AUSTRALIAN ATOMIC ENERGY COMMISSION
RESEARCH ESTABLISHMENT
LUCAS HEIGHTS

LINCAN - A PROGRAM FOR USE IN LINEAR DYNAMIC ANALYSIS

by

C.P. GILBERT

May 1971

ISBN 0 642 99426 9

AUSTRALIAN ATOMIC ENERGY COMMISSION
RESEARCH ESTABLISHMENT
LUCAS HEIGHTS

LINCAN – A PROGRAM FOR USE IN LINEAR DYNAMIC ANALYSIS

by

C. P. GILBERT

ABSTRACT

The facilities provided by LINCAN, a program for use in linear dynamic analysis, are described. These include the calculation of transfer functions from differential equations, the combination and scaling of transfer functions, the use of the Routh Criterion, polynomial factorisation, and the plotting of frequency and time responses.

The program is simple to use, and has been tested on systems whose orders exceed 40.

National Library of Australia card number and ISBN 0 642 99426 9

CONTENTS

Page

1. INTRODUCTION 1

2. GENERAL DESCRIPTION 1

3. OPERATIONS AVAILABLE 3

3.1 Calculation of Transfer Functions from State Equations 3
3.2 Transfer Function Manipulation 4

3.2.1 Representation 4
3.2.2 Basic manipulation 5
3.2.3 Storage 6
3.2.4 Division 6
3.2.5 Scaling 7

3.3 Frequency Response 7
3.4 Factorisation 7
3.5 Cancellation 8
3.6 Routh Criterion 8
3.7 Time Response 8

4. INPUT AND OUTPUT REQUIREMENTS 9

4.1 General Input Conditions 9
4.2 Detailed Data Requirements 10
4.3 Data Output 10
4.4 Sample Calculation 10

5. ERROR CHECKING 11

5.1 General Conditions 11
5.2 Scaling 12
5.3 General Guidance 12

6. ACKNOWLEDGMENT 12

7. REFERENCES 12

APPENDIX I Reduction of Higher-order Equations to sets of First Order Equations

APPENDIX II Computer Input and Output for the Sample Calculation of Section 4.4

Table I LINCAN Operations
Table II Transfer Function Manipulation – Data Requirements
Table III Data for Differential Equation Reduction
Table IV Data for Analysis and Output Operations

Figure 1 Simplified Block Diagram of LINCAN (see page 2)
Figure 2 Multivariable System (Equation 1) (see page 3)
Figure 3 Series Combination of Transfer Functions (U = 1) (see page 5)
Figure 4 Parallel Combination of Transfer Functions (U = 2) (see page 5)
Figure 5 Feedback Combination of Transfer Functions (U = 3) (see page 6)
Figure 6 Block Diagram of System used for the Sample Calculation (see page 11)

1. INTRODUCTION

The program LINCAN provides facilities for the calculations associated with the linear dynamic analysis of single- or multi-variable feedback systems. The basic operations which can be carried out are:

(i) Calculation of transfer function coefficients from sets of differential equations.

(ii) Combination and scaling of transfer functions.

(iii) Application of the Routh Criterion.

(iv) Factorisation of polynomials.

(v) Calculation and plotting of frequency responses.

(vi) Calculation and plotting of time responses.

These operations can be carried out in any order using input data or the results of previous calculations.

Work on LINCAN was initiated in 1966 and an early version written for the IBM 7040 was described by Gilbert (unpublished AAEC report 1967). LINCAN has since been used in a very wide variety of applications, and this experience has guided its further development. Care has been taken to ensure correct operation even with high-order systems (systems involving s^n where $n > 40$), but at the same time the program is very simple to use in straightforward situations.

LINCAN is available on disc on the IBM360/50 computer and uses double-precision representation throughout. Object and Fortran source decks can be provided, but owing to the use of facilities not always available elsewhere, the program will not necessarily work on other similar computers without slight modification.

This report is intended as a LINCAN users' guide. After a general description in Section 2, the individual facilities are described in Section 3, and Section 4 details input and output requirements. Typical computing times are given for the more lengthy operations, and a complete sample calculation is provided. Section 5 is a brief discussion of error checking, which may be necessary with high-order systems.

2. GENERAL DESCRIPTION

LINCAN essentially comprises a number of basic operations which can be applied in any sequence to suitable input data or to the results of previous calculations. Any given operation is called for by specifying the value of a Guide Number U in the input data, the values ranging from 0 to 60. At the conclusion of each operation the program returns to the input data to find the next value of U, and continues until the data are finished. Each card specifying a value of U must be followed by the data required for that particular operation.

The general process of using LINCAN can be visualised by referring to the simplified block diagram (Figure 1) showing the major operations separated into three broad groups. No attempt has been made to show the control exercised by the successive values of U. Data are handled mainly in the form of transfer functions, that is, ratios of polynomials in s relating two particular variables.

Within the INPUT group of operations there are two methods of providing input data; transfer functions are either read directly, or derived from a set of differential equations.

After the transfer functions have been provided they can be manipulated in a variety of ways depending upon the operations called for; for instance they can be scaled, divided, or combined in series, parallel or a feedback connection. These operations which come within the MANIPULATION group of Figure 1, may involve further input data, or the result of the immediately previous operation, or the results of some earlier calculations. This last type of process is facilitated by providing Stores where transfer functions can be left while other processes are carried out.

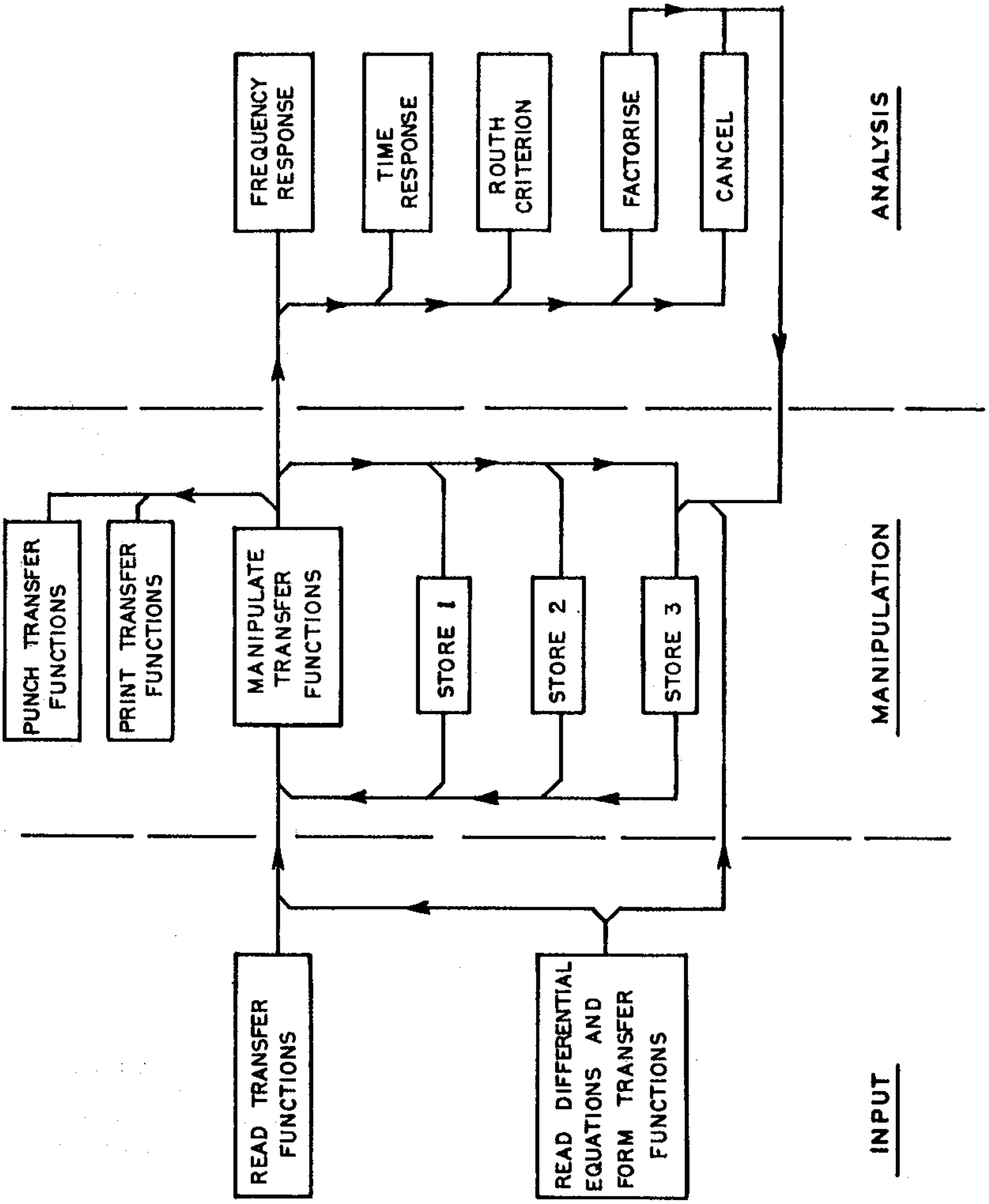


FIGURE 1. SIMPLIFIED BLOCK DIAGRAM OF LINCAN

At any stage a transfer function can be analysed using one of the operations shown in the ANALYSIS group; the results are always printed out, and may also be plotted or punched on cards. The use of these operations in no way affects the existing transfer functions in the program, and any manipulation can be resumed immediately. The only exceptions to this are the Factorise and Cancel operations, which place new data in Store 3; this store is also used during the reduction of the differential equations to the transfer function form.

The individual operations, which do not in all cases correspond exactly to any particular block in Figure 1, are listed in Table I which also gives the Section in which each operation is described, and the table in which the input requirements are detailed.

3. OPERATIONS AVAILABLE

3.1 Calculation of Transfer Functions from State Equations

Consider a linear system (Figure 2) with input variables y and output variables x.

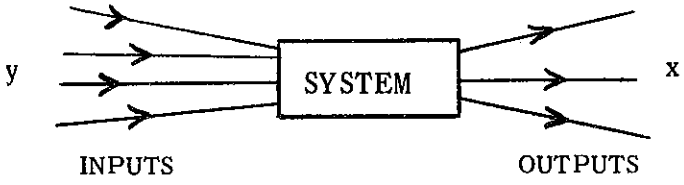


Figure 2 Multivariable System (Equation 1)

If such a system is described by differential and algebraic equations, LINCAN will calculate the transfer function between any given input-output pair as the ratio of two polynomials in the differential operator s. These operations correspond to the lower left hand block of Figure 1.

The equations must be written in matrix form so that:

(a) only the terms containing the input variables y are on the right hand side,

(b) no equation is of higher than first order in s, and

(c) all s terms lie on the leading diagonal of the matrix on the left hand side.

Any practical linear system can be described in this way. Appendix I shows how equations of higher order in s can be reduced to sets of first order equations to satisfy condition (b), and meeting condition (c) is then only a matter of rearrangement.

A simple example of a set of equations in the required form is

$$\begin{bmatrix} s+6 & 3 & 5 & 1 \\ 0 & 1 & 2 & 4 \\ 10 & 0 & 2s+3 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 3 \\ 4 & 1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \qquad (1)$$

Then by Kramer's Rule

$$\frac{x_3}{y_1} = \frac{g_1 + g_2 s}{a_1 + a_2 s + a_3 s^2} = \frac{\begin{vmatrix} s+6 & 3 & 1 & 1 \\ 0 & 1 & 0 & 4 \\ 10 & 0 & 4 & 2 \\ 4 & 3 & 3 & 1 \end{vmatrix}}{\begin{vmatrix} s+6 & 3 & 5 & 1 \\ 0 & 1 & 2 & 4 \\ 10 & 0 & 2s+3 & 2 \\ 4 & 3 & 2 & 1 \end{vmatrix}} \qquad (2)$$

LINCAN calculates and prints out the values of the polynomial coefficients a and g.

For input purposes equations such as (1) are described by the matrix of constant (s^0) coefficients, including those associated with the inputs y, and a vector of coefficients associated with the s terms on the leading diagonal, numbered from the top left-hand corner. In this case the size of the matrix is (4 x 6) and the 24 constant coefficients would be read in by columns as (6. 0. 10. 4. 3. 1. . . . . . 3. 2. 3. 1. 2.). The s-vector has four terms (1.0 0.0 2.0 0.0). These data can be seen in Appendix 2 as part of the example of Section 4.4.

In such a case all the coefficients would be read in using U = 44; details are summarised in Table III. For a sparse matrix, only the location and value of the non-zero terms need to be supplied (U = 45). Similarly, U = 46 allows specified coefficients in an existing matrix to be replaced with new values, U = 47 permitting a similar modification to the s-vector.

To specify the particular input and output variables defining the required transfer function, U = 48 is used. A number of transfer functions can be derived from a given set of equations (8 in the example) all having the same denominator. U = 48 may be applied as many times as required, and with the exception of those given below, any operation may be performed between the input of the original matrix data and the use of U = 48. As all the transfer functions have the same denominator, this is left in Store 3 (see Section 3.2.3). Thus the operations which may not precede U = 48 for a given matrix are those involving Store 3 (U = 30 to 38), factorisation (U = 41 or 51) and cancellation (U = 42 or 43).

LINCAN is dimensioned to accept a (60 x 70) matrix. It has been tested on a (54 x 55) matrix with 205 coefficients representing a 44th order system, and the computing time to find a single transfer function was 80 seconds. The transfer function coefficients had amplitudes over the range $10^{60} : 1$.

3.2 Transfer Function Manipulation

The operations described in this section correspond to the parts of Figure 1 devoted to manipulation and the direct reading of transfer functions.

3.2.1 Representation

Transfer functions are handled in the form of ratios of polynomials in s: the polynomials may have up to 100 coefficients, all of which must be real, and unless otherwise stated all processes are dimensioned for polynomials of up to this size.

Polynomials are specified by an integer giving the number of coefficients (order +1), followed by the coefficients themselves starting with that of lowest order. Thus ($X_5 s^4 + X_4 s^3 + X_3 s^2 + X_2 s + X_1$) would be specified by (5 X_1 X_2 X_3 X_4 X_5). It can be referred to as polynomial X with NX coefficients, a notation which is used below.

Any transfer function (A/B) is specified denominator first when read in or printed out (NB B_1 B_2 .... B_{NB} NA A_1 A_2 ....A_{NA}) the numbers being delimited on the cards by one or more spaces only. All coefficients must be specified, and in the unusual case of a transfer function with terms of higher order in the numerator than in the denominator the latter must be extended with zero coefficients until it is of the same order as the numerator.

Thus the transfer function of a simple differentiator can be expressed

$$s = \frac{s}{1} = \frac{0 + s}{1 + 0s}$$

and would be specified as (2 1.0 0.0 2 0.0 1.0) as shown in the example of Section 4.4.

3.2.2 Basic manipulation

The calculated, or 'output' transfer function X/Y is always printed out, and if U = 8 is specified the last calculated value of X/Y is also punched out, the cards being suitable for use as input data.

The instruction U = 0 simply provides a means for reading in a transfer function A/B to replace the existing output. Thus

$$\frac{X}{Y} = \frac{A}{B}$$

When U = 1, two transfer functions A/B and C/D are read in and combined in series as in Figure 3.

$$\frac{X}{Y} = \pm \frac{A}{B} \times \frac{C}{D} \tag{3}$$

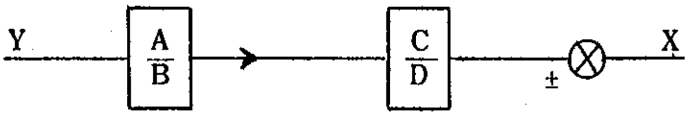


Figure 3 Series Combination of Transfer Functions (U = 1)

When U = 2, the two input transfer functions are combined in parallel, as in Figure 4:

$$\frac{X}{Y} = \frac{A.D \pm B.C}{B.D} \tag{4}$$

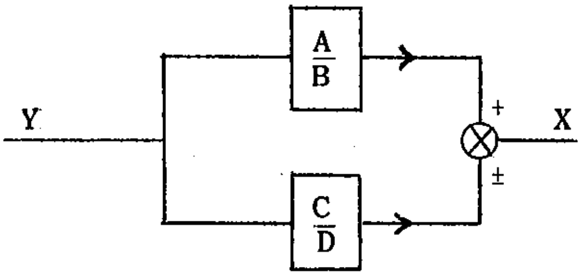


Figure 4 Parallel Combination of Transfer Functions (U = 2)

When U = 3, the input transfer functions form the feedback loop of Figure 5, and the closed-loop transfer function is found:

$$\frac{X}{Y} = \frac{A.D}{B.D \pm A.C} \tag{5}$$

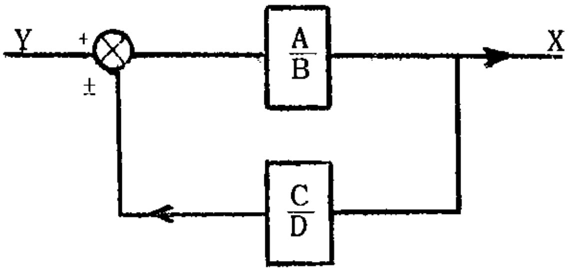


Figure 5 Feedback Combination of Transfer Functions (U = 3)

In equations (3)–(5), the positive sign is always used unless U = 9 is specified before a particular operation: the negative sign is then used for this one operation only.

The three basic operations can also be carried out using a single input transfer function combined with the result of the immediately preceding calculation X'/Y'. These operations use U = 4 to 7 as detailed in Table II.

3.2.3 Storage

To permit transfer functions to be retained and used in later calculations, there are three Stores whose contents can be used as though they were the input data described in Section 3.2.2. U = 10 to 18 operate on Store 1, U = 20 to 28 on Store 2 and U = 30 to 38 on Store 3, as detailed in Table II. It should be noted that some operations leave data in Store 3, and so this Store should not contain a wanted transfer function during such operations. The operations involved are those referred to in Section 3.1 and in Figure 1, that is the reduction of transfer functions from differential equations (U = 44 to 48), factorisation (U = 41 or 51) and cancellation (U = 42 or 43). There are no restrictions on the use of Stores 1 and 2.

3.2.4 Division

The operations described above sometimes result in transfer functions of higher order than is essential. For instance, in equation (4) if B = D, X/Y=(A+C)/B, but the use of U = 2 gives the full output specified above. The transfer function can be reduced to its simpler form by dividing by B/B, using the methods described below.

With U = 53 the previous output transfer function X'/Y' can be divided by an input transfer function A/B, giving the output

$$\frac{X}{Y} = \frac{X'}{Y'} \div \frac{A}{B} = \frac{X'/A}{Y'/B} \tag{6}$$

This process reduces the order of the transfer function i.e.

$$NY = NY' - NB + 1$$

Using U = 52, X'/Y' may be divided by the contents of Store 3. In each case the denominator and numerator remainders are printed out. Input requirements are detailed in Table II.

Division is equivalent to splitting a transfer function into two series elements, and one or both of these will be unstable if the original transfer function was unstable.

3.2.5 Scaling

To provide some control over the size of the transfer function coefficients, amplitude scaling can be achieved. Following U = 19, a single number is specified. The largest coefficient is made equal to this number and all the other coefficients are then similarly scaled so that the transfer function itself is effectively unchanged.

In time scaling the relative sizes of the coefficients are changed. However, LINCAN employs a process which also uses amplitude scaling and this ensures that no coefficient is made larger. Following U = 60, a number is specified (τ say). In the transfer function the coefficients of s^n are effectively multiplied by τ^n, so that the time response is τ times slower ($1/\tau$ times faster).

In both types of scaling the output transfer function X/Y is the scaled version of the previous output X'/Y'. Input requirements are summarised in Table IV, and some important applications of scaling are discussed in Section 5.

3.3 Frequency Response

This, together with the operations described in the remainder of Section 3, belongs to the Analysis group of Figure 1.

A frequency response is obtained when U = 49 is specified, followed by numbers representing a Low Frequency, a Ratio (> 1), and a High Frequency (see Table IV). The next transfer function to be calculated is then evaluated in terms of gain and phase as a function of $s = j\omega$. The process starts at the Low Frequency, and proceeds to the High Frequency, the value of ω being multiplied by Ratio at each step. The result is printed and plotted. The gain and phase at $\omega = 0$ are also found, the phase being assumed to be within the range ± 180°.

If the phase changes by more than 50° between two successive values of ω, the Ratio is reduced to the square root of its existing value: this may happen a number of times. Subsequently, whenever the phase change is less than 10° the Ratio is increased, but it is not permitted to exceed the value originally specified.

The frequency response corresponding to the 44th order transfer function evaluated in Section 3.1 was calculated at 53 frequencies over a range of $10^8 : 1$. The computing time was 36 seconds.

3.4 Factorisation

Two methods of factorisation are available, both of which operate on the last calculated transfer function. Both print out the roots of the denominator and the numerator, and by multiplication of the factors reconstitute the polynomials. The synthesized coefficients are printed beside the original coefficients as a check that the operation has been successful. The original transfer function is left unchanged as X/Y, and the synthesized transfer function is placed in Store 3, from which it can be entered into subsequent calculations.

Factorisation is only an approximate process, and may fail altogether in difficult circumstances. Unfortunately a simple comparison of the original and synthesized coefficients is not a completely reliable test of the accuracy of the roots, since quite small coefficient errors may result from relatively large root errors.

For all normal purposes U = 51 should be specified. This uses a root squaring method dimensioned to receive polynomials of up to 50th order. If this process should fail or if the accuracy of the roots is suspect, U = 41 may be used for polynomials of up to 40th order (see Table IV). This makes use of quadruple precision arithmetic, and is very accurate, but very slow. In this case only, if U = 8 is specified the factors are punched out onto cards, any complex conjugate pairs being combined into second order terms.

In the example of Section 4.4 a sixth order denominator and a third order numerator were factorised in 14 seconds using U = 51.

3.5 Cancellation

In cases where factorisation is not successful, it is frequently useful to be able to reduce the order of a high order transfer function, even at the expense of some loss of accuracy. One way of doing this is to search for approximate common factors in the numerator and denominator using an algorithm due to Euclid. An integer provided with the input data specifies the number of attempts to be made to find the common factors, progressively larger errors being permitted at each successive attempt.

As would be expected in a process of this type, a useful result cannot always be obtained. Because of this the original transfer function is left unchanged as X/Y, and the residual transfer function remaining after cancellation is placed in Store 3. The common factor, the permitted error and the remainder polynomials are printed out. When U = 43, a common factor is divided out as soon as it is obtained, and the next attempt operates on the residual transfer function. When U = 42, a fresh start is made on the original transfer function at each attempt (Table IV).

Despite the approximations involved in this process, the residual transfer function may well be sufficiently accurate over the frequency range of interest, and this can be checked using U = 49.

3.6 Routh Criterion

In stability investigations, when factorisation cannot be achieved it is sometimes useful to apply the Routh Criterion. It is called for by U = 40 (Table IV). This process always works, but the result, which can only indicate an absolute stability boundary, is often of restricted value.

Normally the left-hand column of the Routh array is printed out, followed by 'STABLE' or 'UNSTABLE WITH n UNSTABLE ROOTS'. In some cases the array is terminated due to a zero term in the left-hand column, which normally indicates instability (Gille et al. 1959), or due to zero terms in the original denominator, which always indicate instability. The print-out makes it clear when this has occurred. Applying the Routh Criterion to the 44th order transfer function found as an example in Section 3.1 took 4 seconds, including the scaling operations described in Section 5.2.

3.7 Time Response

The transfer function is converted to the equivalent set of first-order differential equations, and the latter are solved using a fourth-order Runge-Kutta integration routine. Zero initial conditions are assumed. An impulse input at zero time is used if U = 54, a step if U = 55 and a ramp if U = 56 (see Table IV). The order of the transfer function numerator must not be greater than that of the denominator, and should the orders be equal a step input, not an impulse, is applied when U = 54. After the cards giving the value of U and the Title, the following five numbers must be read in:

TUNIT.	This is some convenient time unit: it is assumed that the differential equation coefficients use a time unit of 1 second, but the solution may be required over periods of milli seconds, (read in 0.001), hundreds of seconds (100), days (8.64E+4) etc.
FINTIM.	A final time at which the solution is stopped: this should be within the range 0.1 to 100 of the specified time units, and this helps to indicate a suitable size for the time unit.
STEPS/UNIT.	The minimum number of integration steps per time unit. (The program may increase this number).
ERROR.	The permissible relative error. This is normally specified as 10^{-7}, but some other value may improve any given computation.
PRINTS/UNIT.	The number of results printed out per time unit. This should not be more than STEPS/UNIT.

The time response is printed out and plotted. As some indication of computing time, the step response found in the example of Section 4.4 took 17 seconds.

4. INPUT AND OUTPUT REQUIREMENTS

4.1 General Input Conditions

All input data are read in using the subroutine SCAN (Bennett and Pollard 1967). This permits the data to be assembled in almost any form, with an unrestricted number of entries on a single card (up to column 72). For convenience, real constants should be distinguished from integers by a decimal point or the E or D format, but they will be read even without such an identification. Numbers are delimited simply by one or more spaces. Also, alphabetic matter is ignored when the program expects to read numbers and this can be used to identify the data. Thus, following U = 49 the frequency specification card could contain three numbers only. Alternatively it could contain:

```
LOW FREQ = 0.01          RATIO = 1.1          HIGH FREQ = 1.0E +2
```

The entries could also be inserted on three separate cards. Similarly, to define a matrix following U = 45, the input might be:

```
8 ROWS       10 COLUMNS       FOLLOWED BY 32 COEFFICIENTS

1    3    1.042D-4    1    7    -2.97    3    10    1.
```

and so on, until the locations and values of the 32 coefficients had been specified. Another feature is that SCAN ignores any punching on a card with * in column 1.

Additional facilities provided by SCAN are:

(i) the repeat notation: thus 5 * 1.0 is equivalent to 1.0 1.0 1.0 1.0 1.0

(ii) the increment notation, in which 0(2)10 is equivalent to 0 2 4 6 8 10, and 1.3(2.1)7.6 is equivalent to 1.3 3.4 5.5 7.6

(iii) adjoining increments, which are a logical extension of (ii). In this, 1(1)4(2)10(8)34 would be interpreted as 1 2 3 4 6 8 10 18 26 34.

As usual, these groups are delimited by one or more spaces. Apart from the number of repeats in (i), any of the numbers may be negative.

While such a free input is very convenient indeed, it allows the program to accept incorrect data which a fixed-format system would reject. This applies particularly when the <u>amount</u> of data presented is not the same as SCAN expects. For instance, if a polynomial is specified as having six coefficients, and only five are provided, the program will ignore all alphabetic matter and take the next number as the final coefficient, no matter what it was originally meant to be. If steps were not taken to prevent it, the program might then continue indefinitely misinterpreting the remaining data.

To avoid this situation, the program will only accept Guide Numbers when on a card of their own in the form

```
Ub=b00
```

where b represents a space. U must be in column 1, and 00 represents a two-digit integer in columns 5 and 6, of which column 5 must be blank when U has only one digit. Thus, should data misinterpretation occur the program will normally stop before the next operation begins. Also, many operations require a Title Card following the Guide Number; the title may have any form and is reproduced in the output for identification purposes. If an incorrect title is printed out this also indicates data misinterpretation.

4.2 Detailed Data Requirements

Data requirements are set out in Tables II, III and IV, the values of U in the three tables overlapping to some extent.

Table II relates to the operations in the Manipulation group of Figure 1, and to the direct reading of transfer functions. As indicated in Section 3.2.1, a polynomial is specified by an integer giving the number of coefficients, followed by the coefficients themselves in ascending power of s (see also the example in Section 4.4).

Only the data marked * are required, and the Guide Number and Title Card (if needed) should be as described in Section 4.1.

Table III details the data requirements for the input and subsequent reduction of sets of differential equations in matrix form (Section 3.1).

Table IV lists the data requirements for the remaining operations: these include the 'Analysis' operations of Figure 1, and scaling and output instructions.

4.3 Data Output

The data output may be in three forms: print-out, punched cards, or a plot.

The transfer function resulting from an operation is always printed, denominator first, normally following the title originating in the input data which demanded the operation. There is also a suitable print-out following frequency response calculations, time scaling etc., and these are essentially self explanatory. The standard output format is D11.4: if higher precision is required the use of U = 58 ensures that all following transfer functions are printed using D23.16 until standard precision is restored using U = 59.

If a card output is required it is called for by U = 8 which punches the last calculated transfer function. The only other output data which can be punched are the factors found with U = 41. The factors are always punched out with high precision : for the normal transfer functions, the precision is determined in the same way as for the print-out using U = 58 or U = 59. In all cases the cards can be used directly as data input.

Plotting is performed off-line whenever a frequency or a time response is calculated, and the plots usually become available shortly after the print-out. A CALCOMP plotter which gives a graph 11" x 16" is used.

4.4 Sample Calculation

To illustrate the general method of using LINCAN a simple example is given relating to the system of Figure 6, in which the block marked 'Plant' is defined by equation (1). The complete input and output are listed in Appendix II, only the job control cards preceding the data being necessary to call up LINCAN from disc.

Briefly, the calculations asked for by the input are as follows. First the transfer function from K to M (equation 2) is evaluated from the matrix data (equation 1). Since the denominator has terms of both sign it is clear that this part of the system is unstable; the program would note this if the Routh Criterion were requested. The transfer function from L to M is found by including the effect of the (positive) feedback via path 1. Placing this transfer function in Store 2 permits that of Path 2 to be evaluated from the input data, and the two are then combined in parallel to give the transfer function from P to N. By combining block N to R in series, the transfer function from P to R is found, and the frequency response of this section is printed out and plotted.

Next, negative feedback via Path 3 is applied, giving the transfer function from Q to R. If a feedback gain of 10 is used, the Routh Criterion shows that the system is unstable (as could have been predicted from the frequency response). With a gain of unity the denominator roots all have negative real parts and so the system is stable. The excellent agreement between the original and the synthesized transfer function coefficients suggest that the roots are accurate, and the time response to a step input gives a general picture of the system response.

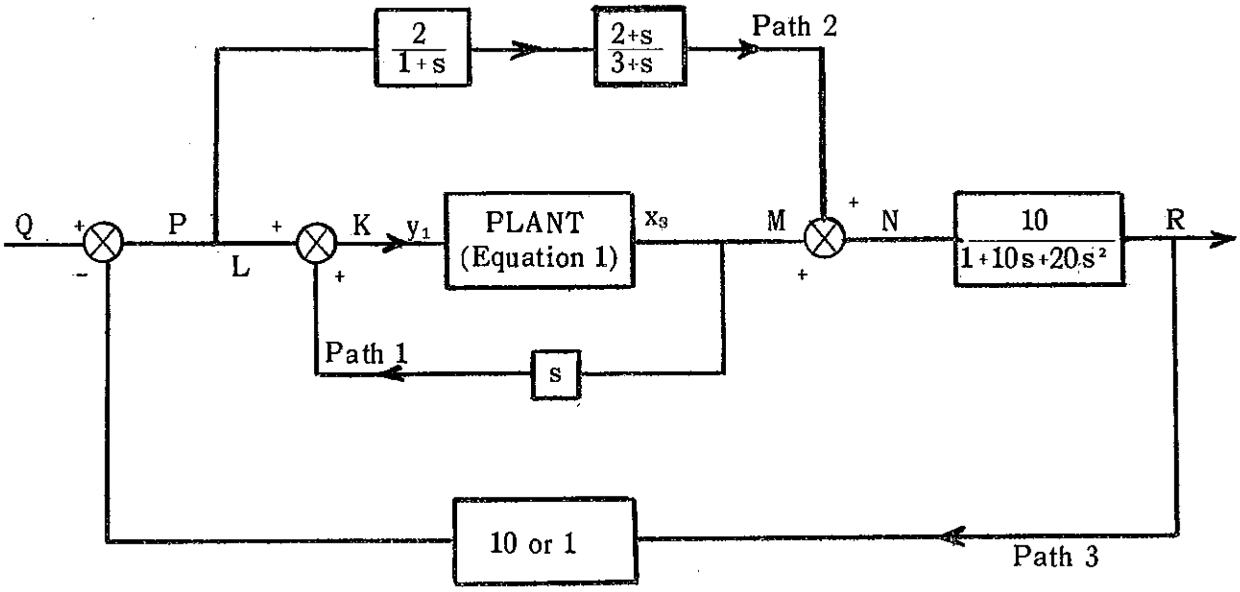


Figure 6 Block Diagram of System used for the Sample Calculation

This problem took 26 seconds of CPU time. LINCAN requires 288,000 storage locations, and this may determine the job classification under M.V.T. (Multi-programming with Variable number of Tasks.

5. ERROR CHECKING

5.1 General Conditions

Without adding considerably to the size, complexity and running time of a general purpose program such as LINCAN it is impossible to prevent the occasional use of input data which causes either underflows or overflows in some of the operations. Using the techniques discussed below such errors can be minimised and their effects assessed.

One exception to these considerations is the frequency response calculation. This basic process is inherently liable to give an excessive range of values, and automatic scaling has been built into the program to avoid such errors.

It must be stressed that unless unusually awkward numbers are involved, no precautions at all need be taken for small systems, those with orders less than 15 say, apart from checking the print-out for extreme amplitudes. Increasing care should be taken if transfer functions of higher order are involved, but in most cases no problems arise. In many cases underflows can be ignored, but an overflow usually indicates the presence of a serious error.

5.2 Scaling

Amplitude scaling of an equation or a transfer function has no effect upon the dynamic properties of the system and can be used to change the general level of coefficient sizes. As its name suggests, time scaling has the effect of speeding up or slowing down the system. Again this does not affect the stability properties, but it usually alters the range of amplitudes of the coefficients describing a given system. These two methods should be used to avoid excessively large or excessively small numbers (of the order of $10^{\pm 50}$).

For transfer functions, both forms of scaling can be applied by LINCAN (see Section 3.2.5). As an example, when the Routh Criterion was applied to a 44th order polynomial with a coefficient range of $10^{60}:1$ (see Section 3.1), underflows occurred and the process terminated due to a 'zero' in the left-hand column. When a time scale of 0.05 was used the coefficient range was reduced to $10^{14} : 1$, no underflows occurred and the correct result was obtained.

If the excessive numbers arise directly from the input data, the data must be scaled by the user. Time scaling is the more usual requirement, particularly for matrix input data. It is very easy to apply since all terms in the s-vector are altered by the same amount. For instance, the matrix reduction required to find the 44th order transfer function of Section 3.1 caused 143 underflows and gave a coefficient range of $10^{60} : 1$. By reducing each s-vector term by a factor of 2.5 the number of underflows was reduced to 35, and the coefficient range was only $10^{44} : 1$. Subsequent time scaling returned the polynomial to its original form. There was no difference between the polynomials found by the two processes (D 11.4), both agreeing with the correct result, but the method outlined for reducing the number of underflows can be very useful.

5.3 General Guidance

If there is any doubt as to the correctness of a result, a repeat calculation with a scaled version of the transfer functions should be tried.

In some cases a calculation can be performed in a different way to provide a check. For instance the two different methods of factorisation can be used, or the time step or error criterion can be changed when calculating a time response. Alternatively the problem can sometimes be recast. As an example, the differentiator of Path 1 (Figure 6) could be introduced as a further row in the 'Plant' equations instead of as a feedback block : indeed, all the branches could be handled in this way, but the computing time would be considerably increased and the information relating to the sub-systems would be absent.

6. ACKNOWLEDGMENT

At various stages the author has received assistance from Mr. E. R. Corran who is, and Mr. P. Gaal who was, with the Instrumentation and Control Division. The author is also grateful to Mr. J. M. Barry and his staff of the Applied Mathematics and Computing Section: in particular, Mrs. S. G. Johnson wrote the subroutine which computes polynomial coefficients from the matrix data, and Mr. R. P. Backstrom provided the 'quadruple precision' routines.

7. REFERENCES

Bennett, N. W. and Pollard, J.P. (1967). – SCAN- A Free Input Subroutine for the IBM 360. AAEC/TM399.

Gille, Pelegrin and Decaulne (1959). Feedback Control Systems. McGraw Hill.

APPENDIX I

REDUCTION OF HIGHER-ORDER EQUATIONS TO SETS OF FIRST ORDER EQUATIONS

The method of reducing a higher order system to a set of first order differential equations is illustrated by means of an example.

Given the system

$$\begin{bmatrix} s+6 & 2 & 5 \\ 1 & \dfrac{3s^3+s^2+2s+5}{4s^3+2s^2+s+2} & 3 \\ 4 & 3 & s \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

then the equation which has to be rearranged is

$$x_1 + \left[\frac{3s^3+s^2+2s+5}{4s^3+2s^2+s+2}\right] x_2 + 3x_3 = 4y_1 + y_2$$

Multiplying through by the denominator leads to

$$(3s^3+s^2+2s+5)x_2 = (4s^3+2s^2+s+2)(4y_1+y_2-x_1-3x_3),$$

and so the remainder of the rearrangement process applies to any differential equation of this type. Separation into powers of s gives

$$s^3\,(3x_2+4x_1+12x_3-16y_1-4y_2) = s^2(8y_1+2y_2-2x_1-x_2-6x_3)$$
$$+s(4y_1+y_2-x_1-2x_2-3x_3)+(8y_1+2y_2-2x_1-5x_2-6x_3)\ .$$

Then dividing by $3s^3$ and rearranging so as to isolate x_2 leads to

$$x_2 = \frac{1}{3}\,(16y_1+4y_2-4x_1-12x_3+\alpha_1)$$

where $\alpha_1 = \dfrac{1}{s}\,(8y_1+2y_2-2x_1-x_2-6x_3+\alpha_2)$,

$$\alpha_2 = \frac{1}{s}\,(4y_1+y_2-x_1-2x_2-3x_3+\alpha_3)\ ,$$

and $\alpha_3 = \dfrac{1}{s}\,(8y_1+2y_2-2x_1-5x_2-6x_3)$.

None of these equations is of higher than first order, and so if the variables α_1, α_2 and α_3 are introduced the original system can be rewritten as

$$\begin{bmatrix} s+6 & 2 & & & & 5 \\ 4 & 3 & -1 & & & 12 \\ 2 & 1 & s & -1 & & 6 \\ 1 & 2 & & s & -1 & 3 \\ 2 & 5 & & & s & 6 \\ 4 & 3 & & & & s \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 16 & 4 \\ 8 & 2 \\ 4 & 1 \\ 8 & 2 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

APPENDIX II

COMPUTER INPUT AND OUTPUT FOR THE SAMPLE CALCULATION OF SECTION 4.4

INPUT

JOB CONTROL

```
//
//CPGLIN   JOB G1B10413,C.P.G.GILBERT,
//             CLASS=J,TIME=12
//JOBLIB  DD  DSNAME=CPGDAG.LIB,UNIT=SYSDA,VOL=REF=PACK6,                *
//             DISP=(OLD,KEEP)
//        EXEC  BUFFPROG,PRG=AEPLOT
//        EXEC      PGM=LINCAN,REGION=300K
//FT01F001   DD   DDNAME=SYSIN
//FT02F001   DD   SYSOUT=B
//FT03F001   DD   SYSOUT=A
//FT06F001   DD   SYSOUT=A
//AEPLOT  DD  SYSOUT=C
//SYSIN  DD   *
```

DATA

```
U = 44
  4 ROWS   6 COLUMNS
MATRIX COEFFS  6.0   0.0   10.0   4.0   3.0   1.0   2.0   3.0   5.2
2.0   3.0   2.0   1.0   4.0   2.0   1.0   1.0   0.0   4.0   3.0
2.0   3.0   1.0   2.0
  S TERMS  1.0   0.0   2.0   0.0
U = 48
TRANSFER FUNCTION X3/Y1  FROM K TO M
OUTPUT X3  AND INPUT Y1
U =  9
U =  6
TRANSFER FUNCTION FROM L TO M WITH POSITIVE FEEDBACK VIA PATH 1
DIFFERFNTIATOR  2  1.0  0.0          2  0.0  1.0
U = 20
U =  1
TRANSFER FUNCTION OF PATH 2
 2  1.0  1.0    1  2.0
2  3.  1.      2  2.  1.
U = 25
COMBINE IN PARALLEL TO GIVE THE TRANSFER FUNCTION FROM P TO N
U = 49
LOWF = 0.0001   RATIO = 1.7783   HIGHF = 100.
U =  4
TRANSFER FUNCTION FROM P TO R, AND FREQUENCY RESPONSE
  3  1.  10.  20.       1  10.
U = 10
U =  6
TRANSFER FUNCTION FROM Q TO R INCLUDING NEGATIVE FEEDBACK VIA PATH 3
1  0.1   1  1.
U = 40
U = 13
REDUCE THE GAIN IN FEEDBACK PATH 3 BY A FACTOR OF 10
1 1.  1  1.
U = 51
U = 55
STEP RESPONSE OF LOW GAIN SYSTEM -
TIME UNIT IS  1.0 SEC   STOP AFTER 25. SECS   AT LEAST 10 STEPS/SEC
  ERROR  1.0E-7   AND 2 PRINTS PER SECOND
```

```
/*
//
//
//
//
//
//
//
```

(Title cards are underlined both here and in output)

OUTPUT

```
IEF298I  CPGLIN   SYSOUT=C.
//CPGLIN   JOB G1B12413,C.P.G.GILBERT,
//             CLASS=J,TIME=12
//JOBLIB  DD  DSNAME=CPGDAG.LIB,UNIT=SYSDA,VOL=REF=PACK6,                *
//             DISP=(OLD,KEEP)
//         EXEC  BUFFPROG,PRG=AEPLOT
XX         PROC  DSNP='PDP.LINKLIB',DSN='PDP.LINKLIB'                     00000010
XXPGM      EXEC  PGM=AEBUFF,REGION=52K                                    00000020
XXSTEPLIB  DD  DSN=&DSNP,DISP=SHR                                         00000030
IEF653I SUBSTITUTION JCL - DSN=PDP.LINKLIB,DISP=SHR
XXSYSLIB   DD  DSN=&DSN.(&PRG),DISP=SHR                                   00000040
IEF653I SUBSTITUTION JCL - DSN=PDP.LINKLIB(AEPLOT),DISP=SHR
XXSYSUT2   DD  SYSOUT=C                                                   00000050
IEF236I ALLOC. FOR CPGLIN   PGM
IEF237I 131   ALLOCATED TO JOBLIB
IEF237I 135   ALLOCATED TO STEPLIB
IEF237I 135   ALLOCATED TO SYSLIB
IEF237I 133   ALLOCATED TO SYSUT2
IEF285I   CPGDAG.LIB                              PASSED
IEF285I   VOL SER NOS= AAE006.
IEF285I   PDP.LINKLIB                             KEPT
IEF285I   VOL SER NOS= AAE003.
IEF285I   PDP.LINKLIB                             KEPT
IEF285I   VOL SER NOS= AAE003.
IEF285I   SYS71195.T112024.SV000.CPGLIN.R0000061       SYSOUT
IEF285I   VOL SER NOS= AAE005.
IEF373I STEP /PGM     / START 71195.1326
IEF374I STEP /PGM     / STOP  71195.1328 CPU   0MIN 00.48SEC MAIN  48K LCS   0K
//         EXEC      PGM=LINCAN,REGION=300K
//FT01F001   DD   DDNAME=SYSIN
//FT02F001   DD   SYSOUT=B
//FT03F001   DD   SYSOUT=A
//FT06F001   DD   SYSOUT=A
//AEPLOT  DD  SYSOUT=C
//SYSIN  DD   *
//
IEF236I ALLOC. FOR CPGLIN
IEF237I 131   ALLOCATED TO JOBLIB
IEF237I 134   ALLOCATED TO FT01F001
IEF237I 132   ALLOCATED TO FT02F001
IEF237I 133   ALLOCATED TO FT03F001
IEF237I 134   ALLOCATED TO FT06F001
IEF237I 132   ALLOCATED TO AEPLOT
```

```
TRANSFER FUNCTION X3/Y1  FROM K TO M

   ORDER  OF  Y    2
 -0.1382D+02   0.3136D+01   0.1000D+01

   ORDER  OF  X    1
  0.1527D+02   0.2273D+01


TRANSFER FUNCTION FRCM L TO M WITH POSITIVE FEEDBACK VIA PATH 1

   ORDER  OF  Y    2
 -0.1382D+02  -0.1214D+02  -0.1273D+01

   ORDER  OF  X    1
  0.1527D+02   0.2273D+01

                         ABCVE VALUES PLACED IN STORE 2
```

```
TRANSFER FUNCTION OF PATH 2

  ORDER   OF   Y     2
  0.3000D+01   0.4000D+01   0.1000D+01

  ORDER   OF   X     1
  0.4000D+01   0.2000D+01

COMBINE IN PARALLEL TO GIVE THE TRANSFER FUNCTION FROM P TO N

  ORDER   OF   Y     4
 -0.4145D+02  -0.5168D+02  -0.6618D+02  -0.1723D+02  -0.1273D+01

  ORDER   OF   X     3
 -0.9455D+01  -0.8273D+01  -0.5000D+01  -0.2727D+00
```

```
TRANSFER FUNCTION FROM P TO R, AND FREQUENCY RESPONSE

 ORDER  OF  Y    6
-0.4145D+02  -0.5062D+03  -0.1812D+04  -0.2513D+04  -0.1497D+04  -0.3573D+03  -0.2545D+02

 ORDER  OF  X    3
-0.9455D+02  -0.8273D+02  -0.5000D+02  -0.2727D+01

 RAD/SEC          AMPLITUDE          DB          PHASE ANGLE
0.0               0.2281D+01      0.7161D+01          0.0
0.1000D-03        0.2281D+01      0.7161D+01         -0.1
0.1778D-03        0.2281D+01      0.7161D+01         -0.1
0.3162D-03        0.2281D+01      0.7161D+01         -0.2
0.5624D-03        0.2281D+01      0.7161D+01         -0.4
0.1000D-02        0.2281D+01      0.7161D+01         -0.6
0.1778D-02        0.2280D+01      0.7161D+01         -1.2
0.3162D-02        0.2280D+01      0.7159D+01         -2.1
0.5624D-02        0.2278D+01      0.7153D+01         -3.7
0.1000D-01        0.2274D+01      0.7135D+01         -6.5
0.1778D-01        0.2259D+01      0.7077D+01        -11.5
0.3163D-01        0.2213D+01      0.6898D+01        -20.3
0.5624D-01        0.2081D+01      0.6365D+01        -35.3
0.1000D+00        0.1763D+01      0.4926D+01        -59.0
0.1779D+00        0.1217D+01      0.1709D+01        -91.7
0.3163D+00        0.6249D+00     -0.4083D+01       -130.6
0.5624D+00        0.2234D+00     -0.1302D+02       -170.1
0.1000D+01        0.5460D-01     -0.2526D+02       -198.8
0.1779D+01        0.1340D-01     -0.3746D+02       -205.2
0.3163D+01        0.3948D-02     -0.4807D+02       -221.7
0.5625D+01        0.8797D-03     -0.6111D+02       -247.2
0.1000D+02        0.1507D-03     -0.7644D+02       -266.6
0.1779D+02        0.2318D-04     -0.9270D+02       -275.1
0.3163D+02        0.3678D-05     -0.1087D+03       -275.7
0.5625D+02        0.6199D-06     -0.1242D+03       -274.0
0.1000D+03        0.1081D-06     -0.1393D+03       -272.4

                       ABOVE VALUES PLACED IN STORE 1

TRANSFER FUNCTION FROM Q TO R INCLUDING NEGATIVE FEEDBACK VIA PATH 3

 ORDER  OF  Y    6
-0.9869D+02  -0.1233D+03  -0.2812D+03  -0.2540D+03  -0.1497D+03  -0.3573D+02  -0.2545D+01

 ORDER  OF  X    3
-0.9455D+01  -0.8273D+01  -0.5000D+01  -0.2727D+00

 L.H. COLUMN OF ROUTH ARRAY

  0.2545D+01
  0.3573D+02
  0.1316D+03
  0.1938D+03
  0.1493D+03
 -0.2152D+02
  0.9869D+02

 UNSTABLE WITH   2 UNSTABLE ROOTS
```

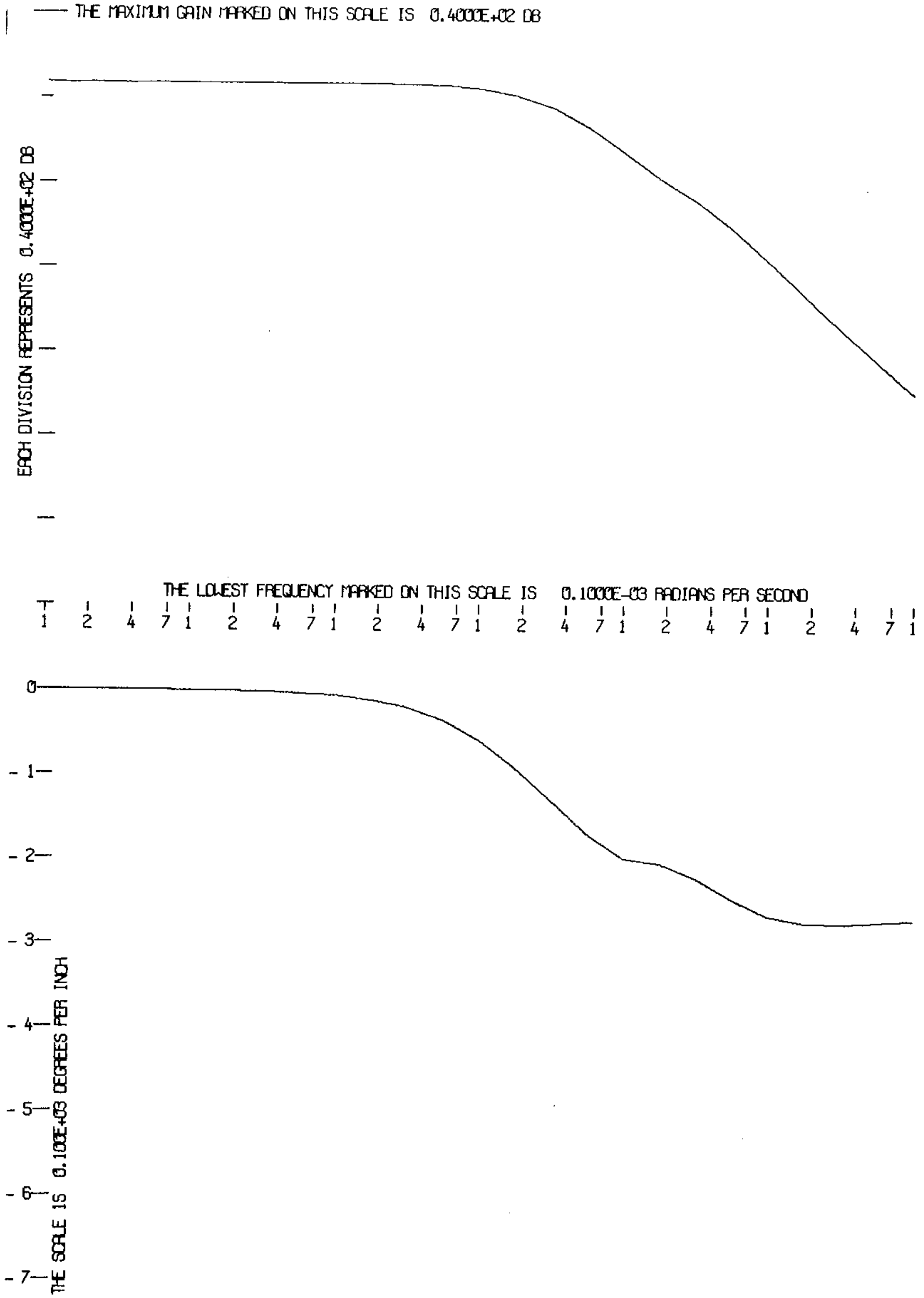
```
REDUCE THE GAIN IN FEEDBACK PATH 3 BY A FACTOR OF 10

  ORDER   OF   Y     6
 -0.1360D+03  -0.5890D+03  -0.1862D+04  -0.2515D+04  -0.1497D+04  -0.3573D+03  -0.2545D+02

  ORDER   OF   X     3
 -0.9455D+02  -0.6273D+02  -0.5000D+02  -0.2727D+01

          DENOMINATOR ROOTS ARE

          REAL PART                    IMAGINARY PART
          -0.1226811124812163D+01      -0.3726067228931942D+00
          -0.1226811124812163D+01       0.3726067228931942D+00
          -0.1544394795890856D+00      -0.3246619024821578D+00
          -0.1544394795890856D+00       0.3246619024821578D+00
          -0.8210858152986960D+01       0.0
          -0.3062354923924824D+01       0.0

          ORIGINAL AND RECONSTRUCTED DENOMINATOR COEFFICIENTS
          A( 1) =-0.1359999999999999D+03      D( 1) =-0.1359999999999997D+03
          A( 2) =-0.5889545454545452D+03      D( 2) =-0.5889545454614140D+03
          A( 3) =-0.1862090909090908D+04      D( 3) =-0.1862090909103713D+04
          A( 4) =-0.2515409090909090D+04      D( 4) =-0.2515409090913727D+04
          A( 5) =-0.1497181818181817D+04      D( 5) =-0.1497181818182205D+04
          A( 6) =-0.3572727272727271D+03      D( 6) =-0.3572727272727271D+03
          A( 7) =-0.2545454545454545D+02      D( 7) =-0.2545454545454545D+02

          NUMERATOR ROOTS ARE

          REAL PART                    IMAGINARY PART
          -0.8490869570085368D+00      -0.1167472008818004D+01
          -0.8490869570085368D+00       0.1167472008818004D+01
          -0.1663515941931626D+02       0.0

          ORIGINAL AND RECONSTRUCTED NUMERATOR COEFFICIENTS
          A( 1) =-0.9454545454545450D+02      D( 1) =-0.9454545454545421D+02
          A( 2) =-0.8272727272727266D+02      D( 2) =-0.8272727272727406D+02
          A( 3) =-0.4999999999999996D+02      D( 3) =-0.4999999999999996D+02
          A( 4) =-0.2727272727272725D+01      D( 4) =-0.2727272727272725D+01

          RECONSTRUCTED TRANSFER FUNCTION PLACED IN STORE 3


STEP RESPONSE OF LOW GAIN SYSTEM

THE TIME UNIT IS    0.1000D+01  SECONDS
   WITH AT LEAST     10 STEPS PER UNIT
THE RELATIVE ERROR IS    0.1000D-06

MAXIMUM =  0.8349D+00       AT TIME =  0.1105D+02

MINIMUM =  0.0              AT TIME =  0.0

 THE OUTPUT FOLLOWING A UNIT STEP INPUT IS

 TIME (UNITS)  0.0          0.525D+00  0.102D+01  0.155D+01  0.205D+01  0.255D+01  0.305D+01  0.355D+01  0.405D+01  0.455D+01
    OUTPUT     0.0          0.290D-02  0.156D-01  0.424D-01  0.776D-01  0.122D+00  0.175D+00  0.234D+00  0.298D+00  0.363D+00

 TIME (UNITS)  0.505D+01  0.555D+01  0.605D+01  0.655D+01  0.705D+01  0.755D+01  0.805D+01  0.855D+01  0.905D+01  0.955D+01
    OUTPUT     0.429D+00  0.493D+00  0.554D+00  0.611D+00  0.662D+00  0.706D+00  0.744D+00  0.775D+00  0.799D+00  0.817D+00

 TIME (UNITS)  0.100D+02  0.105D+02  0.110D+02  0.115D+02  0.120D+02  0.125D+02  0.131D+02  0.136D+02  0.141D+02  0.146D+02
```

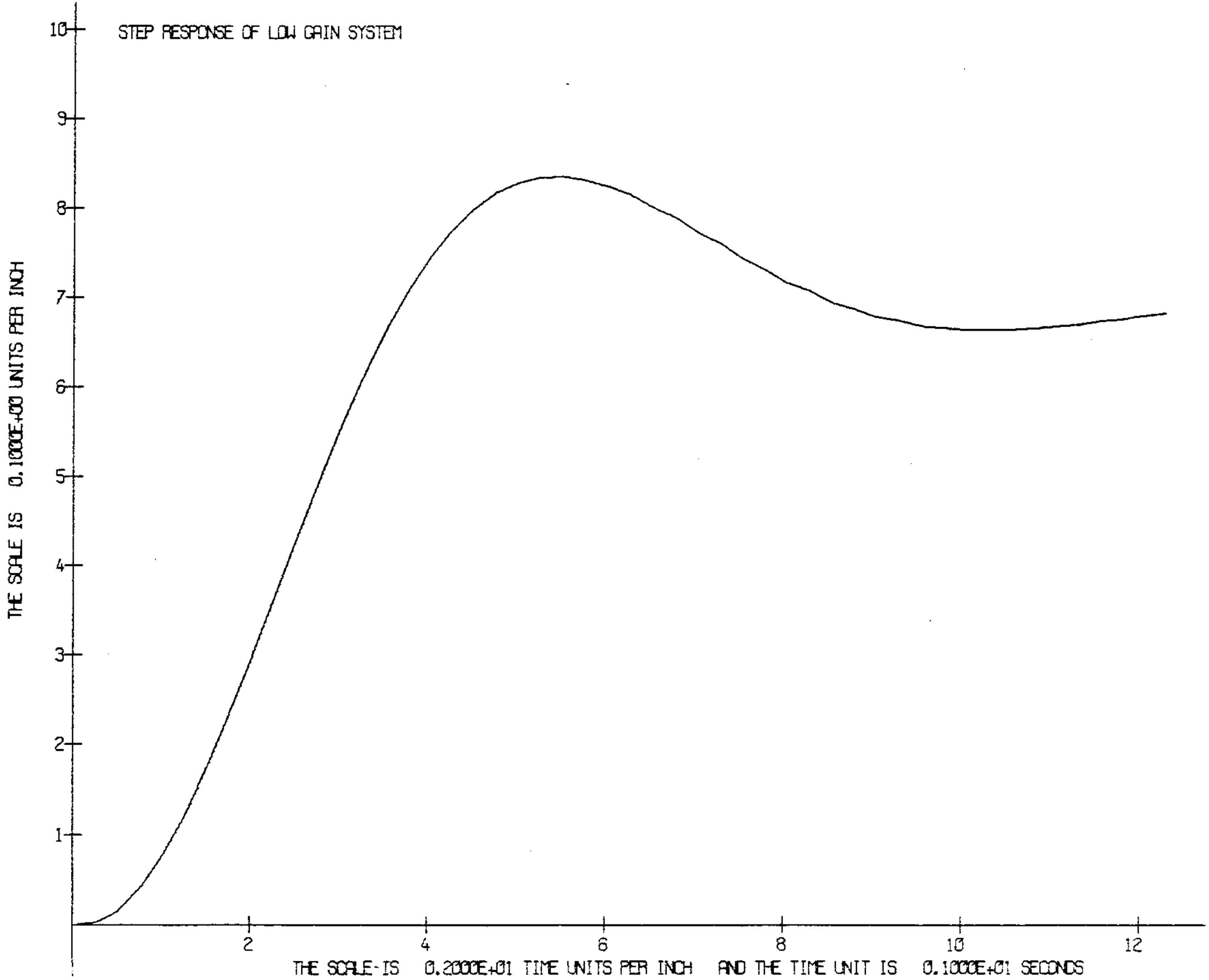
```
     OUTPUT        0.828D+00  0.834D+00  0.835D+00  0.831D+00  0.824D+00  0.815D+00  0.800D+00  0.789D+00  0.772D+00  0.760D+00

TIME (UNITS)       0.151D+02  0.156D+02  0.161D+02  0.166D+02  0.171D+02  0.176D+02  0.181D+02  0.186D+02  0.191D+02  0.196D+02
     OUTPUT        0.743D+00  0.731D+00  0.716D+00  0.707D+00  0.694D+00  0.687D+00  0.678D+00  0.674D+00  0.668D+00  0.666D+00

TIME (UNITS)       0.201D+02  0.206D+02  0.211D+02  0.216D+02  0.221D+02  0.226D+02  0.231D+02  0.236D+02  0.241D+02  0.246D+02
     OUTPUT        0.664D+00  0.664D+00  0.664D+00  0.666D+00  0.668D+00  0.670D+00  0.674D+00  0.676D+00  0.680D+00  0.683D+00

       END OF FILE ON UNIT  1 - EXECUTICN TERMINATED
```

```
IEF285I   CPGDAG.LIB                                      PASSED
IEF285I   VOL SER NCS= AAE006.
IEF285I   SYSOUT                                          SYSOUT
IEF285I   VOL SER NCS=        .
IEF285I   SYS71162.TC91048.RP001.LINCAN.R0000005          DELETED
IEF285I   VOL SER NCS= LINK  .
**EOS           9.13.16 00057 SECS CC=0000
//
IEF285I   CPGDAG.LIB                                      KEPT
IEF285I   VOL SER NCS= AAE006.
IEF285I   SYS71162.TC91048.RP001.LINCAN.R0000001          KEPT
IEF285I   VOL SER NCS= LINK  .
**EOJ LINCAN    9.13.18 0.03 HOURS
```

LINCAN 9115 C.P.GILBERT 70.309 9.52.43

TRANSFER FUNCTION FROM P TO R, AND FREQUENCY RESPONSE





LINCAN 9115 C.P.GILBERT 70.309 9.53.07

TABLE I

LINCAN OPERATIONS

Guide Number U	Operation Specified	Section in which Described	Table Detailing Input Data
0	Read input T.F.	3.2.2	II
1,2,3	Combine input T.Fs.	3.2.2	II
4,5,6,7	Combine input T.F. with last result	3.2.2	II
8	Punch T.F. on cards	3.2.2, 3.4	IV
9	Use negative calculation	3.2.2	II
10	Place T.F. in Store 1	3.2.3	II
11,12,13	Combine input T.F. and T.F. from Store 1	3.2.3	II
14,15,16,17	Combine T.F. from Store 1 with last result	3.2.3	II
18	Obtain output from Store 1	3.2.3	II
19	Apply amplitude scale	3.2.5	IV
20-28	As for 10-18 but Store 2 used	3.2.3	II
30-38	As for 10-18 but Store 3 used	3.2.3	II
40	Apply Routh criterion	3.6	IV
41	Factorise using slow method	3.4	IV
42,43	Cancel T.F. common factors	3.5	IV
44-47	Read differential equation data	3.1	III
48	Find a T.F. from the D.E. data	3.1	III
49	Calculate and plot frequency response	3.3	IV
51	Factorise using standard method	3.4	IV
52-53	Divide T.Fs.	3.2.4	II
54-56	Calculate and plot time response	3.7	IV
58, 59	Specify print-out precision	4.3	IV
60	Apply time scale	3.2.5	IV

NOTE: 1. T.F. stands for Transfer Function

2. Combination implies series, parallel or feedback connection

TABLE II

TRANSFER FUNCTION MANIPULATION – DATA REQUIREMENTS

U	INPUT DATA					SOURCE OF INITIAL POLYNOMIALS		RESULT		
		Polynomials						OUTPUT	COMMENTS	
	Title	B	A	D	C	A/B	C/D	X/Y		
0	*	*	*			INPUT		A/B		
1	*	*	*	*	*	INPUT	INPUT	3	Series	All new data
2	*	*	*	*	*	INPUT	INPUT	4	Parallel	
3	*	*	*	*	*	INPUT	INPUT	5	Feedback	
4	*			*	*	X'/Y'	INPUT	3	Series	Use preceding results and new data
5	*			*	*	X'/Y'	INPUT	4	Parallel	
6	*			*	*	X'/Y'	INPUT	5	Feedback	
7	*	*	*			INPUT	X'/Y'	5	Feedback	
9								X'/Y'	Inserts minus into next computation	
10								X'/Y'	X'/Y' placed in Store 1	
11	*			*	*	X_D/Y_D	INPUT	3	Series	Use data from Store 1 and new data
12	*			*	*	X_D/Y_D	INPUT	4	Parallel	
13	*			*	*	X_D/Y_D	INPUT	5	Feedback	
14	*					X_D/Y_D	X'/Y'	3	Series	Use preceding result and data from Store 1
15	*					X_D/Y_D	X'/Y'	4	Parallel	
16	*					X_D/Y_D	X'/Y'	5	Feedback	
17	*					X'/Y'	X_D/Y_D	5	Feedback	
18								X_D/Y_D	Output obtained from Store 1	
20 28									As for 10-18, but Store 2 used	
30 38									As for 10-18, but Store 3 used	
52	*					X_D/Y_D		6	X'/Y' divided by contents of Store 3	
53	*	*	*			INPUT		6	X'/Y' divided by input data	

NOTE: 1. Initial transfer functions are denoted A/B and C/D
2. The computed, or output, transfer function is denoted X/Y
3. X'/Y' represents the output of the immediately previous operation
4. X_D/Y_D represents a transfer function obtained from a Store
5. Only data marked * should be provided
6. The format of the data is described in Section 4.2
7. The numbers in the OUTPUT column refer to the equations of Section 3

TABLE III

DATA FOR DIFFERENTIAL EQUATION REDUCTION

U	INPUT DATA	COMMENTS
44	NROW NCOL MCO1 MCO2 .... SCO1 SCO2	Matrix size and all coefficients must be read in by columns, followed by s-vector: denominator of required Transfer Function placed in Store 3.
45	NROW NCOL NMCO NR1 NC1 MCO1 NR2 NC2 MCO2 .... NSCO NS1 SCO1 NS2 SCO2	Only non-zero coefficients need to be read in: denominator of required Transfer Function placed in Store 3.
46	NMCO NR1 NC1 MCO1 NR2 NC2 MCO2	Values of a specified number of coefficients of an existing matrix are to be changed to the given values: denominator of required Transfer Function placed in Store 3.
47	NSCO NS1 SCO1 NS2 SCO2	Values of a specified number of coefficients of an existing s-vector are to be changed: denominator of required Transfer Function placed in Store 3.
48	TITLE NCO NCI	NCO specifies the output variable column, and NCI specifies the input variable column counting from the equals sign. Complete Transfer Function printed out, left as X/Y, and retained in Store 3.

NOTE: NROW = Total number of rows in the matrix

NCOL = Total number of columns in the matrix

NMCO = Number of matrix coefficients to be read in

NSCO = Number of s-vector coefficients to be read in

NRn = Row number of a particular matrix coefficient

NCn = Column number of a particular matrix coefficient

NSn = Position number of a particular s-vector coefficient counting from the top left-hand corner

MCOn = A particular matrix coefficient

SCOn = A particular s-vector coefficient

TABLE IV

DATA FOR ANALYSIS AND OUTPUT OPERATIONS

U	INPUT DATA	COMMENTS
8	–	X'/Y' punched out on cards.
19	AMAX	X'/Y' is amplitude scaled so that the largest coefficient is equal to AMAX.
40	–	Routh Criterion applied to X'/Y', which is left unchanged.
41	–	X'/Y' factorised using slow method: should not be used unless U=51 has failed. Synthesized Transfer Function placed in Store 3.
42	TITLE, N	Common factors of X'/Y' are sought and cancelled from the numerator and denominator. Each of the N attempts starts with X'/Y': the final residual Transfer Function is left in Store 3.
43	TITLE, N	As for U=42, but each of the N attempts starts with the preceding residual Transfer Function.
49	LFREQ, RATIO, HFREQ	Frequency response of the next Transfer Function to be found is printed and plotted. The response starts at LFREQ and finishes at HFREQ with frequency ratios which do not exceed RATIO.
51	–	X'/Y' factorised using standard method: synthesized Transfer Function left in Store 3.
54 55 56	TITLE, TUNIT, FINTIM STEPS/UNIT, ERROR, PRINTS/UNIT	Impulse Input Step Input Ramp Input Time response of X'/Y' printed and plotted. See Section 3.7 for description of input data.
58	–	High precision print-out for all output until reset.
59	–	Normal precision print-out.
60	τ	X'/Y' is time scaled: the coefficients of s^n are effectively multiplied by τ^n.