN2(I)
       GO TO 100
104    ZN(I)=ZN1(I)
100    CONTINUE
98     IZN=ND-1
       IF(IXN*IZ4) 105,105,106
105    IZD=0
       GO TO 109
106    IZD=IXN+IZ4
       DO 107 I=1,IZD
107    ZD(I)=0.0
       DO 108 I=1,IXN
       DO 108 J=1,IZ4
       K=I+J-1
       ZD(K)=ZD(K)+XD(I)*YD(J)
108    CONTINUE
       IZD=IZD-2
       IZD1=IZD+1
       IZN1=IZN+1
       PUNCH 943
943    FORMAT(2X,62HORDERS OF THE NUM. AND DINO. OF THE MODIFIED DIAGONAL
      1  ELEMENT)
109    PUNCH 152,IZN,IZD
152    FORMAT (2I10)
       PUNCH 942
```

```
942     FORMAT(22 (1H-))                                                       SRD 03
        PUNCH 944                                                              SRD 03
944     FORMAT(2X,61HNUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPESRD 03
       1 CTIVELY)                                                              SRD 07
        PUNCH153,(ZN(I),I=1,IZN1),(ZD(I),I=1,IZD1)                             SRD 03
153     FORMAT (8F10 3)                                                        SRD 03
110     CONTINUE                                                               SRD 03
        PUNCH 941                                                              SRD 03
941     FORMAT(79 (1H*))                                                       SRD 03
        STOP                                                                   SRD 03
        END                                                                    SRD 03
*******************************************************************************
**                                                                             SRD 00
        SUBROUTINE IROUTH(A,N1)                                                RHC  5
        DIMENSION A(27),X(27,27),B(27)                                         RHC C5
C       N1 IS THE ORDER OF THE POLYNOMIALS                                     RHC 00
        N=N1+1                                                                 RHC 00
        K=1                                                                    RHC C5
105     IF(A(K)) 101,102,101                                                   RHC C5
102     N=N-1                                                                  RHC C5
        DO 103 I=1,N                                                           RHC C5
103     A(I)=A(I+1)                                                            RHC C5
        GO TO 105                                                              RHC C5
101     DO 3 I=1,N                                                             RHC C5
3       B(I)=A(I)                                                              RHC 00
        S=(-1)**N                                                              RHC 00
        IF(S) 40,40,41                                                         RHC 00
40      N2=(N+1)/2                                                             RHC 00
        GO TO 44                                                               RHC 00
41      N2=N/2                                                                 RHC 00
44      DO 4 I=1,N2                                                            RHC 00
        JJ=1                                                                   RHC 00
        LL=2*I-1                                                               RHC 00
4       X(JJ,I)=B(LL)                                                          RHC 00
        T=(-1)**N                                                              RHC 00
        IF(T) 42,42,43                                                         RHC 001
42      N3=(N-1)/2                                                             RHC 001
        GO TO 45                                                               RHC 001
43      N3=N/2                                                                 RHC 001
45      DO 5 I=1,N3                                                            RHC 001
        JJ=2                                                                   RHC 001
        MM=2*I                                                                 RHC 002
5       X(JJ,I)=B(MM)                                                          RHC 002
        IF(X(2,1)) 91,92,91                                                    RHC 002
92      GO TO 8                                                                RHC 002
91      DO 6 JJ=3,N                                                            RHC 002
        K=N3-JJ+2                                                              RHC 002
        IF(K) 80,81,81                                                         RHC 002
80      KK=K+2                                                                 RHC 002
        GO TO 90                                                               RHC 002
81      KK=K+1                                                                 RHC 002
90      DO 70 I=KK,N3                                                          RHC 003
70      X(JJ,I)=0.0                                                            RHC 003
        IF(K) 51,50,49                                                         RHC 003
50      K=K+1                                                                  RHC 003
49      DO 6 I=1,K                                                             RHC 003
```

```
6      X(JJ,I)=X(JJ-2,I+1)-(X(JJ-2,1)*X(JJ-1,I+1))/X(JJ-1,1)            RHC 0035
51     DO 7 JJ=1,N                                                      RHC 0036
       IF(X(JJ,1)) 8,7,7                                                RHC 0037
7      CONTINUE                                                         RHC 0038
       GO TO 9                                                          RHC 0039
8      PUNCH 13                                                         RHC 0040
       GO TO 100                                                        RHC 0041
9      DO 10 I=1,N                                                      RHC 0042
       IF(X(I,1))8,11,10                                                RHC 0043
10     CONTINUE                                                         RHC 0044
       PUNCH 15                                                         RHC 0045
       GO TO 100                                                        RHC 0046
11     PUNCH 13                                                         RHC 0047
15     FORMAT (15X,6HSYSTEM,10X,2HIS,10X,6HSTABLE)                      RHC 0048
13     FORMAT (15X,6HSYSTEM,10X,2HIS,10X,8HUNSTABLE)                    RHC 0049
100    RETURN                                                           RHC 0050
       END                                                              RHC 0051
******************************************************************************
```

APPENDIX - B

```
*****************************************************************************
C  C  S.K.SHARMA PROGRAM FOR ENSURING TIGHT FEEDBACK
C     SUBROUTINE TIFB(A,B,N,M)
C     THIS SUBROUTINE ENSURES THE TIGHT FEEDBACK
C     A(I) IS THE NUMERATOR OF SCALAR RETURN DIFFERENCE
C     B(I) IS THE DINOMINATOR OF SCALAR RETURN DIFFERENCE
C     N-1 IS THE ORDER OF NUM. AND DINO. /F SCALAR RETUREN DIFFERENCE
C     M IS THE NUMBER OF SYSTEMS TO BE CHECKED
      DIMENSION A(10),B(10)
      READ 1,N,M
    1 FORMAT (2I10)
      NN=N $ KK=1  $MN=N
      DO 2 II=1,M
      READ 3,(A(I),I=1,N),(B(I),I=1,N)
    3 FORMAT (8F10.3)
    4 IF(A(KK)) 7,5,7
    5 N=N-1 $ DO 6 I=1,N
    6 A(I)=A(I+1) $ GO TO 4
    7 IF(B(KK)) 10,8,10
    8 NN=NN-1 $ DO 9 I=1,NN
    9 B(I)=B(I+1) $ GO TO 7
   10 W=50. $ AR=0. $ AI=0. $ BR=0. $ BI=0. $ DO 11 J=1,N $ L=N-J
      IF(A(J)) 25,11,25
   25 IF((-1)**L) 15,15,12
   12 IF((-1)**L/2) 13,13,14
   13 AR=AR-A(J)*(W**L) $ GO TO 11
   14 AR=AR+A(J)*(W**L) $ GO TO 11
   15 AI=AI+A(J)*(W**L)
   11 CONTINUE $ TN=SQRTF(AR**2+AI**2) $ DO 16 J=1,NN $ L=NN-J
      IF(B(J)) 26,16,26
   26 JK=(-1)**L $ JL=(-1)**L/2 $ IF(JK) 20,20,17
   17 IF(JL) 18,18,19
   18 BR=BR-B(J)*(W**L) $ GO TO 16
   19 BR=BR+B(J)*(W**L) $ GO TO 16
   20 BI=BI+B(J)*(W**L)
   16 CONTINUE $ TD=SQRTF(BR**2+BI**2) $ X=TN/TD $ X=1.+X $ Y=1./X
      N=MN $ NN=MN
      IF(Y-1.) 21,21,22
   21 PUNCH 23,II$ GO TO 2
   22 PUNCH 24,II
   23 FORMAT (16HTHE FEEDBACK OF ,I3,16H SYSTEM IS TIGHT)
   24 FORMAT (16HTHE FEEDBACK OF ,I3,20H SYSTEM IS NOT TIGHT)
    2 CONTINUE $ STOP $ END
*****************************************************************************
```

APPENDIX - C

```
*******************************************************************************
**
C  C  S.K.SHARMA PROGRAM FOR NYQUIST PLOT GENERATION
      READ 1,N,M
    1 FORMAT (2I10)
      DIMENSION A(10),B(10),X(200),Y(200)
      KK=1 $ MN=N $ NN=N
      PUNCH33 $ PUNCH 34 $ PUNCH 35 $ PUNCH 33
   33 FORMAT  (80(1H-))
   34 FORMAT  (4(1H*,3X,2HX-,4X,1H*,3X,2HY-,4X))
   35 FORMAT  (8(1H*,9HCORDINATE))
      DO 2II=1,M
      READ 3,(A(I),I=1,N),(B(J),J=1,N)
    3 FORMAT  (8F10.3)
    4 IF(A(KK)) 7,5,7
    5 N=N-1 $ DO 6 I=1,N
    6 A(I)=A(I+1) $ GO TO 4
    7 IF(B(KK)) 10,8,10
    8 NN=N-1 $ DO 9 I=1,NN
    9 B(I)=B(I+1) $ GO TO 7
   10 W=0. $ DO 27 IJ=1,20
      AR=0. $ AI=0. $ BR=0. $ BI=0. $ DO 11 J=1,N $ L=N-J
      JK=(-1)**L $ JL=(-1)**L/2 $ IF(A(J)) 25,11,25
   25 IF(JK) 15,15,12
   12 IF(JL) 13,13,14
   13 AR=AR-A(J)*(W**L) $ GO TO 11
   14 AR=AR+A(J)*(W**L)$ GO TO 11
   15 AI=AI+A(J)*(W**L)
   11 CONTINUE $ TN=SQRTF(AR**2+AI**2) $ DO 16 J=1,NN $ L=NN-J
      IF(B(J)) 26,16,26
   26 JK=(-1)**L $ JL=(-1)**L/2 $ IF(JK) 20,20,17
   17 IF(JL) 18,18,19
   18 BR=BR-B(J)*(W**L) $ GO TO 16
   19 BR=BR+B(J)*(W**L) $ GO TO 16
   20 BI=BI+B(J)*(W**L)
   16 CONTINUE $ TD=SQRTF(BR**2+BI**2) $ XM=TN/TD $ XA=ATANF(AI/AR)-ATAN
     1F(BI/BR) $ X(IJ)=XM*COSF(XA) $ Y(IJ)=XM*SINF(XA)
      W=W+.5
   27 CONTINUE $ PUNCH 28,(X(IJ),Y(IJ),IJ=1,20 )
   28 FORMAT  (8(1H*,F9.3))
      N=MN $ NN=MN
    2 CONTINUE $ STOP $ END

*******************************************************************************
```

APPENDIX - D

```
*****************************************************************************
**                                                                    STR 049
C  C  S.K.  SHARMA  STEP RESPONSE OF REACTER WITH CONTROLLOR          STR 049
C     N IS THE ORDER OF THE NUMERATOR OF THE ELEMENT OF TRANSFER FUNCTIOVT*AT49
C     M IS THE ORDER OF THE DINO. OF THE ELEMENT OF TRANSFER FUNCTION MAX,*XT50
C     N1 IS THE ORDER OF THE NUMERATOR OF THE STEP INPUT              X,* T50
C     M1 IS THE ORDER OF THE DINOMINATOR OF STEP INPUT                X,* T50.
C     XN, XD, ARE THE NUMERATOR AND DINOMINATOR OF STEP INPUT RESPECTIVEX,* T50.
C     L IS THE NUMBER OF ITERATIONS                                   X,* T50
C     FIRST ITERATION IS FOR K(1,3),K(2,3), AND K(3,3)                X,* T50.
C     SECOND ITERATION IS FOR K(1,4),K(2,4), AND K(3,4)               X,* T50
C     THIRD ITERATION IS FOR K(1,19),K(2,19), AND K(3,19)             X,* T50
C     FOURTH ITERATION IS FOR K(1,20),K(2,20), AND K(3,20)            X,* T50
C     X1, X2, AND X3, ARE THE OUTPUTS OF THE NUCLEAR REACTER          X,* T50
      READ 1,N,M,N1,M1,L                                              X,* T51
1     FORMAT(5I10)                                                    X,* T51.
      DIMENSION G11N(4,40),G11D(4,40),G22N(4,40),G22D(4,40),G33N(4,40) X,* T51.
      DIMENSION G33D(4,40),G1N(40),G1D(40),RN1(40),RN2(40),RN3(40)    X,* T51
      DIMENSION RD1(40),RD2(40),RD3(40),ZN1(40),ZN2(40),ZD1(40),YN(40) X,* T51.
      DIMENSION YN1(40),YN2(40),ZN(40),ZD(40),G12N(40),G13N(40),G12D(40)X,* T51.
      DIMENSION G13D(40),G21N(40),G21D(40),G23N(40),G23D(40),G31N(40) X,* T51
      DIMENSION G31D(40),G32N(40),G32D(40),XD(40),XN(40)              X,* T51
      IR=4                                                            X,* T51.
      IXN=N+1                                                         X,* T51.
      IXD=M+1                                                         X,* T52
      IYN=N1+1                                                            52
      IYD=M1+1                                                            52.
      $EUD=,(((RI((*(I*(GU)D(J,I),I=1,IXN),J=1,L)                         52.
      READ2,((G22N(J,I),G22D(J,I),I=1,IXN),J=1,L)                     STR 052
      READ2,((G33N(J,I),G33D(J,I),I=1,IXN),J=1,L)                     STR 052.
      READ 2,(G12N(I),G12D(I),I=1,IR)                                 STR 052
      READ 2,(G13N(I),G13D(I),I=1,IR)                                 STR 052
      READ 2,(G21N(I),G21D(I),I=1,IR)                                 STR 052.
      READ 2,(G23N(I),G23D(I),I=1,IR)                                 STR 052
      READ 2,(G31N(I),G31D(I),I=1,IR)                                 STR 053
      READ 2,(G32N(I),G32D(I),I=1,IR)                                 STR 053
      READ 2,(XN(I),XD(I),I=1,IYN)                                    STR 053.
2     FORMAT (8F10.3)                                                 STR 053.
      DO 101 J=1,L                                                    STR 053
      DO 16 I=1,IXN                                                   STR 053
16    G1N(I)=G11N(J,I)                                                STR 053
      DO 17 I=1,IXD                                                   STR 053
17    G1D(I)=G11D(J,I)                                                STR 053.
      IS=1                                                            STR 053.
100   CALL PMULT  (RN1,IRN1,G1N,IXN,XN,IYN)                           STR 054
      CALL PMULT  (RD1,IRD1,G1D,IXD,XD,IYD)                           STR 054.
      CALL PMULT  (RN2,IRN2,G12N,IR,XN,IYN)                           STR 054
      CALL PMULT  (RD2,IRD2,G12D,IR,XD,IYD)                           STR 054.
      CALL PMULT  (RN3,IRN3,G13N,IR,XN,IYN)                           STR 054
      CALL PMULT  (RD3,IRD3,G13D,IR,XD,IYD)                           STR 054
      IRN1=IRN1+1                                                     STR 054
      IRN2=IRN2+1                                                     STR 054
      IRN3=IRN3+1                                                     STR 054.
      IRD1=IRD1+1                                                     STR 054
      IRD2=IRD2+1                                                     STR 055
      IRD3=IRD3+1                                                     STR 055.
      CALL  PMULT  (ZN1,IZN1,RD1,IRD1,RN2,IRN2)                       STR 055
      CALL PMULT  (ZN2,IZN2,RN1,IRN1,RD2,IRD2)                        STR 055
```

```
      CALL PMULT  (ZD1,IZD1,RD1,IRD1,RD2,IRD2)
      IZN1=IZN1+1
      IZN2=IZN2+1
      IZD1=IZD1+1
      CALL  PSUM  (YN,ISN,ZN1,IZN1,ZN2,IZN2)
      CALL  PMULT  (YN1,IYN1,ZD1,IZD1,RN3,IRN3)
      ISN=ISN+1
      IYN1=IYN1+1
      CALL  PMULT  (YN2,IYN2,YN,ISN,RD3,IRD3)
      CALL  PMULT  (ZD,IZD,ZD1,IZD1,RD3,IRD3)
      IYN2=IYN2+1
      CALL  PSUM  (ZN,IZN,YN1,IYN1,YN2,IYN2)
      IZN3=IZN+1
      IZD3=IZD+1
      ITRAN=J
      PUNCH 70,ITRAN
70    FORMAT (10X,9HITERATION,10X,6HNUMBER,10X,2HIS,5X,I4)
      IF(IS-1)20,21,20
21    PUNCH 71,IZN
      PUNCH 72,IZD
      PUNCH 222
      PUNCH22,(ZN(I),I=1,IZN3)
      PUNCH 232
      PUNCH23,(ZD(I),I=1,IZD3)
71    FORMAT (10X,5HORDER,10X,2HOF,7HXN1(I)=,5X,I4)
222   FORMAT (10X,4HNUM.,5X,2HOF,5X,5HFIRST,5X,6HOUTPUT,5X,3HX1.,5X,
     1 2HIS,5X,5HGIVEN,5X,3HAS.,5X)
22    FORMAT (8F10.3)
72    FORMAT (10X,5HORDER,10X,2HOF,10X,7HXD1(I)=,5X,I4)
232   FORMAT (10X,5HDINO.,5X,2HOF,5X,5HFIRST,5X,6HOUTPUT,5X,3HX1.,5X,
     1 2HIS,5X,5HGIVEN,5X,3HAS.,4X)
23    FORMAT (8F10.3)
20    IF(IS-2)32,25,24
25    PUNCH 73,IZN
      PUNCH 74,IZD
      PUNCH 262
      PUNCH26,(ZN(I),I=1,IZN3)
      PUNCH 272
      PUNCH27,(ZD(I),I=1,IZD3)
73    FORMAT (10X,5HORDER,10X,2HOF,10X,7HXN2(I)=,5X,I4)
262   FORMAT (10X,4HNUM.,5X,2HOF,5X,6HSECOND,5X,6HOUTPUT,5X,3HX2.,5X,
     1 2HIS,5X,5HGIVEN,5X,3HAS.,4X)
26    FORMAT (8F10.3)
74    FORMAT (10X,5HORDER,10X,2HOF,10X,7HXD2(I)=,5X,I4)
272   FORMAT (10X,5HDINO.,5X,2HOF,5X,6HSECOND,5X,6HOUTPUT,5X,3HX2.,5X,
     1 2HIS,5X,5HGIVEN,5X,3HAS.,3X)
27    FORMAT (8F10.3)
24    IF (IS-3)32,29,101
29    PUNCH 75,IZN
      PUNCH 76,IZD
      PUNCH 302
      PUNCH30,(ZN(I),I=1,IZN3)
      PUNCH 312
      PUNCH31,(ZD(I),I=1,IZD3)
75    FORMAT (10X,5HORDER,10X,2HOF,10X,7HXN3(I)=,5X,I4)
```

```
A02   FORMAT (10X,4HNUM.,5X,2HOF,5X,5HTHIRD,5X,6HOUTPUT,5X,3HX3.,5X,   STR 060
     1 2HIS,5X,5HGIVEN,5X,3HAS.,5X)                                    STR 061
30    FORMAT (8F10.3)                                                  STR 061
76    FORMAT (10X,5HORDER,10X,2HOF,10X,7HXD3(I)=,5X,I4)                STR 061
312   FORMAT (10X,5HDINO.,5X,2HOF,5X,5HTHIRD,5X,6HOUTPUT,5X,3HX3.,5X,  STR 061
     1 2HIS,5X,5HGIVEN,5X,3HAS.,4X)                                    STR 061
31    FORMAT (8F10.3)                                                  STR 061
32    IS=IS+1                                                          STR 061
      IF(IS-2)5,6,5                                                    STR 061
6     DO 7 I=1,IXN                                                     STR 061
7     G1N(I)=G22N(J,I)                                                 STR 061
      DO 8 I=1,IXD                                                     STR 062
8     G1D(I)=G22D(J,I)                                                 STR 062
      DO 9 I=1,IR                                                      STR 062
      G12N(I)=G21N(I)                                                  STR 062
      G12D(I)=G21D(I)                                                  STR 062
      G13N(I)=G23N(I)                                                  STR 062
9     G13D(I)=G23D(I)                                                  STR 062
      GO TO 100                                                        STR 062
5     IF(IS-3)101,11,101                                               STR 062
11    DO 12 I=1,IXN                                                    STR 062
12    G1N(I)=G33N(J,I)                                                 STR 063
      DO 13 I=1,IXD                                                    STR 063
13    G1D(I)=G33D(J,I)                                                 STR 063
      DO 14 I=1,IR                                                     STR 063
      G12N(I)=G31N(I)                                                  STR 063
      G12D(I)=G31D(I)                                                  STR 063
      G13N(I)=G32N(I)                                                  STR 063
14    G13D(I)=G32D(I)                                                  STR 063
      GO TO 100                                                        STR 063
101   CONTINUE                                                         STR 063
18    STOP                                                             STR 064
      END                                                              STR 064
***********************************************************************
**                                                                     STR 064
      SUBROUTINE  PMULT  (Z,IZ,C,IC,D,ID)                              STR 064
C     THIS SUBROUTINE PERFORMS THE MULTIPLICATION OF TWO POLYNOMIALS   STR 064
      DIMENSION Z(40),C(40),D(40)                                      STR 064
      IX=IC                                                            STR 064
      IY=ID                                                            STR 064
      IF(IX*IY)50,50,51                                                STR 064
50    IZ=0                                                             STR 064
      GO TO 54                                                         STR 065
51    IZ=IX+IY                                                         STR 065
      DO 52 I=1,IZ                                                     STR 065
52    Z(I)=0.0                                                         STR 065
      DO 53 I=1,IX                                                     STR 065
      DO 53 J=1,IY                                                     STR 065
      K=I+J-1                                                          STR 065
      Z(K)=Z(K)+C(I)*D(J)                                              STR 065
53    CONTINUE                                                         STR 065
      IZ=IZ-2                                                          STR 065
54    RETURN                                                           STR 066
      END                                                              STR 066
***********************************************************************
**                                                                     STR 066
```

```
      SUBROUTINE PSUM  (Z,IZ,X,IT,Y,IS)
C     THIS SUBROUTINE PERFORMS THE SUM OF TWO POLYNOMIALS

      DIMENSION Z(40),X(40),Y(40)
      IX=IT
      IY=IS
      ND=IX
      IF(IX-IY)55,56,56
55    ND=IY
56    IF(ND)57,57,58
58    DO 59 I=1,ND
      IF(I-IX)60,60,61
60    IF(I-IY)62,62,63
62    Z(I)=X(I)+Y(I)
      GO TO 59
61    Z(I)=Y(I)
      GO TO 59
63    Z(I)=X(I)
59    CONTINUE
57    IZ=ND-1
      RETURN
      END
*************************************************************************************************
```

APPENDIX – E

```
*****************************************************************************
**
C   C  S.K.SHARMA FINDING THE ROOTS OF A POLYNOMIAL                 PRT 00
C      N IS THE TOTAL NUMBER OF COEFFICIENTS IN THE POLYNOMIALS     PRT 00
C      P1 AND Q1 ARE THE STARTING VALUES OF THE ROOTS               PRT 00
C      E IS THE ACCURACY OF THE ROOTS                               PRT 00
       DIMENSION A(100),B(100),C(100)                               PRT 00
       READ 100,P1,Q1,E,N                                           PRT 00
       READ 106,(A(I),I=1,N)                                        PRT 00
       K=1
105    IF(A(K)) 101,102,101
102    N=N-1
       DO 103 I=1,N
103    A(I)=A(I+1)
       GO TO 105
101    N=N-1
       DO 12 I=1,N                                                  PRT 00
12     A(I)=A(I+1)/A(1)                                             PRT 00
       LL=N                                                         PRT 00
       DO 17 K=1,N                                                  PRT 00
151    IF (A(LL)) 13,14,13                                          PRT 00
14     PUNCH 16                                                     PRT 00
16     FORMAT (3HONE,2X,4HROOT,2X,7H=0.0000)                        PRT 00
       LL=LL-1                                                      PRT 00
17     CONTINUE                                                     PRT 00
13     LZ=K-1                                                       PRT 00
       N=N-LZ                                                       PRT 00
       P=P1                                                         PRT 00
       Q=Q1                                                         PRT 00
       GO TO 75                                                     PRT 00
    10 B(1)=A(1)-P                                                  PRT 00
       B(2)=A(2)-P*B(1)-Q                                           PRT 00
       DO 6 K=3,N                                                   PRT 00
    6  B(K)=A(K)-P*B(K-1)-Q*B(K-2)                                  PRT 00
       IF(N-3) 91,91,5                                              PRT 00
     5 L=N-1                                                        PRT 00
       C(1)=B(1)-P                                                  PRT 00
       C(2)=B(2)-P*C(1)-Q                                           PRT 00
       DO 7 J=3,L                                                   PRT 00
     7 C(J)=B(J)-P*C(J-1)-Q*C(J-2)                                  PRT 00
       CBAR=C(L)-B(L)                                               PRT 00
       DNR=(C(N-2))**2-CBAR*C(N-3)                                  PRT 00
       IF(DNR) 15,500,15                                            PRT 00
   500 P=P+1.                                                       PRT 00
       Q=Q+1.                                                       PRT 00
       GO TO 10                                                     PRT 00
    15 DLTP=(B(N-1)*C(N-2)-B(N)*C(N-3))/DNR                         PRT 00
     9 DLTQ=(B(N)*C(N-2)-B(N-1)*CBAR)/DNR                           PRT 00
       P=P+DLTP                                                     PRT 00
       Q=Q+DLTQ                                                     PRT 00
       ABP=ABSF(DLTP)                                               PRT 00
       ABQ=ABSF(DLTQ)                                               PRT 00
       XM=ABP+ABQ                                                   PRT 00
```

```
      IF(XM-E) 50,50,10
C     COMPUTING ROOTS OFQUADRATIC EQUATION
   50 DCRN=P*P-4.*Q
      IF(DCRN) 52,53,54
   52 ABD=ABSF(DCRN)
      ABSQ=SQRT(ABD)
      XR=P/2.
      XIM=ABSQ/2.
      PUNCH 206,XR,XIM
  206 FORMAT(18HREAL PART OF ROOT=,F10.6,2X,15HIMAGINARY PART=,F10.6)
      GO TO 70
   53 X=P/2.
      PUNCH 207,X
  207 FORMAT(12HEQUAL ROOTS=,F10.6)
      GO TO 70
   54 ABSQ=SQRTF(DCRN)
      X1=-P/2.-ABSQ/2.
      X2=-P/2.+ABSQ/2.
      PUNCH 209,X1,X2
  209 FORMAT(11HFIRST ROOT=,F10.6,12HSECOND ROOT=,F10.6)
   70 N=N-2
      DO 71 K=1,N
   71 A(K)=B(K)
   75 IF(N)99,99,76
   76 IF(N-4)77,77,10
   77 GOTO (80,50,10,10),N
   80 X=-A(1)
      PUNCH 205,X
      GO TO 99
   91 C(1)=B(1)-P
      C(2)=B(2)-P*C(1)-Q
      CBAR=C(L)-B(L)
      DNR=(C(N-2))**2-CBAR
      IF(DNR) 600,700,600
  700 P=P+1.
      Q=Q+1.
      GO TO 91
  600 DLTP=(B(N-1)*C(N-2)-B(N))/DNR
      GO TO 9
  100 FORMAT(3F10.5,I3)
  205 FORMAT(11HLAST ROOT= ,F10.6)
106   FORMAT (8F10.3  )
   99 STOP
      END
**********************************************************************************
```

```
*************************************************************************************
**                                                                         PRF 0C
C  C  S.K.SHARMA  PARTIAL FRACTION EXPENSION                               PRF 0C
C     NNM IS THE ORDER OF THE NUMERATOR OF THE FUNCTION TO BE PARTIAL      PRF 0C
C     FRACTIONED                                                           PRF 00
C     IT IS THE NUMBER OF ROOTS WHICH ARE DISTINCT                         PRF 0C
C     NM IS THE TOTAL NUMBER OF FUNCTIONS TO BE PARTIAL FRACTIONED         PRF 00
C     MA IS THE ORDER OF THE NUMERATOR PLUS ONE                            PRF 0C
C     A(K,I) IS THE NUMERATOR POLYNOMIALS MATRIX                           PRF 00
C     M(K,I) IS THE ROOT MULTIPLICITY MATRIX                               PRF 00
C     CR(K,I) IS THE REAL ROOTS MATRIX OF THE POLYNOMIALS                  PRF 00
C     CI(K,I) IS THE IMAGINARY ROOTS MATRIX OF THE POLY NOMIALS            PRF 00
C     RESR(K,I) IS THE MATRIX FOR THE REAL PART OF RESIDUES                PRF 00
C     RESI(K,I) ISTHE MATRIX FOR THE IMAGINARY PARTS OF THE RESIDUES       PRF 00
      DIMENSION A(2,40),P(40),D(40),DX(40),M(2,40),Q(40),DXX(40)           PRF 00
      DIMENSION S2(40),ANS(40),AX(40),P1(40),P2(40),S1(40)                 PRF 00
      DIMENSION RESR(40,8),RESI(40,8),CR(2,40),CI(2,40)                    PRF 00
      READ1,NNM,IT,NM,MA                                                   PRF 00
1     FORMAT(4I10)                                                         PRF 00
      DO 995 K=1,NM                                                        PRF 00
      READ2,(A(K,I),I=1,MA)                                                PRF 00
      READ2, (CR(K,I),I=1,IT),(CI(K,I),I=1,IT)                             PRF 00
2     FORMAT(8F10.3)                                                       PRF 00
995   CONTINUE                                                             PRF 00
      READ 222,((M(K,I),I=1,IT),K=1,NM)                                    PRF 00
222   FORMAT (8I10)                                                        PRF 00
      DO 1001 K=1,NM                                                       PRF 00
      DO 199 I=1,MA                                                        PRF 00
      D(I)=0.0                                                             PRF 00
      P(I)=0.0                                                             PRF 00
      Q(I)=0.0                                                             PRF 00
      DXX(I)=0.0                                                           PRF 00
199   DX(I)=0.0                                                            PRF 00
      DO 4 L=1,MA                                                          PRF 00
4     AX(L)=A(K,L)                                                         PRF 00
      KK=1                                                                 PRF 00
805   IF(AX(KK)) 801,802,801                                               PRF 00
802   MA=MA-1                                                              PRF 00
      DO 803 II=1,MA                                                       PRF 00
803   AX(II)=AX(II+1)                                                      PRF 00
      GO TO 805                                                            PRF 00
801   NNM=MA-1                                                             PRF 00
200   DO 30 I=1,IT                                                         PRF 00
      DO 5 L=1,MA                                                          PRF 00
      P1(L)=AX(L)                                                          PRF 00
5     P2(L)=0.0                                                            PRF 00
      DO 6 L=1,IT                                                          PRF 00
      S1(L)=CR(K,L)                                                        PRF 00
6     S2(L)=CI(K,L)                                                        PRF 00
C     CALCULATING THE VALUE OF NUMERATOR POLYNOMIAL AT I,TH ROOT           PRF 00
      YR=0.                                                                PRF 00
      YI=0.                                                                PRF 00
      DO 7 J=1,MA                                                          PRF 00
      L2=MA-J                                                              PRF 00
      YR=YR+P1(J)*(S1(I)**L2)                                              PRF 00
      L3=(-1)**L2                                                          PRF 00
      IF(L3) 7003,7003,7000                                                PRF 00
7000  L4=(-1)**(L3/2)                                                      PRF 00
      IF(L4) 7001,7001,7002                                                PRF 00
7001  YR=YR-P1(J)*(S1(I)**L2)                                              PRF 00
      GO TO 7                                                              PRF 00
7002  YR=YR+P1(J)*(S1(I)**L2)                                              PRF 00
      GO TO 7                                                              PRF 00
```

```
      IF(IQ1*ID1)10,10,20
   10 IDX=0
      GO TO 50
   20 IDX=IQ1+ID1
      DO 35 I=1,IDX
   35 DX(I)=0.0
      DO 36 I=1,IQ1
      DO 36 J=1,ID1
      L=I+J-1
      DX(L)=DX(L)+Q(I)*D(J)
   36 CONTINUE
      IDX=IDX-2
   50 D(1)=CI(K,JJ)*CI(K,JJ)+CR(K,JJ)*CR(K,JJ)
      D(2)=2.*CR(K,JJ)
      D(3)=1.
      ID=2
      IP1=IP+1
      ID1=ID+1
      IF(IP1*ID1)51,51,52
   51 IDXX=0
      GO TO 55
   52 IDXX=IP1+ID1
      DO 53 I=1,IDXX
   53 DXX(I)=0.0
      DO 54 I=1,IP1
      DO 54 J=1,ID1
      L=I+J-1
      DXX(L)=DXX(L)+P(I)*D(J)
   54 CONTINUE
      IDXX=IDXX-2
   55 IS=IDX+1
      IDX=IDXX+1
      ND=IS
      IF(IS-IDX) 56,57,57
   56 ND=IDX
   57 IF(ND)58,58,59
   59 DO 60 I=1,ND
      IF(I-IS) 61,61,62
   61 IF(I-IDX)63,63,64
   63 P(I)=DX(I)+DXX(I)
      GO TO 60
   62 P(I)=DXX(I)
      GO TO 60
   64 P(I)=DX(I)
   60 CONTINUE
   58 IP=ND-1
      IR=IQ+1
      IY=ID+1
      IF(IR*IY)65,65,66
   65 IX=0
      GO TO 69
   66 IX=IR+IY
      DO 67 I=1,IX
   67 DX(I)=0.0
      DO 68 I=1,IR
      DO 68 J=1,IY
```

```
7003  YI=YI+P1(J)*(S2(I)**L2)                                        PRF 00
7     CONTINUE                                                        PRF 00
C     FINDING THE RESIDUES AT DISTINCT ROOTS                          PRF 00
      PI=0.                                                           PRF 00
      IMI=M(K,I)                                                      PRF 00
      PR=1.                                                           PRF 00
      IF(IMI)30,30,390                                                PRF 00
390   DO 31 J=1,IT                                                    PRF 00
      IF(I-J)32,31,32                                                 PRF 00
32    IMJ=M(K,J)                                                      PRF 00
      IF(IMJ)31,31,33                                                 PRF 00
33    A1=CR(K,I)-CR(K,J)                                              PRF 00
      B1=CI(K,I)-CI(K,J)                                              PRF 00
      DO 34 L=1,IMJ                                                   PRF 00
      A2=PR*A1-PI*B1                                                  PRF 00
      B2=PR*B1+PI*A1                                                  PRF 00
      PR=A2                                                           PRF 00
      PI=B2                                                           PRF 00
34    CONTINUE                                                        PRF 00
31    CONTINUE                                                        PRF 00
      DIV=PR*PR+PI*PI                                                 PRF 00
      RESR(I,IMI)=(PR*YR+PI*YI)/DIV                                   PRF 00
      RESI(I,IMI)=(PR*YI-PI*YR)/DIV                                   PRF 00
      PUNCH 528,RESR(I,IMI),RESI(I,IMI)                               PRF 00
528    FORMAT (4HREAL,1X,4HPART,1X,2HOF,1X,9HRESIDUES=,E14.8,         PRF 00
     1 9HIMAGINARY,1X,4HPART,1X,2HOF,1X,9HRESIDUES=,E14.8)            PRF 00
      SK=S1(I)                                                        PRF 00
      SS=S2(I)                                                        PRF 00
      SKA=RESR(I,IMI)                                                 PRF 00
      SA=RESI(I,IMI)                                                  PRF 00
      PUNCH 7005,SK,SKA,SS,SA                                         PRF 00
7005  FORMAT(4F15.4)                                                  PRF 00
30    CONTINUE                                                        PRF 00
C     FINDING OUT THE RESIDUES FOR THE ROOT MULTIPLICITY              PRF 00
      DO 300 I=1,IT                                                   PRF 00
      IF(M(K,I)-1)300,300,301                                         PRF 00
300   CONTINUE                                                        PRF 00
      GO TO 205                                                       PRF 00
301   JJ=0                                                            PRF 01
      DO 40 I=1,MA                                                    PRF 01
      Q(I)=0.0                                                        PRF 01
      P(I)=0.0                                                        PRF 01
40    CONTINUE                                                        PRF 01
      IQ=0                                                            PRF 01
      IP=0                                                            PRF 01
      Q(1)=1.0                                                        PRF 01
42    JJ=JJ+1                                                         PRF 01
      ICH=M(K,JJ)                                                     PRF 01
      IF(ICH)42,42,422                                                PRF 01
422   IF(CI(K,JJ))43,45,43                                            PRF 01
43    D(1)=-2.*RESI(JJ,ICH)*CI(K,JJ)-2.*RESR(JJ,ICH)*CR(K,JJ)         PRF 01
      D(2)=2.*RESR(JJ,ICH)                                            PRF 01
      ID=1                                                            PRF 01
      IQ1=IQ+1                                                        PRF 01
      ID1=ID+1                                                        PRF 01
```

```
      IF(IQ1*ID1)10,10,20
10    IDX=0
      GO TO 50
20    IDX=IQ1+ID1
      DO 35 I=1,IDX
35    DX(I)=0.0
      DO 36 I=1,IQ1
      DO 36 J=1,ID1
      L=I+J-1
      DX(L)=DX(L)+Q(I)*D(J)
36    CONTINUE
      IDX=IDX-2
50    D(1)=CI(K,JJ)*CI(K,JJ)+CR(K,JJ)*CR(K,JJ)
      D(2)=2.*CR(K,JJ)
      D(3)=1.
      ID=2
      IP1=IP+1
      ID1=ID+1
      IF(IP1*ID1)51,51,52
51    IDXX=0
      GO TO 55
52    IDXX=IP1+ID1
      DO 53 I=1,IDXX
53    DXX(I)=0.0
      DO 54 I=1,IP1
      DO 54 J=1,ID1
      L=I+J-1
      DXX(L)=DXX(L)+P(I)*D(J)
54    CONTINUE
      IDXX=IDXX-2
55    IS=IDX+1
      IDX=IDXX+1
      ND=IS
      IF(IS-IDX) 56,57,57
56    ND=IDX
57    IF(ND)58,58,59
59    DO 60 I=1,ND
      IF(I-IS) 61,61,62
61    IF(I-IDX)63,63,64
63    P(I)=DX(I)+DXX(I)
      GO TO 60
62    P(I)=DXX(I)
      GO TO 60
64    P(I)=DX(I)
60    CONTINUE
58    IP=ND-1
      IR=IQ+1
      IY=ID+1
      IF(IR*IY)65,65,66
65    IX=0
      GO TO 69
66    IX=IR+IY
      DO 67 I=1,IX
67    DX(I)=0.0
      DO 68 I=1,IR
      DO 68 J=1,IY
```

```
      L=I+J-1
      DX(L)=DX(L)+Q(I)*D(J)
68    CONTINUE
      IX=IX-2
69    LL=IX+1
      DO 70 I=1,LL
70    Q(I)=DX(I)
      IQ=IX
      JJ=JJ+1
      IF(IT-JJ)71,71,42
 45   D(1)=-CR(K,JJ)
      D(2)=1.
      ID=1
      DO 74 IN=1,ICH
      ID1=ID+1
      IP1=IP+1
      IF(ID1*ID1)75,75,76
75    IXX=0
      GO TO 79
76    IXX=0
761   IXX=ID1+IP1
      DO 77 I=1,IXX
77    DX(I)=0.
      DO 78 I=1,ID1
      DO 78 J=1,IP1
      L=I+J-1
      DX(L)=DX(L)+D(I)*P(J)
78    CONTINUE
      IXX=IXX-2
79    LL=IXX+1
      DO 81 I=1,LL
81    P(I)=DX(I)
      IP=IXX
74    CONTINUE
      IQP1=IQ+1
      DO 82 IXX=1,IQP1
      DXX(IXX)=RESR(JJ,ICH)*Q(IXX)
82    CONTINUE
      IK=IP
      IK1=IK+1
      IQ1=IQ+1
      ND=IK1
      IF(IK1-IQ1)83,84,84
83    ND=IQ1
84    IF(ND)85,85,86
86    DO 91 I=1,ND
      IF(I-IK1)87,87,88
87    IF(I-IQ1)89,89,90
89    P(I)=DX(I)+DXX(I)
      GO TO 91
88    P(I)=DXX(I)
      GO TO 91
90    P(I)=DX(I)
91    CONTINUE
85    IP=ND-1
      DO 49 IXX=1,ICH
      IX=ID+1
```

```
      IY=IQ+1
      IF(IX*IY)92,92,93
92    IKX=0
      GO TO 96
93    IKX=IX+IY
      DO 94 I=1,IKX
94    DX(I)=0.
      DO 95 I=1,IX
      DO 95 J=1,IY
      L=I+J-1
      DX(L)=DX(L)+D(I)*Q(J)
95    CONTINUE
      IKX=IKX-2
96    LL=IKX+1
      DO 97 I=1,LL
97    Q(I)=DX(I)
      IQ=IKX
49    CONTINUE
71    IX=NNM+1
      IY=IP+1
      ND=IX
      IF(IX-IY)98,99,99
98    ND=IY
99    IF(ND)100,100,101
101   DO 102 I=1,ND
      IF(I-IX)103,103,104
103   IF(I-IY)105,105,106
105   ANS(I)=AX(I)-P(I)
      GO TO 102
104   ANS(I)=-P(I)
      GO TO 102
106   ANS(I)=AX(I)
102   CONTINUE
100   IANS=ND-1
      LL=IANS+1
      DO 107 I=1,LL
107   AX(I)=ANS(I)
      NNM=IANS
      TOL=0.001
      I=0
501   I=I+1
      IF(I-IT) 701,701,200
701   IF(M(K,I)) 501,501,502
502   M(K,I)=M(K,I)-1
      IF(CI(K,I))507,504,507
507   D(1)=CR(K,I)*CR(K,I)+CI(K,I)*CI(K,I)
      D(2)=-2.*CR(K,I)
      D(3)=1.
      ID=2
      I=I+1
      IF(I-IT) 702,702,505
702   M(K,I)=M(K,I)-1
      GO TO 505
504   D(1)=-CR(K,I)
      D(2)=1.
```

```
      ID=1                                                              PRF 02
505   IX=NNM+1                                                          PRF 02
      IY=ID+1                                                           PRF 02
508   IF(IY)509,509,510                                                 PRF 02
510   IF(ABSF(D(IY))-TOL)511,511,509                                    PRF 02
511   IY=IY-1                                                           PRF 02
      GO TO 508                                                         PRF 02
509   IF(IY)512,512,513                                                 PRF 02
513   IANS=IX-IY+1                                                      PRF 02
      IF(IANS)514,515,516                                               PRF 02
514   IANS=0                                                            PRF 02
515   IER=0                                                             PRF 02
517   GO TO 525                                                         PRF 02
512   IER=1                                                             PRF 02
      GO TO 525                                                         PRF 02
516   IX=IY-1                                                           PRF 03
      IA=IANS                                                           PRF 03
519   II=IA+IX                                                          PRF 03
      ANS(IA)=AX(II)/D(IY)                                              PRF 03
      IF(IX) 990,990,991                                                PRF 03
991   DO 520 L=1,IX                                                     PRF 03
      LL=L-1+IA                                                         PRF 03
      AX(LL)=AX(LL)-ANS(IA)*D(L)                                        PRF 03
520   CONTINUE                                                          PRF 03
990   IA=IA-1                                                           PRF 03
      IF(IA) 521,521,519                                                PRF 03
521   IF(IX)522,522,523                                                 PRF 03
523   IF(ABSF(AX(IX))-TOL) 524,524,522                                  PRF 03
524   IX=IX-1                                                           PRF 03
      GO TO 521                                                         PRF 03
522   IANS=IANS-1                                                       PRF 03
      GO TO 515                                                         PRF 03
525   LL=IANS+1                                                         PRF 03
      DO 526 IB=1,LL                                                    PRF 03
526   AX(IB)=ANS(IB)                                                    PRF 03
      NNM=IANS                                                          PRF 03
      IF(I-IT)501,200,200                                               PRF 03
1001  CONTINUE                                                          PRF 03
205   STOP                                                              PRF 03
      END                                                               PRF 03
*******************************************************************************
```

APPENDIX - G

```
***************************************************************************
**                                                                   TRS 0C
C  C  S.K.SHARMA TRANSIENT RESPONSE CALCULATIONS AND PLOTTING          TRS 0C
      DIMENSION XINE(120),X13(120),X14(120)                           TRS 0C
      DIMENSION A(40),B(40),C(40),D(40),E(40),F(40),G(40),H(40)       TRS 00
      READ 1,BLANK,DOT,X,STAR                                         TRS 00
1     FORMAT (4A1)                                                    TRS 00
      READ 2,TMAX,XMAX,N,M,L,KL                                       TRS 0C
2     FORMAT(2F10.3,4I10)                                             TRS 00
97    IF(L-2) 96,120,120                                              TRS 00
**                                                                   TRS 00
      DO 4006 JK=1,KL                                                 TRS 00
      READ 4002,A(JK),B(JK),C(JK),D(JK)                               TRS 00
4002  FORMAT(4F15.4)                                                  TRS 00
4006  CONTINUE                                                        TRS 00
      DO 4007 JK=1,KL                                                 TRS 00
      READ 4004,E(JK),F(JK),G(JK),H(JK)                               TRS 00
 4004 FORMAT(4F15.4)                                                  TRS 00
4007  CONTINUE                                                        TRS 00
96    IF(L) 229,230,231                                               TRS 00
229   PUNCH 232                                                       TRS 00
232   FORMAT (10X,9HTRANSIENT,1X,9HRESPONSES,1X,3HFOR,1X,5HFIRST,1X,6 TRS 00.
     1 HOUTPUT)                                                       TRS 00.
      GO TO 95                                                        TRS 00.
230   PUNCH 233                                                       TRS 00.
233   FORMAT (10X,9HTRANSIENT,1X,9HRESPONSES,1X,3HFOR,1X,6HSECOND,1X,6 TRS 00.
     1HOUTPUT)                                                        TRS 00.
      GO TO 95                                                        TRS 00.
231   PUNCH 234                                                       TRS 00.
234   FORMAT (10X,9HTRANSIENT,1X,9HRESPONSES,1X,3HFOR,1X,5HTHIRD,1X,6 TRS 00.
     1HOUTPUT)                                                        TRS 00.
      IF(TMAX) 95,120,95                                              TRS 00.
95    JMAX=2*N+1                                                      TRS 00.
      DO 100 J=1,JMAX                                                 TRS 00.
100   XINE(J)=DOT                                                     TRS 00.
      PUNCH 3,(XINE(J),J=1,JMAX)                                      TRS 00.
3     FORMAT (1H*,79A1)                                               TRS 00.
      DO 105 J=1,JMAX                                                 TRS 00.
105   XINE(J)=BLANK                                                   TRS 00.
      T=0.0                                                           TRS 00.
      RM=M                                                            TRS 00.
      RN=N                                                            TRS 004
      DELT=TMAX/RM                                                    TRS 004
      DO 110 K=1,M                                                    TRS 004
      T=T+DELT                                                        TRS 004
      IF(L-2) 101,120,120                                             TRS 004
101   IF(L) 201,301,401                                               TRS 00.
301   XMAX=2.*XMAX                                                    TRS 00.
      GO TO 201                                                       TRS 00.
401   XMAX=3.*XMAX                                                    TRS 00.
201   X13R=0.                                                         TRS 004
      X13I=0.                                                         TRS 00.
      DO 4001 JK=1,KL                                                 TRS 05.
      RX=EXPF(A(JK)*T)*(B(JK)*COSF(C(JK)*T)-D(JK)*SINF(C(JK)*T))      TRS 00.
      X13R=X13R+RX                                                    TRS 00.
      RY=EXPF(A(JK)*T)*(D(JK)*COSF(C(JK)*T)+B(JK)*SINF(C(JK)*T))      TRS 00.
4001  X13I=X13I+RY                                                    TRS 00.
```

```
      X13(K)=SQRTF(X13R**2+X13I**2)                                     TRS 0(
      X14R=0.                                                           TRS 0(
      X14I=0.                                                           TRS 0(
      DO 4003 JK=1,KL                                                   TRS 0(
      RZ=EXPF(E(JK)*T)*(F(JK)*COSF(G(JK)*T)-H(JK)*SINF(G(JK)*T))        TRS 0(
      X14R=X14R+RZ                                                      TRS 0(
      SZ=EXPF(E(JK)*T)*(H(JK)*COSF(G(JK)*T)+F(JK)*SINF(G(JK)*T))        TRS 0C
4003  X14I=X14I+SZ                                                      TRS 0C
      X14(K)=SQRTF(X14R**2+X14I**2)                                     TRS 0C
501   RI=RN*(X13(K)/XMAX)                                               TRS 0C
      SI=RI+1.                                                          TRS 0C
      XI=RN*(X14(K)/XMAX)                                               TRS 0C
      ZI=XI+1.                                                          TRS 0C
      IF(SI-79.) 25,25,26                                               TRS 0C
25    I=SI                                                              TRS 0C
      XINE(I)=X                                                         TRS 0C
      GO TO 29                                                          TRS 0C
26    I=N+1                                                             TRS 0C
      XINE(I)=BLANK                                                     TRS 0C
29    IF(ZI-79.) 27,27,28                                               TRS 0C
27    II=ZI                                                             TRS 0C
      XINE(II)=STAR                                                     TRS 0C
      GO TO 75                                                          TRS 0C
28    I=N+1                                                             TRS 0C
      XINE(I)=BLANK                                                     TRS 0C
75    PUNCH 4,(XINE(J),J=1,JMAX)                                        TRS 0C
4     FORMAT (1H*,79A1)                                                 TRS 0C
      DO 179 J=1,JMAX                                                   TRS 0C
179   XINE(J)=BLANK                                                     TRS 0C
110   CONTINUE                                                          TRS 00
      PUNCH 21                                                          TRS 0C
21    FORMAT (79(1H@))                                                  TRS 0C
      IF(L) 20,23,24                                                    TRS 00
20    PUNCH 22                                                          TRS 0C
22    FORMAT (5X,10HMAGNITUDES,1X,9HX.PLOTTED,1X,5HCURVE,1X,3HTHE,1X,   TRS 0C
     15HFIRST,1X,6HOUTPUT)                                              TRS 0C
      PUNCH 221,(X13(I),I=1,M)                                          TRS 00
221   FORMAT (8E10.3)                                                   TRS 00
      PUNCH 222                                                         TRS 00
222   FORMAT(5X,10HMAGNITUDES,1X,2HOF,1X,9H*.PLOTTED,1X,5HCURVE,1X,3HFORTRS 00
     1,1X,3HTHE,1X,5HFIRST,1X,6HOUTPUT)                                 TRS 00
      PUNCH 221,(X14(I),I=1,M)                                          TRS 00
      GO TO 120                                                         TRS 0C
23    PUNCH 224                                                         TRS 0C
224   FORMAT (5X,10HMAGNITUDES,1X,2HOF,1X,9HX.PLOTTED,1X,5HCURVE,1X,3   TRS 01
     1HFOR,1X,3HTHE,1X,6HSECOND,1X,6HOUTPUT)                            TRS 01
      PUNCH 221,(X13(I),I=1,M)                                          TRS 01
      PUNCH 226                                                         TRS 0
226   FORMAT (5X,9HMAGNITUDE, 1X,2HOF,1X,9H*.PLOTTED,1X,5HCURVE,1X,3HFORTRS 0.
     11X,3HTHE,1X,6HSECOND,1X,6HOUTPUT)                                 TRS 0.
      PUNCH 221,(X14(I),I=1,M)                                          TRS 0.
      GO TO 120                                                         TRS 01
24    PUNCH 227                                                         TRS 01
227   FORMAT (5X,9HMAGNITUDE, 1X,2HOF,1X,9HX.PLOTTED,1X,5HCURVE,        TRS 01
     1 1X,3HTHE,1X,5HTHIRD,1X,6HOUTPUT)                                 TRS 01
```

```
      PUNCH 221,(P13(I),I=1,M)                                                    TRS 011
      PUNCH 228                                                                   TRS 011
228   FORMAT (5X,9HMAGNITUDE, 1X,2HOF,1X,9H*.PLOTTED,1X,5HCURVE,                  TRS 011
     1 1X,3HTHE,1X,5HTHIRD,1X,6HOUTPUT)                                           TRS 011
120   STOP                                                                        TRS 011
      END                                                                         TRS 011
*******************************************************************************************
```

RESULTS OF SRD ALGORITHM **********************************

****

ORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DINOMINATOR

4 4

SCALAR RETURN DIFFERENCE NUMERATOR AND DINOMINATOR RESPECTIVELY

```
   1.000      .738     6.174     2.861     2.935     1.000      .738      1.1
    .111     0.000
```

****

. SYSTEM IS UNSTABLE

****

ORDERS OF THE NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT

27 27

NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPE CTIVELY

```
   0.000     0.000     0.000     0.000     0.000     0.000     2.426     14.5
  55.380   149.041   315.186   541.397   774.815   935.668   958.443    834.5
 613.869   378.414   191.788    77.886    24.299     5.421      .820      .0
    .005     0.000     0.000     0.000     0.000     0.000     5.000     33.5
 143.999   440.806  1073.495  2147.144  3635.502  5281.963  6657.291   7319.0
7035.034  5913.235  4328.874  2746.428  1492.637   686.434   260.606     79.5
  18.575     3.149      .373      .030      .001     0.000     0.000      0.0
```

****

ORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DINOMINATOR

4 4

SCALAR RETURN DIFFERENCE NUMERATOR AND DINOMINATOR RESPECTIVELY

```
   0.000     1.000    70.738    39.674    41.201     0.000     1.000       .7
   1.174      .111
```

****

SYSTEM IS STABLE

****

ORDERS OF THE NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT

27 27

NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPE CTIVELY

```
    0.000     0.000     0.000     0.000     0.000     0.000     0.000     33.9
  204.310   775.326  2086.578  4412.603  7579.563 10847.420 13099.368 13418.2
11683.396  8594.166  5297.809  2685.035  1090.411   340.188    75.907     11.4
    1.138      .070      .002     0.000     0.000     0.000     0.000     70.0
  470.120  2015.987  6171.288 15028.935 30060.023 50897.038 73947.491 93202.0
ER F8Z .10246657E+06Z                   Z
           98490.483 82785.303 60604.247 38449.997 20896.926  9610.077  3648.4
 1114.127   260.050    44.093     5.233      .420      .021     0.000      0.0
```

****

ORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DINOMINATOR

4 4

SCALAR RETURN DIFFERENCE NUMERATOR AND DINOMINATOR RESPECTIVELY

```
   0.000     1.000    20.738    12.174    11.851     0.000     1.000       .7
   1.174      .111
```

****

SYSTEM IS STABLE

****

ORDERS OF THE NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT

```
     27        27
---------
  NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPE CTIVELY
    0.000      0.000      0.000      0.000      0.000      0.000      0.000      9.7
   58.374    221.521    596.166   1260.744   2165.590    099.262   3742.677   3833.7
 3338.113   2455.476   1513.659    767.153    311.546     97.196     21.687      3.2
     .325       .020      0.000      0.000      0.000      0.000      0.000     20.0
  134.320    575.996   1763.225   4293.981   8588.578  14542.011  21127.854  26629.1
29276.163  28140.138  23652.943  17315.499  10985.713   5970.550   2745.736   1042.4
  318.322     74.300     12.598      1.495       .120       .006      0.000      0.0
*****************************************************************************
  ORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DINOMINATOR
      4         4
  -------
  SCALAR RETURN DIFFERENCE NUMERATOR AND DINOMINATOR RESPECTIVELY
    0.000      1.000     10.738      6.674      5.981      0.000      1.000       .7
    1.174       .111
  **********************************************************************
              SYSTEM           IS           STABLE
*****************************************************************************
  ORDERS OF THE NUM. AND DINO. OF THE MODIFIED DIAGONAL  ELEMENT
     27        27
---------
  NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPE CTIVELY
    0.000      0.000      0.000      0.000      0.000      0.000      0.000      4.8
   29.187    110.760    298.082    630.371   1082.794   1549.632   1871.338   1916.8
 1669.056   1227.738    756.829    383.576    155.773     48.598     10.843      1.6
     .162       .010      0.000      0.000      0.000      0.000      0.000     10.0
   67.160    287.998    881.612   2146.990   4294.289   7271.005  10563.926  13314.5
14638.081  14070.069  11826.471   8657.749   5492.856   2985.275   1372.868    521.2
  159.161     37.150      6.299       .747       .060       .003      0.000      0.0
*****************************************************************************
  ORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DINOMINATOR
      4         4
  -------
  SCALAR RETURN DIFFERENCE NUME9ATOR AND DINOMINATOR RESPECTIVELY
    0.000      1.000      5.738      3.924      3.046      0.000      1.000       .7
    1.174       .111
  **********************************************************************
              SYSTEM           IS           STABLE
*****************************************************************************
  ORDERS OF THE NUM. AND DINO. OF THE MODIFIED DIAGONAL  ELEMENT
     27        27
---------
  NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPE CTIVELY
    0.000      0.000      0.000      0.000      0.000      0.000      0.000      2.
   14.593     55.380    149.041    315.186    541.397    774.815    935.668    958.
  834.528    613.869    378.414    191.788     77.886     24.299      5.421       .
     .081       .005      0.000      0.000      0.000      0.000      0.000      5.
   33.580    143.999    440.806   1073.495   2147.144   3635.502   5281.963   6657.
 7319.040   7035.034   5913.235   4328.874   2746.428   1492.637    686.434    260.
   79.580     18.575      3.149       .373       .030       .001      0.000      0.
*****************************************************************************
  ORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DINOMINATOR
      4         4
```

```
--------
SCALAR RETURN DIFFERENCE NUMERATOR AND DINOMINATOR RESPECTIVELY
   1.000       .738      6.174      3.801      2.935      1.000       .738      1.17
    .111      0.000
*************************************************************************
                  SYSTEM            IS            UNSTABLE
**************************************************************************
ORDERS OF THE NUM. AND DINO. OF THE MODIFIED DIAGONAL  ELEMENT
        27         27
----------
NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPE CTIVELY
   0.000      0.000      0.000      0.000      0.000      2.934     17.022     65.09
 176.198    379.795    667.653    987.386   1241.626   1339.110   1243.341    991.11
 677.050    390.885    189.016     74.079     22.979      5.222       .811       .08
    .005      0.000      0.000      0.000      0.000      0.000      5.000     34.52
 149.795    464.138   1140.132   2298.615   3916.741   5721.843   7243.280   7990.10
7697.951   6477.301   4741.194   3002.374   1626.163    743.370    279.961     84.50
  19.464      3.259       .382       .030       .001      0.000      0.000      0.00
**************************************************************************
ORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DINOMINATOR
        4          4
--------
SCALAR RETURN DIFFERENCE NUMERATOR AND DINOMINATOR RESPECTIVELY
   0.000      1.000     70.738     52.834     41.201      0.000      1.000       .73
   1.174       .111
*************************************************************************
                  SYSTEM            IS            STABLE
**************************************************************************
ORDERS OF THE NUM. AND DINO. OF THE MODIFIED DIAGONAL  ELEMENT
        27         27
----------
NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPE CTIVELY
   0.000      0.000      0.000      0.000      0.000      0.000     41.090    238.32
 911.370   2466.819   5317.194   9347.215  13823.475  17382.831  18747.581  17406.80
13875.589  9478.706   5472.400   2646.234   1037.114    321.710     73.110     11.36
   1.154       .072       .002      0.000      0.000      0.000      0.000     70.00
ER F8Z .10140592E+06Z                    Z
  483.280   2097.132   6497.939  15961.847  32180.616  54834.377  80105.808
ER F8Z .11186151E+06Z                    Z
ER F8Z .10777132E+06Z                    Z
                      90682.216  66376.729  42033.243  22766.294  10407.181   3919.46
 1183.111    272.506     45.637      5.361       .427       .021      0.000      0.00
**************************************************************************
ORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DINOMINATOR
        4          4
--------
SCALAR RETURN DIFFERENCE NUME9ATOR AND DINOMINATOR RESPECTIVELY
   0.000      1.000     20.738     15.934     11.851      0.000      1.000       .73
   1.174       .111
*************************************************************************
                  SYSTEM            IS            STABLE
**************************************************************************
ORDERS OF THE NUM. AND DINO. OF THE MODIFIED DIAGONAL  ELEMENT
        27         27
----------
NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPE CTIVELY
   0.000      0.000      0.000      0.000      0.000      -.005     11.721     68.04
```

```
   260.291    704.640   1518.969   2670.366   3949.300   4966.301   5356.294   4973.
  3964.411   2708.185   1563.538    756.065    296.318     91.917     20.888      3.
      .329       .020      0.000      0.000      0.000      0.000      0.000     20.
   138.080    599.180   1856.554   4560.528   9194.462  15666.964  22887.373  28973.
 31960.434  30791.805  25909.204  18964.779  12009.497   6504.655   2973.480   1119.
   338.031     77.859     13.039      1.531       .122       .006      0.000      0.
***********************************************************************************
  ORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DINOMINATOR
         4          4
  --------
  SCALAR RETURN DIFFERENCE NUMERATOR AND DINOMINATOR RESPECTIVELY
     0.000      1.000     10.738      8.554      5.981      0.000      1.000       [illegible]
     1.174       .111
  ****************************************************************
             SYSTEM           IS          STABLE
***********************************************************************************
  ORDERS OF THE NUM. AND DINO. OF THE MODIFIED DIAGONAL  ELEMENT
        27         27
---------
  NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPE CTIVELY
     0.000      0.000      0.000      0.000      0.000      0.000      5.867     34.
   130.181    352.379    759.566   1335.278   1974.745   2483.230   2678.204   2486.
  1982.220   1354.098    781.770    378.033    148.159     45.958     10.444      1.
      .164       .010      0.000      0.000      0.000      0.000      0.000     10.
    69.040    299.590    928.277   2280.263   4597.231   7833.482  11443.686  14486.
 15980.217  15395.903  12954.601   9482.390   6004.749   3252.327   1486.740    559.
   169.015     38.929      6.519       .765       .061       .003      0.000      0.
***********************************************************************************
  ORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DINOMINATOR
         4          4
  --------
  SCALAR RETURN DIFFERENCE NUMERATOR AND DINOMINATOR RESPECTIVELY
     0.000      1.000      5.738      4.864      3.046      0.000      1.000       .7
     1.174       .111
  ****************************************************************
             SYSTEM           IS          STABLE
***********************************************************************************
  ORDERS OF THE NUM. AND DINO. OF THE MODIFIED DIAGONAL  ELEMENT
        27         27
--------
  NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPE CTIVELY
     0.000      0.000      0.000      0.000      0.000      0.000      2.934     17.
    65.094    176.195    379.791    667.649    987.382   1241.623   1339.107   1243.
   991.112    677.049    390.885    189.016     74.079     22.979      5.222       .
      .082       .005      0.000      0.000      0.000      0.000      0.000      5.
    34.520    149.795    464.138   1140.132   2298.615   3916.741   5721.843   7243.
  7990.108   7697.951   6477.301   4741.194   3002.374   1626.163    743.370    279.
    84.507     19.464      3.259       .382       .030       .001      0.000      0.
***********************************************************************************
  ORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DINOMINATOR
         4          4
  --------
  SCALAR RETURN DIFFERENCE NUMERATOR AND DINOMINATOR RESPECTIVELY
     1.000       .738      6.174      1.051      0.000      1.000       .738      1.
      .111      0.000
  *************************@*************************************
             SYSTEM           IS          UNSTABLE
***********************************************************************************
  ORDERS OF THE NUM. AND DINO. OF THE MODIFIED DIAGONAL  ELEMENT
        27         27
---------
  NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPE CTIVELY
     0.000      0.000      0.000      0.000      0.000      5.869     34.044    127.
   336.962    703.739   1193.383   1688.589   2016.516   2043.749   1761.999   1283.
```

```
 785.457   395.605   160.860    50.663    11.814     1.975      .231       .
   0.000     0.000     0.000     0.000     0.000     0.000     5.000     31.
 129.903   377.782   872.333  1647.420  2621.025  3556.838  4155.496   4197.
3663.263  2757.257  1773.735   966.192   436.377   159.887    45.564      9.
   1.476      .158      .011     0.000     0.000     0.000     0.000      0.
*****************************************************************************
 ORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DINOMINATOR
       4         4
 --------
 SCALAR RETURN DIFFERENCE NUMERATOR AND DINOMINATOR RESPECTIVELY
   0.000     1.000    70.738    14.334      .111     0.000     1.000      .7
   1.174      .111
 *********************************************************************
             SYSTEM          IS          STABLE
*****************************************************************************
 ORDERS OF THE NUM. AND DINO. 6F THE MODIFIED DIAGONAL  ELEMENT
      27        27
---------
 NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPE CTIVELY
   0.000     0.000     0.000     0.000     0.000     0.000    82.180    476.
1782.927  4717.555  9852.464 16707.500 23640.389 28231.341 28612.566 24668.
17971.965 10996.407  5538.479  2252.042   709.283   165.396    27.661      3.
    .258      .013     0.000     0.000     0.000     0.000     0.000     70.
 444.780  1818.651  5288.949 12212.667 23063.882 36694.351 49795.740 58176.
58761.126 51285.694 38601.601 24832.302 13526.690  6109.286  2238.425    637.
 135.241    20.673     2.225      .163      .007     0.000     0.000      0.
*****************************************************************************
 ORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DINOMINATOR
       4         4
 --------
 SCALAR RETURN DIFFERENCE NUMERATOR AND DINOMINATOR RESPECTIVELY
   0.000     1.000    20.738     4.934      .111     0.000     1.000      .7
   1.174      .111
 *********************************************************************
             SYSTEM          IS          STABLE
*****************************************************************************
 ORDERS OF THE NUM. AND DINO. OF THE MODIFIED DIAGONAL  ELEMENT
      27        27
---------
 NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPE CTIVELY
   0.000     0.000     0.000     0.000     0.000     0.000    23.480    136.
 509.407  1347.872  2814.990  4773.571  6754.397  8066.097  8175.019   7048.
5134.847  3141.830  1582.422   643.440   202.652    47.256     7.903       .
    .073      .003     0.000     0.000     0.000     0.000     0.000     20.
 127.080   519.614  1511.128  3489.333  6589.680 10484.100 14227.354  16621.
16788.893 14653.055 11029.029  7094.943  3864.768  1745.510   639.550    182.
  38.640     5.906      .635      .046      .002     0.000     0.000      0.
*****************************************************************************
 ORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DINOMINATOR
       4         4
 --------
 SCALAR RETURN DIFFERENCE NUMERATOR AND DINOMINATOR RESPECTIVELY
   0.000     1.000    10.738     3.054      .111     0.000     1.000      .7
   1.174      .111
 *********************************************************************
             SYSTEM          IS          STABLE
```

****
ORDERS OF THE NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT
27 27

NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPE CTIVELY

0.000	0.000	0.000	0.000	0.000	0.000	11.740	68.0
254.703	673.936	1407.495	2386.785	3377.198	4033.048	4087.509	3524.0
2567.423	1570.915	791.211	321.720	101.326	23.628	3.951	.4
.036	.001	0.000	0.000	0.000	0.000	0.000	10.0
63.540	259.807	755.564	1744.666	3294.840	5242.050	7113.677	8310.9
8394.446	7326.527	5514.514	3547.471	1932.384	872.755	319.775	91.1
19.320	2.953	.317	.023	.001	0.000	0.000	0.0

****
ORDER OF SCALAR RETURN DIFFERENCE OF NUMERATOR AND DINOMINATOR
4 4

SCALAR RETURN DIFFERENCE NUMERATOR AND DINOMINATOR RESPECTIVELY

0.000	1.000	5.738	2.114	.111	0.000	1.000	.7
1.174	.111						

****
SYSTEM IS STABLE
****
ORDERS OF THE NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT
27 27

NUM. AND DINO. OF THE MODIFIED DIAGONAL ELEMENT RESPE CTIVELY

0.000	0.000	0.000	0.000	0.000	0.000	5.870	34.0
127.351	336.968	703.747	1193.392	1688.599	2016.524	2043.754	1762.0
1283.711	785.457	395.605	160.860	50.663	11.814	1.975	.2
.018	0.000	0.000	0.000	0.000	0.000	0.000	5.0
31.770	129.903	377.782	872.333	1647.420	2621.025	3556.838	4155.4
4197.223	3663.263	2757.257	1773.735	966.192	436.377	159.887	45.5
9.660	1.476	.158	.011	0.000	0.000	0.000	0.0

****

REFERENCES

1. D.Q. Mayne and S.C. Chuang " The sequential return difference method for designing linear multivariable Systems" Presented in the Conference " Computer Aided Control System Design", University of Cambridge, April 1973.

2. H.H.ROSENBROCK: " Design of multivariable control systems using the inverse Nyquist array" Proc. I.E.E. Vol.116, PP 1929, 1969.

3. D.Q. Mayne: " The effect of feedback on linear multivariable systems" Automatica, Vol.10, PP 405-412,1974.

4. A.G.J. MacFarlane, Return difference and return ratio matrices and their use in analysis and design of multivariable feedback control systems, Proc.IEE, Vol.117, PP 2037, 1970.

5. P.D. McMorran, S.M. Design of Gas turbine controller using inverse Nyquist method, Proc. IEE, Vol.117, PP 2050, 1970.

6. D.Q. Mayne, " The design of linear multivariable systems" Automatica, Vol.9, PP 201-207, 1973.

7. H.H. ROSENBROCK: The stability of multivariable systems, Trans. I.E.E.E. Aut. Control 16, 105, 1971.

8. E.J. DAVISON and H.W. SMITH, Pole assignment in linear time invariant multivariable systems with constant disturbances, Automatica, Vol.7, PP 489-498 July 1971.

9. JAYANTA PAL: Dissertation on Optimal Control of a Nuclear Reactor" for M.Sc. (Elect.Egg), 1973 Imperial College of Science and Technology, University of London.

10. Lunerberger, D.G. : Observers for multivariable systems - I.E.E.E. Trans., AC-11, PP 190-197,1966

11. G. Quazza : Automatic control in electric power systems: Automatica, Vol.6, PP 123, 1970.

12. W.M. Wonham : On pole assignment in multi-input controllable linear systems I.E.E.E. Tran. AC-12, PP 660-665, 1967.