NASA CONTRACTOR REPORT

NASA CR-475

NASA CR-475

A USER'S MANUAL FOR THE AUTOMATIC SYNTHESIS PROGRAM

(PROGRAM C)

by R. E. Kalman and T. S. Englar

Prepared by
MARTIN-MARIETTA CORPORATION
Baltimore, Md.
for Ames Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JUNE 1966

NASA CR-475

A USER'S MANUAL FOR THE AUTOMATIC SYNTHESIS PROGRAM

(PROGRAM C)

By R. E. Kalman and T. S. Englar

Prepared under Contract No. NAS 2-1107 by
MARTIN-MARIETTA CORPORATION
Baltimore, Md.

for Ames Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – Price $5.50

PREFACE

ASP, The Automatic Synthesis Program, is a digital-computer program. It is written in FAP and compatible with most IBM 7090-7094 installations.

ASP is an executive program enabling the user to sequence arbitrarily some thirty subroutines. ASP is intended for a person who has some background in modern system theory, especially linear algebra. Such a user can obtain, with the help of ASP, easily and efficiently the numerical solution of most (perhaps all) problems in system theory which involve linear mathematics. Time-varying linear systems are included; in fact they will constitute the most important practical applications. Among others, the following subroutines are included: input-output calculations, the operations of matrix algebra, computation of the exponential of a matrix, the solution of both discrete and continuous-time riccati equations.

The subroutines in ASP are sufficient to solve the most general problem of extremization of quadratic functionals of the state of a linear dynamical system. This includes, in particular, the synthesis of the Kalman-filter gains and of the optimal feedback gains for minimization of a quadratic performance index, possibly with hard terminal constraints. Use of the program is by no means restricted to these two problems, however, as is evident from the table of contents.

In short, ASP is a very flexible program which can be conveniently and quickly programmed to give the solution to a variety of problems. Input consists of a description of the equations to be solved as well as numbers specifying the system matrices. The programming has been made as "macroscopic" as possible; many special features have been built into the subroutines for the convenience of the user and to lessen numerical difficulties. We hope that the availability of a convenient and reliable program such as this will contribute to greater awareness of the powers of modern system theory, particularly in the areas of control, statistical filtering, and optimization.

After this very terse description, let us outline what ASP is not. It is not an unusually accurate program nor is it a very fast program. Its limitations arise mainly from the fact that integration is done by the matrix exponential (hence all n^2 couplings are considered whether or not they exist) and because the matrix exponential computation involves scaling problems that do not arise in ordinary numerical integration schemes. We try to compensate for this by automatic scaling. A major cause of time wastage is the compilation of the instruction data every time the deck goes on the machine. That is, the ASP executive routine is not recompiled, but the list of ASP instructions to the executive routine are translated to a system of calling sequences for the ASP subroutines. A binary program could be punched easily from the machine contents, but we have never done enough repetitive work to justify it.

Given these limitations, what uses do we see for ASP?

One of the chief benefits will be educational. This manual contains text-book examples and some more involved practical problems. These illustrate not only the use of ASP but also the steps required to solve these various problems. They provide concrete cases, with real numbers, for the novice in optimal system theory.

ASP is expected to be very useful as an analytic design tool where extreme precision is not required. In addition to the problems discussed in this report, the program has been used satisfactorily on several reentry problems, lunar landing problems, airplane control, and control of large flexible boosters.

Some remarks may be useful here on optimal control as a practical tool. Many engineers have questioned the relevancy of linear optimal control to practical problems because the performance index has no simple relation to the criteria which must be imposed (bounded state variables or bounded control variables, for instance). There are two important considerations here:

(i) It has been our experience that it is usually easy to alter the performance index by experimentation to achieve the desired shape of the trajectory. That is, after a few trials we can usually find an optimal (in the modern sense) control which will appear satisfactory also to the engineer who works with older style criteria.

(ii) The computation of exact nonlinear control laws is at present prohibitively expensive. It is far more sensible to iterate a general-purpose linear optimization program by trial and error than to write a separate, exact, nonlinear optimization program for each case. This will remain true even when and if nonlinear optimization methods become more generally available. (Note that today (1965) only a few special nonlinear optimization procedures have been studied from the point of view of numerical analysis.)

As for the availability of the ASP program, it can be obtained on request from the Theoretical Guidance and Control Branch, Ames Research Center, NASA, Moffett Field, California. Further details regarding tapes, listings, and check cases can be arranged.

We are indebted to many persons for their efforts, but we particularly wish to thank Mr. Elwood C. Stewart of Ames Research Center for his interest and assistance and Miss Elsie Cerutti of RIAS for her excellent programming work.

This research was supported by National Aeronautics and Space Administration Contract NAS2-1107 administered by the Ames Research Center, Moffett Field, California.

TABLE OF CONTENTS

INTRODUCTION

This report may be divided into three parts: Chapter I, Chapters II-VIII, and Chapters IX-XVIII.

Chapter I describes the basic mechanics of ASP, its input format, vocabulary, error returns, and idiosyncrasies. For the person who understands his problem, knows what mathematical steps are required in its solution, only Chapter I is needed. In the dictionary, the subroutines are described by specifying the output and how each of the arguments is used. Then referring to the coding sheets at the end of Chapter I, we can determine in which columns the argument names appear. For instance, in the dictionary description of EAT, a matrix PH is output. When using the EAT routine we may call this output by any other two-character symbol, but that symbol must appear in the columns where PH appears in the coding sheets.

Chapters II through VIII describe in detail the more important subroutines available. Generally some theoretical background will be given deriving from basic principles the application of the particular routine to linear systems theory. In addition, the method of computation will be explained with remarks about numerical problems and several check problems given. These chapters will be useful not so much for helping in the actual use of the program but as somewhat more than an introduction to the theory, particularly as it is used in computation. For instance, Chapter VIII provides an ad hoc derivation of the optimal control law for a discretized system with quadratic performance index.

Chapters IX through XVIII are examples in some depth of computation and analysis. The selection of problems solved is far from comprehensive, yet the basic problems of using the subroutines are well covered. The specific ASP programs used appear in the figures for each chapter and when read, should provide a working knowledge of the ASP language. It is our hope that these problem chapters will be an introduction to ASP and a tutorial in linear systems theory from a computational point of view.

CHAPTER I

ASP - THE AUTOMATIC SYNTHESIS PROGRAM

1. History. The nature of our work is such that a basic set of matrix subroutines which can be sequenced in a variety of ways would solve many problems. In the original formulation of a program, several approaches were considered. Since there were available at the time many of the necessary subroutines, a programmer could write new connective code very easily for each separate problem in the 709-7094 SOS system. This method has the advantage of immediate availability with very little program verification since the component programs were already checked out. With increased outside interest in the program, it was discovered that many of the IBM computer installations (e.g., Wright Field) will not accept SOS programs. Consideration was also being given to writing a Fortran program with input which would determine the sub-programs to be utilized. A single program with code accepting decision points to determine flow would be easy to write but the data becomes extremely unwieldy when the number of permutations of routines is large; an even greater difficulty lay in anticipating every type of problem which would be solved. We decided to adopt a scheme which may be regarded roughly as a matrix formula translator enabling us to write macro-instructions directly in terms of the needed matrix operations. Since so much of the programming had already been done in the SOS system, we decided to keep the SOS input-output unit and assemble it as the ASP I-Ø package. Again we ran into difficulty with another computer installation which did not have tape units attached directly to the 7094. Therefore ASP has evolved into a FAP coded-formula translator with Fortran II input-output, which may be loaded with the main ASP deck or called off the system tape.

2. I-Ø Requirement. Before going into more detail about the main program, several features concerning input-output should be pointed out. Contact is made with the I-Ø routines through the following entry points: (TSB), (STB), (TSH), (STH), (SPH), (BST), (RSW), (RIR), (WIR), (FIL), (RTN) and EXIT. It appears possible that with a knowledge of the function of each routine and with more detail on calling sequences and format, one could run this deck at any IBM installation with a 32 K IBM 7090-7094 and any tape units required by the I-Ø. The format specifications are standard Fortran II as described in Form C28-6066-5. Use is made of the A, H, E, F, and I mode, multiline blocks of print of 120 characters specified by a slash, and column skip specified by an X. The logical tape number for input is 5, for output 6. The only nonstandard feature is that the format specification is stored in increasing storage locations. The code which indicates this type of format statement is a 1 in the decrement of the calling sequence.

3. Description of ASP. ASP is composed of two parts, an executive routine, whose function is to read, list, and store alphameric data and a group of independent subroutines which are classified by use as mathematical - those having to do with numerical analysis; data handling - those involved in making storage available, moving information in core or between core and tape; output - those which write alphameric or decimal information; input - those which bring in decimal information; and logical - those which provide a means of altering the progress of the program.

The executive program described diagramatically in Figure 4 expects as a first card a date card of which there is one and only one in each data deck. Then it begins reading until it finds a card with the operation BEGIN in Columns 7-11, which signals the executive program that alphameric information follows. This alphameric information is separated into 12, 6 character words, the first of which is a label or heading name, the second word specifies the operation and words 3 through 12 contain the operands. Operands may be referred to by anytwo alphameric characters; followed by

either a comma or blank character; their position following the name of the operation specifies the role each plays in the operation. Resultant matrix storage locations are created in the order in which they appear from left to right. For instance when

```
PSEUO    A,      AI, RK,
ADD      A,  B,  C,
```

is first executed, if conformability requirements are satisfied and AI, RK, and C have not been previously defined, they will appear in core in that order. If one of the matrices AI, RK and C have been previously defined and if the size of the new matrix is the same as the size of the core matrix then the resultants will be stored in the defined region.

The alphameric information has a fixed field format. If a matrix identifier is -A (- represents a blank) then any reference to A must be written -A. A- will be defined as a different matrix. The same rule describes the label. The format then can be described in a very general way as follows:

Columns	
1-6	labels (may be blank)
7-11	operation
13-14	operand 1 or blank
15	blank or comma
16-17	operand 2 or blank
18	blank or comma
.	
.	
.	etc.

Specfically every instruction has an operation in Column 7-11. Whether operand 1 is in Column 13-14 depends on the instruction. See Figs. 1-3 for the specific location of the matrix identifiers.

In addition to the fixed format restriction there is a limit of 120 for the number of matrix identifiers. If this limit is exceeded error statement number 5 is printed.

Certainly every operation does not require 10 words for the operands. Therefore a test is made to eliminate anything beyond the first completely blank word. This implies certain restrictions on the use of double-blank as an identifier. Such a use is not recommended. This alphameric data from the second word to the first blank word is stored downward unchanged in a reserved block behind the program. Two words precede the stored data, a machine instruction which acts as a transfer to the 30 subprograms and a control word with the number of operands for this instruction. This is done for every alphameric data card except the BEGIN. Observe then, that the data is stored to look like a series of call statements with their respective arguments, thereby acting as a stored program. One is able to change the direct flow through the stored program by use of the IF instruction which in turn uses the label that was in Columns 1-6 of an alphameric data card. If word 1 is not blank it is stored in a vector (dimension 40) of labels along with the address of the core location which contains the transfer for that operation.

The END instruction signals the end of the instruction data; and a transfer is made to the first stored data point. Since the input tape is now positioned after the END record any LOAD given in the stored program will cause the input tape to be read for decimal information from that point. To summarize, the data deck is in the following order: 1) Date, 2) BEGIN, 3) any of the 30 numeric, input or output statements, 4) END, 5) decimal data for the LOAD instructions or instruction in the order that the LOAD will be executed. For information about the decimal data see LOAD in the dictionary which follows.

Every subprogram has two entry points. The first entry point uses the symbolic material, which is stored following the transfer instruction, to determine the location of the first element of the operands. A number is stored, in place of the alphameric symbol, which defines the position of

that matrix in the ordered matrix storage area. This part of the subprogram also defines a matrix region for the resultant and examines the operations for conformability. If the region has already been defined, the size of the defined region is compared with the size of the resultant matrix. If the sizes are not equal error statement number 8 is printed and execution is deleted. An error statement is also written if the matrices are not conformable. Finally "the transfer to entry point 1" instruction is replaced by a "transfer to entry point 2". The second part of the subprogram performs the numerical calculation. It is this part of the subprogram which is repeated if the operation is in the range of an executed IF instruction. Care must be exercised in storing results in previously defined areas. The row and column dimensions of the one may not be the same as the row and column dimensions of the other; thus allowing undetected nonconformable operations in any of the repeated operations since the confomability check is made in the first part of the subprogram.

At present one can communicate with ASP by the following vocabulary

Mathematical

1. E A T
2. E T P H I
3. T R N S I
4. R I C A T
5. P S E U O
6. S A M P L
7. I N V R S
8. D E C O M
9. A D D
10. S U B T
11. M U L T
12. T R A N P
13. N O R M
14. T R A C E
15. P I Z E R
16. J U X T C
17. J U X T R

Data Handling

18. S A V E
19. E Q U A T
20. B L O T
21. R E W
22. W E F
23. B S R

Output

24. P R I N T
25. R I N T
26. W R I T E
27. P U N C H

Input

28. L O A D
29. B R I N G

Logical

30. I F
31. B E G I N
32. E N D

D I C T I O N A R Y

1. E A T. This program computes $PH = e^{T \cdot F}$ and $B = \int_0^T e^{tF} dt$; PH and B will be printed if P R I N T is punched in columns 25 to 29. F can be 30×30.

2. E T P H I. This program computes $PH = e^{T \cdot F}$; PH will be printed if P R I N T is punched in columns 25 to 29, F can be 30×30.

ETPHI has the option that if a matrix symbol is contained in column 22 to 24 the value of t which will be used will not be the input t but a value $\frac{t}{2^k} \leq \frac{10}{\|F\|}$. The value $\frac{t}{2^k}$ will be stored in core, identified by the symbol in 22, 23.

3. T R N S I. This program computes $y(t)$ and $u(t)$ according to the difference equations

$$x(t + \tau_1) = Px(t) + Gu(t)$$

$$u(t) = JR - Kx(t_o + i\tau_2),\ i\tau_2 \leq t - t_o < (i + 1)\tau_2$$

where

$$y(t) = Hx(t)$$

where $x(t_o) = X$ and i is a positive integer.

Rather than describe the output in detail we have included a sample T R N S I output as Fig. 19.

Print of t, the first two components of R, Y, and the first three components of u is done at intervals of τ_1. The control computation is done at intervals of τ_2 with τ_2 some positive integral multiple of τ_1 and these steps are marked by an asterisk. Because there is room for only 7 components of y, the H matrix should have less than 8 rows.

The problem terminates when the final time t_f is reached.

The matrix T has elements τ_2, τ_1, t_o, and t_f in that order.

4. R I C A T. This program computes $P(t)$ and $K(t)$ by the difference equation

$$P(t + \tau) = [\theta_{21} + \theta_{22}P(t)][\theta_{11} + \theta_{12}P(t)]^{-1}$$

$$K(t) = CP(t)$$

with $P(0) = Z$ and

$$\begin{bmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix} = PH.$$

In addition to the given equation there is an option in which if there is a matrix name X in Columns 28-30 the computation for P(1) and only P(1) will be

$$P(1) = [\theta_{21}X + \theta_{22}P(0)][\theta_{11}X + \theta_{12}P(0)]^{-1}$$

If this option is used, $[\theta_{11}X + \theta_{12}P(0)][\theta_{11}X + \theta_{12}P(0)]^{-1}$ is automatically printed.

This program also computes

$$AL = \frac{\Sigma|P_{ii}(t+\tau) - P_{ii}(t)|}{\Sigma|P_{ii}(t+\tau)|}$$

Computation continues until $t + \tau$ is equal to or greater than T_F, an input number, or until AL is less than ϵ, an input number. Printing is controlled as follows. p_1 controls the intermediate printing of K, p_2 controls the intermediate printing of P, p_3 controls final K and P print. If

p_1 or p_2 is

- 0 there is no intermediate print
- M there is a print every Mth iteration

if

p_3 is

- 0 there is no final K and P print
- 1 there is a final K and P print.

PC is a matrix with fixed point elements p_1, p_2, p_3, and β (in that order). D is a matrix with elements ϵ, τ and T_F.

Where β is 0 if the Riccati equation is known to be linear and nonzero otherwise.

5. P S E U O. This program computes the pseudo-inverse of P, storing it in PI and the rank r of P, storing it in RK. The final pivotal element and

$$\rho = \min\left(\frac{\|(B^{\#}B - I)B\|}{\|B\|} + \frac{\|(B^{\#}B - I)B^{\#}\|}{\|B^{\#}\|}\right)p_2^{\,r}$$

(see Chapter V and dictionary entry 15) are also available as the second and third elements of the 3 element matrix RK. Pseuo will use the pizer in the machine and the ρ computation to determine the rank of P, if a 1 is punched in Column 18; otherwise the pseudo inverse of P will be computed with a pizer as read in by the PIZER instruction and no iteration as determined by ρ. Computation therefore will cease when rank is maximal or any diagonal element is less than the product of pizer and the maximum diagonal element. If the matrix P is nonnegative definite symmetric, a + sign should be punched in Column 16; this will save time and improve accuracy, since it will drop the symmetrizing steps. PI will be printed if PRINT is punched in Columns 25-29 and ρ as a function of rank, will follow PI if a 1 appeared in Column 18.

6. S A M P L. This program computes $P(t_o + N\tau)$ and $K(t_o + N\tau)$ by the difference equations

$$P(t + \tau) = F[P(t) - P(t)G'(GP(t)G' + R)^{\#}GP(T)]F' + Q$$

$$K(t) = FP(t)G'[GP(t)G' + R]^{\#}$$

with $P(t_o) = PO$.

The pseudo-inverse in these equations is computed using the value of pizer which has been stored by the PIZER instruction and no ρ iteration as described above.

Computation continues until the number of steps exceeds N, an input number, or until AL is less than ϵ, an input number. Printing is controlled as follows; p_1 controls the intermediate printing of K, p_2 controls the intermediate printing of P, p_3 controls the final K and P print. If

p_1 or p_2 is
- 0 there is no intermediate print
- M there is a print every Mth iteration

if p_3 is
- 0 there is no final print
- 1 there is a final print.

D is a matrix with first element ϵ and second element τ.

PC is a matrix with fixed point elements p_1, p_2, p_3, and N (in that order).

7. I N V R S. The program finds $Y = P^{-1}$, provided P is not a scalar. To invert scalars, P S E U O must be used. (See p. 21, Error 1 when det P = 0.)

8. D E C O M. For any nonnegative definite symmetric, n-dimensional matrix B of rank (in the sense of P S E U O) r, this program returns matrices T, E_r and E such that T is nonsingular, E is an n-dimensional matrix as defined by $E = TBT'$ and E_r is some r columns of the n-dimensional identity matrix where $E_rE_r' = TBT'$. In addition the matrix $S = T^{-1}$, the permutation matrix P such that

$P\,E\,E'\,P' = \begin{bmatrix} I_r & 0 \\ 0 & 0_{n-r} \end{bmatrix}$, and a 3×1 matrix RK with r, final pivotal element, and ρ (in that order) are also provided. If a 1 is punched in column 18, the machine pizer will be used and the rank of B will be determined by the computation $\rho = \min \|S\,E\,E'\,S' - B\|p_2^r$. If column 18 is blank, computation will cease when rank is maximal or any diagonal element is less than the product of pizer and the maximum diagonal element.

Some care must be exercised in using DECOM in a loop since E_r may change its dimensions. This of itself is not a problem since E_r is stored in a sufficiently large area, $n \times n$, but if E_r is transposed and the dimension of E_r' increases as iterations continue, then E_r' will destroy the storage following itself. This usually is not a serious restriction on programming.

9. A D D. This program computes $C = A + B$. These matrices can be 30×30.

10. S U B T. This program computes $C = A - B$. These matrices can be 30×30.

11. M U L T. This program computes $C = AB$ if 1) A and B are conformable or 2) A is a scalar. The output of this program can have 900 elements.

12. T R A N P. This program computes the transpose $B = A'$.

13. N O R M. This program computes $NA = \{\max_i \Sigma |a_{ij}|, \max_j \Sigma |a_{ij}|\}$. The input to this routine can be a 30×30 matrix.

14. T R A C E. This program computes $TB = \text{trace } B = \sum_{i=1}^{n} b_{ii}$.

15. P I Z E R will store two constants P_1 and P_2 which are used in the P S E U O, D E C O M, and S A M P L instructions. The program values of the two constants are 10^{-2} and 1 unless the P I Z E R instruction stores other values. For a complete description of their use see Chapters V and VII.

16. J U X T C. This program takes the m by n matrix A, the m by p matrix B, and forms the m by (n+p) matrix

$$C = [A \quad B].$$

JUXTC can not be used in a loop if the output dimensions are changing.

17. J U X T R. This program takes the $m \times n$ matrix A, the $p \times n$ matrix B, and forms the (m+p) by n matrix

$$C = \begin{bmatrix} A \\ \\ B \end{bmatrix}$$

JUXTR can not be used in a loop if the output dimensions are changing.

18. S A V E. This routine will write the given matrices on any of 9 units or channels as defined by the IØU tables of the installation monitor system. In Column 18 is punched a number 1 through 9 which is used as an entry to a table common to all the tape instruction. This table will contain a logical tape number which in turn is used by the Fortran IØU to select a tape channel and unit. The logical tape number as stored in the present version of ASP corresponds to the B channel units 1 through 9. The table is as follows:

Column 18	Logical number	Actual channel in FORTRAN II IØ
1	8	B-1
2	2	B-2
3	3	B-3
4	7	B-4
5	10	B-5
6	12	B-6
7	14	B-7
8	16	B-8
9	18	B-9

One SAVE instruction can be used to store as many as 9 matrices, giving each a tape name which can be different from the core name. In the sample, matrices A, B, and C are being saved on tape 5 and are being called D, E, and F respectively. Operation is as follows: The unit reads the tape until a record with the code word END is encountered, backspaces over that record, writes the matrices, writes a record with the code END and then backspaces over this last record. Some initialization therefore is necessary before the first occurrence of the SAVE instruction. This initilization can be REW, WEF, REW.

19. E Q U A T. This instruction replaces completely except for identification the matrix in the second half of the word with the matrix in the first half, if the sizes are equal. In the example F1 will be replaced by F2, F3, by F4. If F1 is undefined at the time of the instruction, F2 will be stored in a second area and given the identification F1. F2 under any situation will still be defined in storage. This instruction will handle as many as 10 pairs of matrices.

20. B L O T. This instruction erases the matrix F and every matrix stored after F, thus making the storage area available for reuse. This instruction cannot be used in a loop and will result in error statement 12 if an attempt is made to execute a given B L O T instruction a second time.

21. R E W. This instruction will rewind (to tape beginning) any one of nine tapes as designated by the table shown in the S A V E instruction (in the example, tape 5). The tape number is punched in Column 18.

22. W E F. This instruction will write a record with the code word E N D on any one of nine tapes as designated by the table shown in the S A V E instruction (in the example, tape 5). The tape number is punched in Column 18.

23. B S R. This instruction will backspace 1 to 9 records (in the example 3) on any one of 9 tapes as designated by the table shown in the S A V E instruction (in the example tape 5). The number of records is punched in Column 15, the tape number in Column 18.

24. P R I N T. This instruction will print up to ten matrices in one execution, printing each matrix, one row per line in a five-digit format, with its row size, column size, exponent, and a three character title. In the example, for instance, the matrix called CF in core will be printed with title C L F. If several matrices can be contained completely on a page, P R I N T will not eject to a new page for each matrix.

In addition to the above, there is an option in which every matrix title following the first on the list will be printed in functional notation using the first element of the first matrix on the list as the argument. To use this option put the argument name in the first word as usual (Columns 13, 14) but leave the three character title region blank (Columns 16, 17, 18).

25. R I N T. This instruction will print up to ten matrices in one execution. Printing is done row-wise, six elements per line to the end of the row, in an eight decimal place floating point format if the column size is greater than 6. Otherwise each row beings on a separate line. Several matrices are printed per page if they can be completely contained on a page. As in P R I N T, the row size, column size, and a three character title are printed. Also as in P R I N T, the instruction can print the title in functional notation.

26. W R I T E. This instruction enables the person using the program to print comments during the course of the program. These comments can use Columns 14-72 and might tell what the program is doing at various points. They are not necessary to a knowledge of the output, however, because the entire input data deck is printed by ASP. A page eject may be accomplished by a W R I T E instruction with a 1 in Column 13, a line eject by 0 in Column 13. A non-blank character must appear before Column 18; otherwise that card is ignored.

27. P U N C H. Just as P R I N T and R I N T, the punch instruction will handle a maximum of ten matrices in one execution, punching cards in a form which is compatable with the L O A D routine. For each matrix will be punched a row and column card with the identifying matrix name in Columns 4-6. There is also an option in which matrices 2 through 10 can be identified as a function of matrix 1. This option in addition to the identification in Columns 4-6 will punch the first element of matrix 1 in Column 24 to 38 of the row and column card of matrices 2 through 10. This number is not loaded onto the computer on a L O A D instruction. It is merely an additional identification for the programmer. To use this option leave Columns 16-18, of matrix identification 1 blank.

28. L O A D. This instruction may read into core from the input tape up to 20 matrices at one execution. If more than 20 matrices must be entered, L O A D instructions may be given at any time in the flow of a problem.

The decimal data for the LOAD is written after the END which defines that particular problem that uses the decimal data. The first card for every input matrix is a card with the row and column size punched in Columns 11, 12 and 17, 18 respectively. As in all FORTRAN I-Ø routines the fixed point numbers (no decimal point) must be right justified; that is if the row-column sizes are single digits they much be punched in Columns 12 and 18 respectively. Following the row-column card are the elements punched 5 per card with a field width of 15 between 1 and 75.

This instruction will also load into a previously defined area if the size of the matrix to be loaded is the same as the size of the previously defined area. Otherwise for every different identifier in the LOAD instruction, there will be read a set of decimal data, stored in the following order: matrix identification, row size, column size, $a_{1,1} a_{1,2}, \ldots, a_{m,n}$.

This instruction will handle matrices of size 30×30.

29. B R I N G. This routine will read up to 9 matrices from any one of 9 tapes as defined by the tables in the SAVE instruction. The tape number is punched in Column 18. In the example the matrices which are identified on tape 5 as F, G, and H are being read into core as F, G, and PH.

Operation is as follows. The tape unit reads forward from its location at the time of the instruction until it finds the first required matrix. It then reads in that matrix and searches forward (without rewinding) for the next matrix. If a matrix is not found an exit is made from the machine and error statement number 7 is printed; the program does not continue. Thus not all matrices on a tape need be brought, but those which are must appear in the same relative order as they are stored on tape.

This means that to recover the first matrix saved on a tape, an REW instruction must precede the BRING instruction. The operation was programmed in this manner so that a sequence of matrices all having the same identification symbol could be successively brought into core.

30. I F. This is the only instruction able to alter the course of the program. In the example, if the first element of I4 is greater than or equal to the first element of F, a transfer is made to an instruction which is labeled HEAD 1 in Column 1 to 6. If the condition does not exist, a transfer is made to HEAD 2 or if the area comprised of Columns 25-30 is blank the next executed instruction will be the one following the IF instruction.

An unconditional transfer can be made by comparing the same matrix.

The headings in Columns 1-6 may be any size character symbol and may be punched in Columns 1-6 of any data card except decimal data and the BEGIN control card. In case the heading required by an IF statement does not exist, then a problem exit will be made. Error statement number 4 will be printed.

Another restriction to the IF statement is the number of headings for each problem. If there are more than 40 headings, error statement number 13 will be printed.

A sample heading appears on the IF statement itself.

31. B E G I N. This instruction marks the beginning of a problem in the data deck. It precedes the first operation data card of every problem. BEGIN instructs the translator that Hollerith information follows. If the BEGIN card should be omitted the data will be read without processing until a BEGIN is found or to the end of the data. As an example of the use of this instruction see Fig. 5.

32. E N D. This instruction must directly follow the last operation card of any problem. It instructs the executive program to return to the first operation command and begin executing the program data. If the END card should be omitted the executive program will attempt to read the decimal data as Hollerith information and return with error statement 2.

Following is a list of program limitations and error statements internal to ASP. If an ASP error should be encountered, after writing an error statement transfer is made to the executive program which will begin by reading again without processing until the operation BEGIN is found. In addition to the ASP error statements the FORTRAN I-Ø will indicate such errors as illegal decimal data on the input tape or redundancy on either input tape or the tapes being used by ASP itself. If Fortran errors are encountered the program will exit to the monitor.

Of the following limitations number 1 is the only one which the user of ASP cannot remedy in general. The others can be controlled by taking advantage of the replacement capability or the BLOT instruction or even storing matrices on tape in one problem (as defined by BEGIN and END)

and then developing a new problem following the first (with a second BEGIN and END) which calls these matrices from tape.

1) All matrices, except as specifically exempted in the dictionary, must have their larger dimension less than 16. There is no general error return for using too large a matrix.

2) There is a limit of 120 on the number of matrices of any size which can be carried in core. If the number is exceeded, error statement number 5 is printed.

3) The program reserves a block of 11,000 words of core storage for the matrices and the operation instructions. If the problem is too large, error statement number 6 will be written.

ERROR STATEMENTS

1) ERROR RETURN FOR INVERSE IN (X X X X X X X X X X X),

In the operations which require inverses, RICAT, DECOM, INVRS, if the matrix is singular the above will be printed with the symbol in parentheses identifying the operation ahd the first six characters following the operation.

2) ERROR IN OPERATION (X X X X X X X X X X X).

This statement will be written if the executive program does not recognize the operation as one of the thirty (excluding BEGIN and END).

3) ERROR IN IDENTIFICATION (X X X X X X X X X X X) will be printed out, with the symbol in parentheses identifying the operation (and the first six characters following the operation) in which a non-existent matrix has been requested.

4) ERROR IN IDENTIFICATION (HEAD 1).

This will be printed if the non-existent heading (HEAD 1) is requested in an IF instruction.

5) NUMBER OF MATRIX SYMBOLS EXCEEDS 120 AT (X X X X X X X X X X X).

This statement is printed if an attempt is made to define more than 120 matrices for any one problem.

6) OVERLAP IN MATRIX DATA AND MATRIX OPERATION DATA.

This will be printed if the block of 11,000 words for matrix data and operations is exceeded.

7) IDENTIFICATION ERROR ON TAPE B XX. (MATRIX (XX).

This will be printed if the matrix requested, (named in the parentheses) is not found on tape during the execution of a BRING instruction. It is the tape analog of the core error 3.

8) DIMENSION ERROR IN (X X X X X X X X X X X X) will be printed, with the symbol in parentheses identifying the operation (and the first six characters following the operation) in which the matrices are nonconformable for some reason, or if the operation is storing a result in a previously defined matrix region which does not agree in size with the result; for instance in TRACE or INVRS if the argument is not a square matrix or in JUXTR if the number of columns in the two matrices are not equal.

9) ELEMENT TOO LARGE FOR PRINT FORMAT IN TRANSIENT,

The print format in transient has space for only 4 characters to the left of the decimal point. Therefore all numbers in TRNSI should be greater than -1000 and less than +1000.

10) SPILL IN (X X X X X X X X X X X X). Index register 1, 2, 4, address of spill location and spill code follow.

This statement alone will indicate an underflow and the computation will be continued. The above statement along with the following will indicate overflow and the program will exit.

11) OVERFLOW ERROR IN -- (X X X X X X X X X X X X).

12) BLOT IS NOT AN ACCEPTABLE OPERATION IN AN IF LOOP.

For a complete sample output see Fig. 5.

13) EXCEEDED RESERVED BLOCK FOR HEADING VECTOR WITH X X X X X X .

This statement is printed if there are more than 40 operations which have headings in Column 1-6. X X X X X X is the forty-first heading.

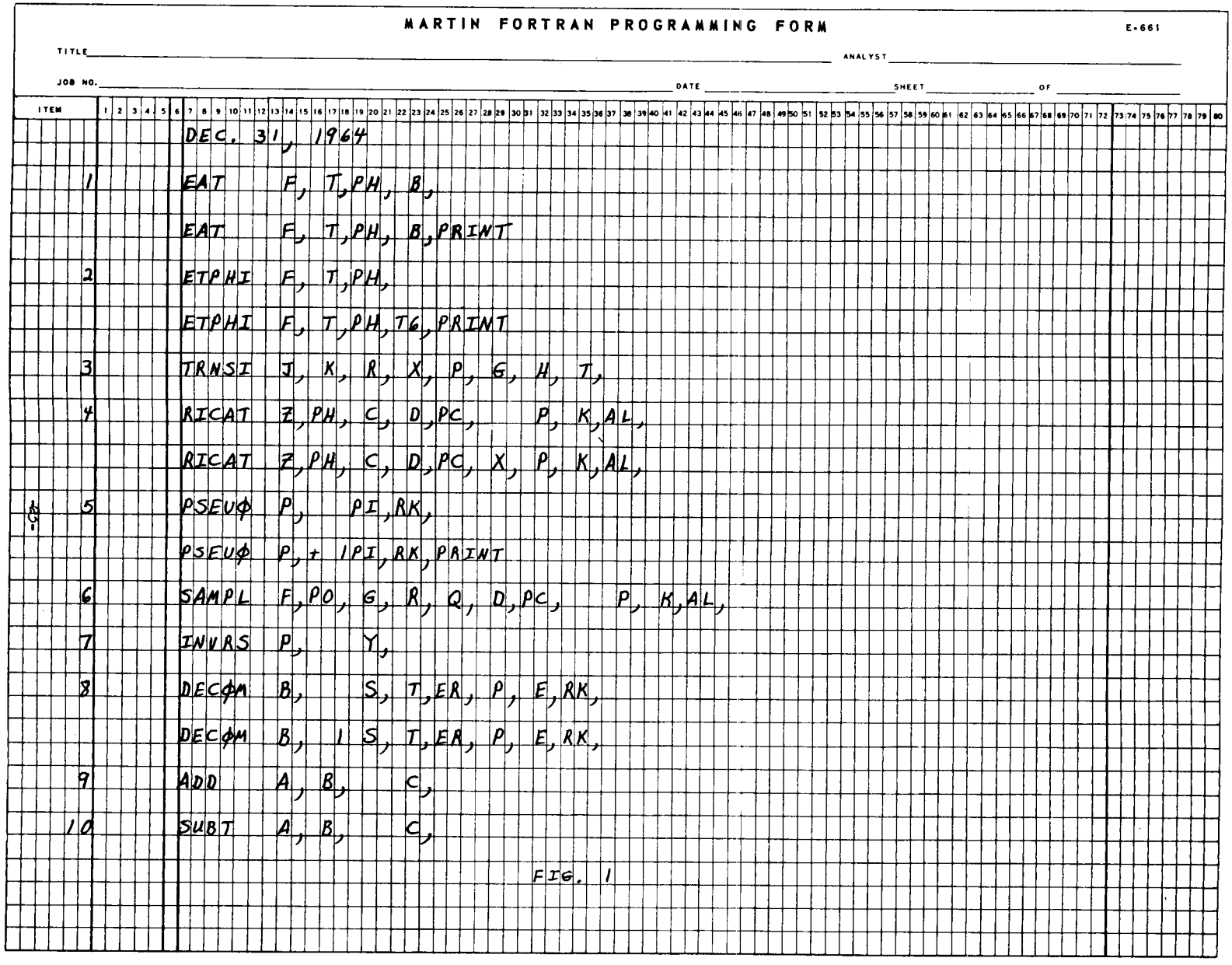
MARTIN FORTRAN PROGRAMMING FORM
E-661
TITLE
ANALYST
JOB NO.
DATE
SHEET
OF
ITEM
DEC. 31, 1964
EAT F, T,PH, B,
EAT F, T,PH, B,PRINT
ETPHI F, T,PH,
ETPHI F, T,PH,TG,PRINT
TRNSI J, K, R, X, P, G, H, T,
RICAT Z,PH, C, D,PC, P, K,AL,
RICAT Z,PH, C, D,PC, X, P, K,AL,
PSEUΦ P, PI,RK,
PSEUΦ P,+1PI,RK,PRINT
SAMPL F,PO, G, R, Q, D,PC, P, K,AL,
INVRS P, Y,
DECΦM B, S, T,ER, P, E,RK,
DECΦM B, 1 S, T,ER, P, E,RK,
ADD A, B, C,
SUBT A, B, C,
FIG. 1

Fig. 1

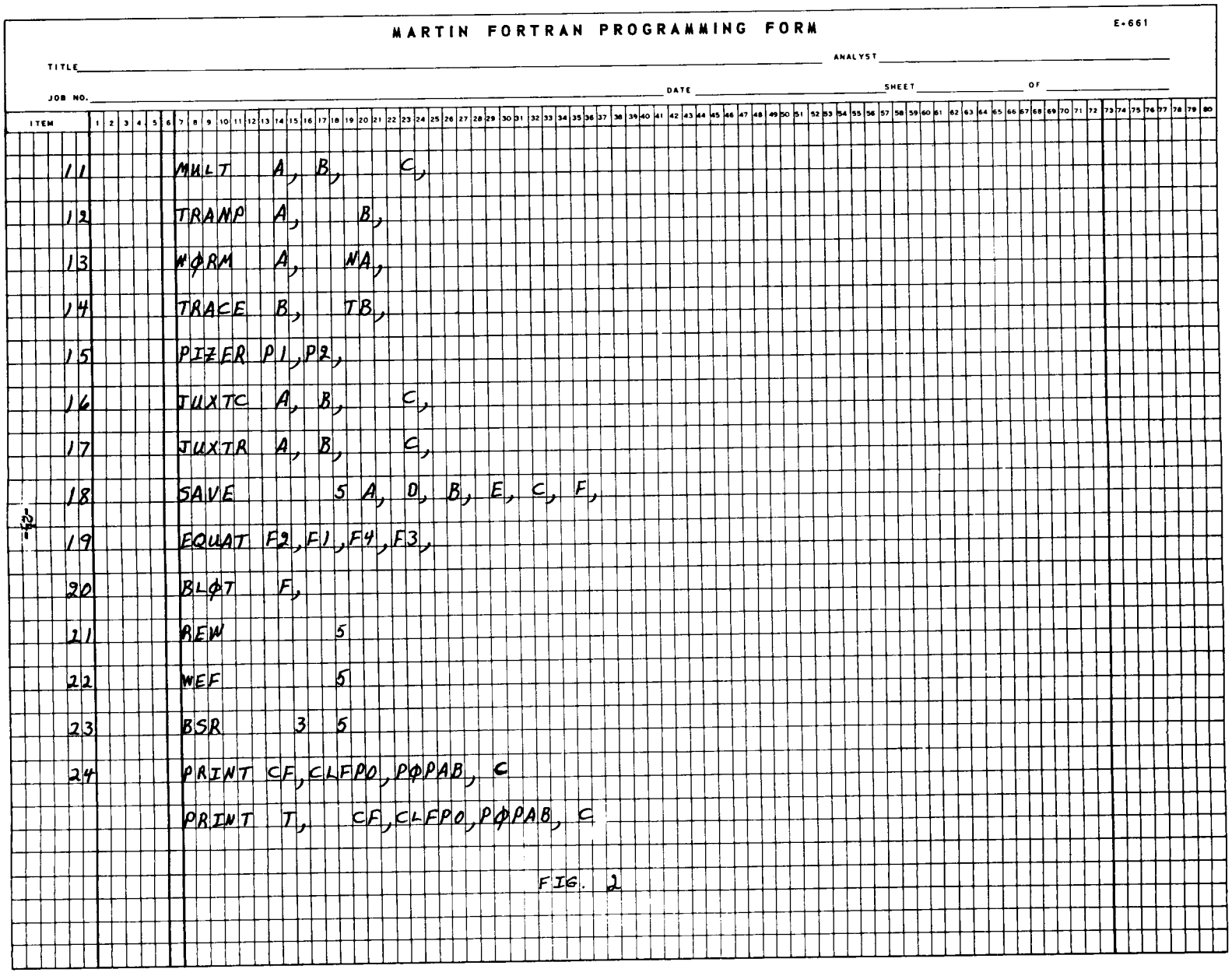
MARTIN FORTRAN PROGRAMMING FORM
E-661
TITLE
ANALYST
JOB NO.
DATE
SHEET
OF
ITEM
11 MULT A, B, C,
12 TRAMP A, B,
13 WORM A, WA,
14 TRACE B, TB,
15 PIZER P1, P2,
16 JUXTC A, B, C,
17 JUXTR A, B, C,
18 SAVE 5 A, D, B, E, C, F,
19 EQUAT F2, F1, F4, F3,
20 BLOT F,
21 REW 5
22 WEF 5
23 BSR 3 5
24 PRINT CF, CLFPO, POPAB, C
PRINT T, CF, CLFPO, POPAB, C

FIG. 2

Fig. 2

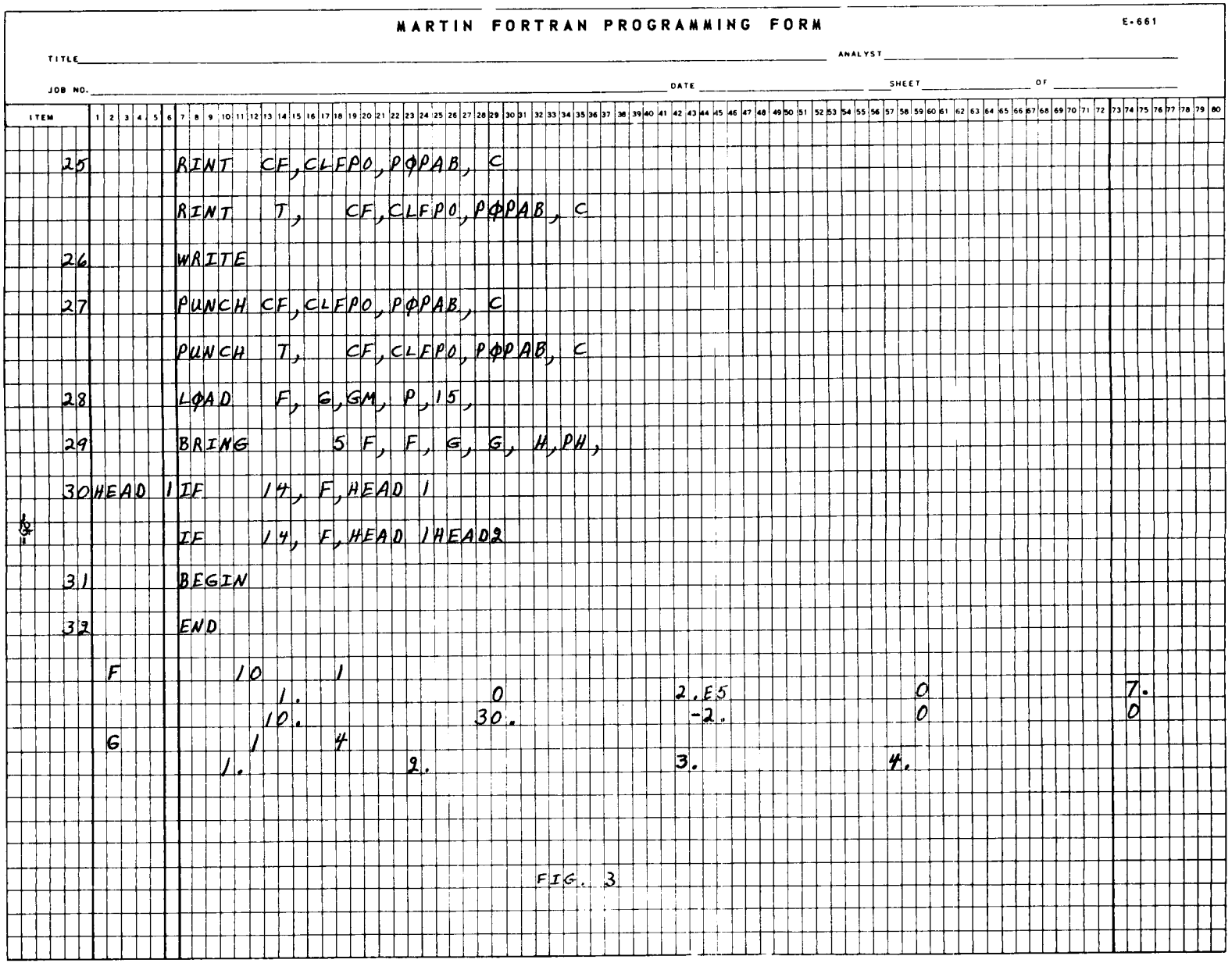
MARTIN FORTRAN PROGRAMMING FORM
E-661
TITLE
ANALYST
JOB NO.
DATE
SHEET
OF
ITEM
25 RINT CF,CLFPO,PΦPAB, C
RINT T, CF,CLFPO,PΦPAB, C
26 WRITE
27 PUNCH CF,CLFPO,PΦPAB, C
PUNCH T, CF,CLFPO,PΦPAB, C
28 LΦAD F, G,GM, P,15,
29 BRING 5 F, F, G, G, H,PH,
30HEAD 1IF 14, F,HEAD 1
IF 14, F,HEAD 1HEAD2
31 BEGIN
32 END
F
10 1
1. 0
10. 30.
2.E5 0
-2. 0
7.
0
6
1 4
1. 2.
3. 4.
FIG. 3

Fig. 3

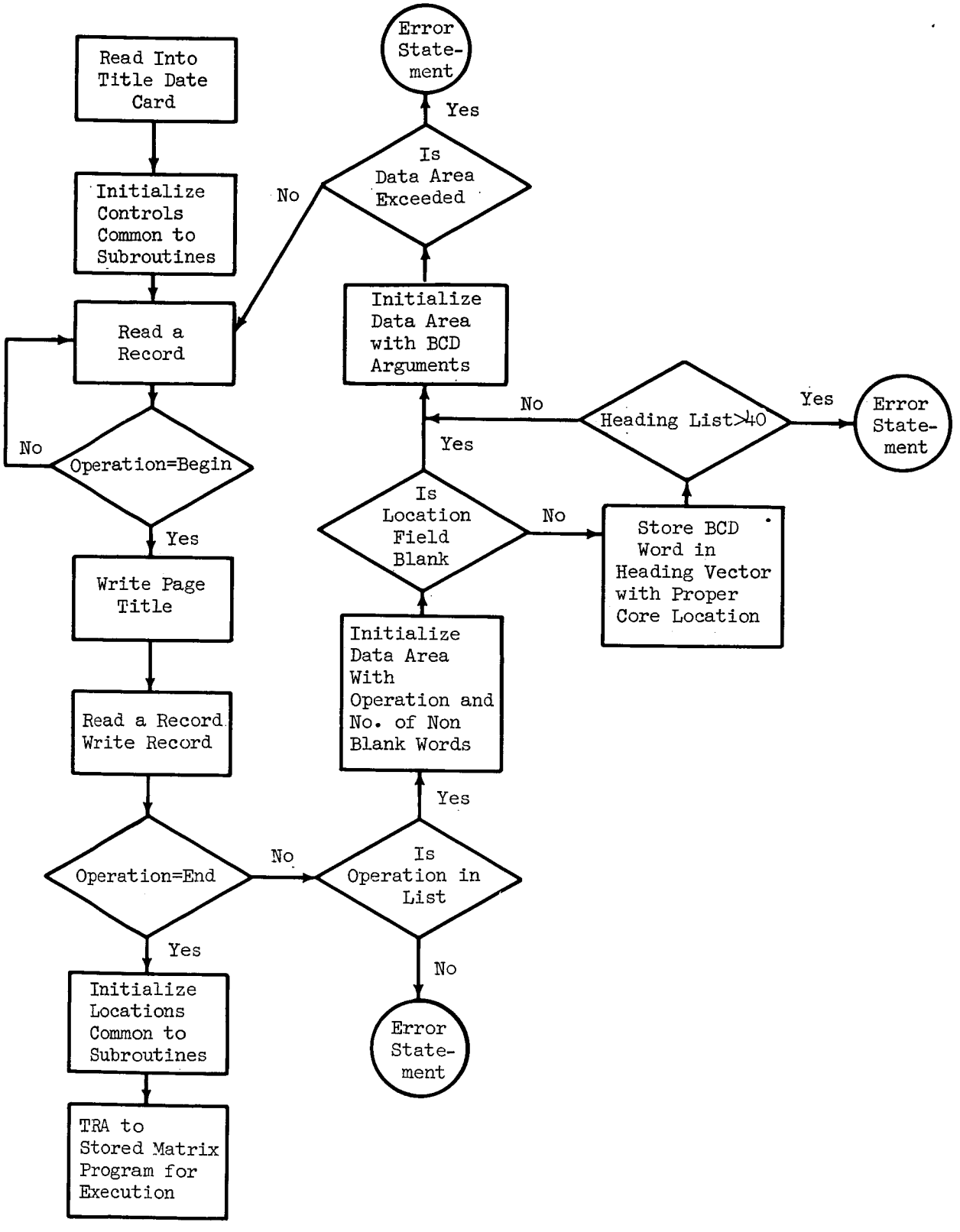
Read Into Title Date Card
Initialize Controls Common to Subroutines
Read a Record
Operation=Begin
No
Yes
Write Page Title
Read a Record Write Record
Operation=End
No
Yes
Initialize Locations Common to Subroutines
TRA to Stored Matrix Program for Execution
Is Operation in List
No
Error Statement
Yes
Initialize Data Area With Operation and No. of Non Blank Words
Is Location Field Blank
No
Store BCD Word in Heading Vector with Proper Core Location
Yes
Heading List>40
No
Yes
Error Statement
Initialize Data Area with BCD Arguments
Is Data Area Exceeded
No
Yes
Error Statement

Fig. 4

```
BEGIN
LOAD   F, G,T1,T2, K, J, R, X, H, T
EAT    F,T1,PH,IN,PRINT
EAT    F,T2,FH,I2,PRINT
WRITE      END OF INSTRUCTION 1
EIPHI  F,T1,P1,   PRINT
EIPHI  F,T2,P2,   PRINT
EIPHI  F,T2,P3,I3,PRINT
WRITE      END OF INSTRUCTION 2
RINT   I3,I3
MULT   IN, G,   G1,
TRNSI  J, K, R, X,PH,G1, H, T
RINT   X, X
WRITE      END OF INSTRUCTION 3
END
```

FIG. 5

THE FOLLOWING DATA HAS BEEN STORED IN THE INDICATED MATRIX STORAGE.

MATRIX F

5 5

0.	0.	0.	0.	0.	0.
0.50000000E 00	0.	0.	0.	0.	0.
0.09999999E 01	0.09999999E 01	0.	0.	0.	0.
0.09999999E 01	0.	0.	0.	0.	0.
0.20000000E 01					

MATRIX G

5 2

0.09999999E 01	0.	0.09999999E 01	0.09999999E 01	0.	0.
0.09999999E 01	0.	0.	0.09999999E 01		

MATRIX T1

1 1

0.50000000E 00

MATRIX T2

1 1

0.59999999E 01

MATRIX K

2 5

0.09999999E 01	0.	0.50000000E 00	0.	0.20000000E 01	0.
0.30000000E 01	0.	0.09999999E 01	0.		

MATRIX J

2 1

0. 0.

MATRIX R

1 1

0.

MATRIX X

5 1

0.09999999E 01 0.09999999E 01 0.09999999E 01 0. 0.09999999E 01

MATRIX H

7 5

0.09999999E 01	0.	0.	0.	0.	0.
0.09999999E 01	0.	0.	0.	0.	0.
0.09999999E 01	0.	0.	0.	0.	0.
0.09999999E 01	0.	0.	0.	0.	0.
0.09999999E 01	0.09999999E 01	0.	0.	0.	0.
0.	0.09999999E 01	0.	0.09999999E 01	0.	

MATRIX T

4 1

0.20000000E 01 0.50000000E 00 0. 0.34999999E 01

FIG.7

TRANSITION MATRIX PH= EXP(F * T1)

T1 = 0.50000000E 00 T1 /2**K = 0.50000E 00 K = 0

NUMBER OF ROWS 5 NUMBER OF COLUMNS 5

EXPONENT 0

1.0000	0.0000	0.0000	0.0000	0.0000
0.0000	1.2840	0.0000	0.0000	0.0000
0.0000	0.0000	1.6487	0.8244	0.0000
0.0000	0.0000	0.0000	1.6487	0.0000
0.0000	0.0000	0.0000	0.0000	2.7183

FIG. 8

INTEGRAL EXPONENTIAL IN = INT EXP(F * T1)

T1 = 0.50000C00E 00 T1 72**K = 0.50000E 00 K = 0

NUMBER OF ROWS 5 NUMBER OF COLUMNS 5

EXPONENT -1

5.0000	0.0000	0.0000	0.0000	0.0000
0.0000	5.6805	0.0000	0.0000	0.0000
0.0000	0.0000	6.4872	1.7564	0.0000
0.0000	0.0000	0.0000	6.4872	0.0000
0.0000	0.0000	0.0000	0.0000	8.5914

FIG. 9

TRANSITION MATRIX FH= EXP(F * T2)

T2 = 0.59999999E 01 T2 /2**K = 0.30000E 01 K = 1

NUMBER OF ROWS 5 NUMBER OF COLUMNS 5

EXPONENT 5

0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0002	0.0000	0.0000	0.0000
0.0000	0.0000	0.0040	0.0242	0.0000
0.0000	0.0000	0.0000	0.0040	0.0000
0.0000	0.0000	0.0000	0.0000	1.6275

FIG. 10

```
INTEGRAL EXPONENTIAL      I2  = INT EXP(  F * I2 )

I2 =  0.5999999E 01     I2 72**K =  0.30000E 01     K =  1

NUMBER OF ROWS  5      NUMBER OF COLUMNS  5

EXPONENT   4

 0.0006  0.0000  0.0000  0.0000  0.0000
 0.0000  0.0038  0.0000  0.0000  0.0000
 0.0000  0.0000  0.0402  0.2018  0.0000
 0.0000  0.0000  0.0000  0.0402  0.0000
 0.0000  0.0000  0.0000  0.0000  8.1377
    END OF INSTRUCTION 1
```

FIG. 11

TRANSITION MATRIX P1= EXP(F * T1)

T1 = 0.50000000E 00 T1 /2**K = 0.50000E 00 K = 0

NUMBER OF ROWS 5 NUMBER OF COLUMNS 5

EXPONENT 0

1.0000	0.0000	0.0000	0.0000	0.0000
0.0000	1.2840	0.0000	0.0000	0.0000
0.0000	0.0000	1.6487	0.8244	0.0000
0.0000	0.0000	0.0000	1.6487	0.0000
0.0000	0.0000	0.0000	0.0000	2.7183

FIG. 12

TRANSITION MATRIX P2= EXP(F * T2)

T2 = 0.59999999E 01 T2 /2**K = 0.30000E 01 K = 1

NUMBER OF ROWS 5 NUMBER OF COLUMNS 5

EXPONENT 5

0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0002	0.0000	0.0000	0.0000
0.0000	0.0000	0.0040	0.0242	0.0000
0.0000	0.0000	0.0000	0.0040	0.0000
0.0000	0.0000	0.0000	0.0000	1.6275

FIG. 13

TRANSITION MATRIX P= EXP(F * T3)

T2 = 0.599999E 01 T2 /2**K = 0.937500E 01 K = 1

NUMBER OF ROWS 5 NUMBER OF COLUMNS 5

EXPONENT 2

0.0100 0.0000 0.2000 0.0000 0.0000

0.0000 0.0448 0.0000 0.1000 0.0000

0.0000 0.0000 0.2019 0.6020 0.0000

0.0000 0.0000 0.0000 0.2009 0.0000

0.0000 0.0000 0.0000 0.0000 4.0343

END OF INSTRUCTION 2

FIG. 14

MATRIX T3

NUMBER OF ROWS 1 NUMBER OF COLUMNS 1

0.30000000E 01

FIG. 15

TRANSIENT BEHAVIOR

$U_N = J * R_N - K * X_N$ $X_{N+1} = PH * X_N + GI * U_N$

TAU1 = 0.500000E 00 TAU2 = 0.200000E 01

	TIME	R-1	R-2	Y-1	Y-2	Y-3	Y-4	Y-5	Y-6	Y-7	U-1	U-2	U-3
*	0.00	0.0000		1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	-3.5000	-3.0000	
	0.50	0.0000		-0.7500	-2.4063	1.0340	-2.2705	0.1409	-0.7500	-4.6788	-3.5000	-3.0000	
	1.00	0.0000		-2.5000	-6.7847	-0.7617	-6.0140	-2.1945	-2.5000	-12.7966	-3.5000	-3.0000	
	1.50	0.0000		-4.2500	-12.4040	-6.8613	-12.1858	-4.5428	-4.2500	-24.5899	-3.5000	-3.0000	
*	2.00	0.0000		-6.0000	-19.6194	-21.3725	-22.3617	-25.7991	-6.0000	-41.9511	68.5845	81.2198	
	2.50	0.0000		28.2922	59.9047	-42.6147	7.6240	-9.3499	28.2922	67.5287	68.5845	81.2198	
	3.00	0.0000		62.5845	162.0156	-51.9287	57.0620	58.5242	62.5845	219.0776	68.5845	81.2198	

FIG. 16

MATRIX X

NUMBER OF ROWS 5 NUMBER OF COLUMNS 1

0.62584461E 02

0.16201556E 03

-0.51928743E 02

0.57062043E 02

0.68828239E 02

END OF INSTRUCTION 3

FIG. 17

```
BEGIN
LOAD  PO, F, C, D,PC,PK, T, X
BIPHI  F, T,PH    PRINT
TRANP PH,   PT
RICAT PO,PH, C, D,PC,    P, K,AL
RICAT PO,PT, C, D,PK, X,PO, K,AL
RINT  PO, P  K, K AL,AL  P,PI
WRITE      END OF INSTRUCTION 4
PSEUO  P,  IPI,RK,PRINT
MULT  PI, P,    I,
MULT   P, I,   P1,
PRINT  I, I PI,PI
WRITE      END OF INSTRUCTION 5
END
```


FIG. 18

THE FOLLOWING DATA HAS BEEN STORED IN THE INDICATED MATRIX STORAGE.

MATRIX PO

4 4

0.	0.	0.	0.	0.	0.09999999E-00
0.	0.	0.	0.	0.50000000E 00	0.
0.	0.	0.	0.		

MATRIX F

6 6

0.09999999E 01	0.	0.	0.	0.	0.
0.	0.	0.	0.09999999E 01	0.	0.
0.	0.	0.	0.	0.	0.
0.09999999E 01	0.	0.	0.	0.	0.
0.	0.	0.	0.09999999E 01	0.	0.
0.	0.	0.	0.	0.	-0.09999999E 01
-0.09999999E 01	0.	0.	0.	0.	0.
0.	0.	0.	-0.09999999E 01	0.	0.
0.	0.	0.	0.	0.	0.
-0.09999999E 01	0.	-0.09999999E 01	0.	0.	0.
0.	0.	0.	-0.09999999E 01		

MATRIX C

1 4

0.09999999E 01	-0.09999999E 01	0.	0.

MATRIX D

3 1

0.	0.09999999E 01	0.50000000E 01

MATRIX PC

4 1

0.	0.25000000E 01	0.	0.

FIG. 19

MATRIX PK

4 1

0.09999999E 01 0.09999999E 01 0.09999999E 01 0.09999999E 01

MATRIX T

1 1

0.09999999E 01

MATRIX X

4 4

0.20000000E 01	0.	0.	0.	0.	0.20000000E 01
0.	0.	0.	0.	0.20000000E 01	0.
0.	0.	0.	0.20000000E 01		

FIG. 20

TRANSITION MATRIX PH= EXP(F * T)

T = 0.09999999E 01 T /2**K = 0.10000E 01 K = 0

NUMBER OF ROWS 8 NUMBER OF COLUMNS 8

EXPONENT 0

2.7183	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	2.7183	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	2.7183	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	2.7183	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	-1.1752	0.3679	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.3679	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3679	0.0000
-1.1752	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3679

FIG. 21

MATRIX P(2)

TAU = 0.1000E 01 EPSILON = 0.

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

EXPONENT -1

0.0000 0.0000 0.0000 -4.9084

0.0000 0.0183 0.0000 0.0000

0.0000 0.0000 0.0916 0.0000

-4.9084 0.0000 0.0000 0.0000

ALPHA = 0.63890574E 01

FIG. 22

MATRIX A ** -1 * A

WHERE A = (G11 * X + G21 * P0(I))

0.99999999E 00	0.	-0.	-0.
0.	0.99999999E 00	0.	0.
0.	-0.	0.99999999E 00	0.
0.	-0.	0.	0.99999999E 00

FIG. 23

MATRIX PO(1)

TAU = 0.1000E 01 EPSILON = 0.

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

EXPONENT -2

0.0000 0.0000 0.0000 0.0000

0.0000 0.6767 0.0000 0.0000

0.0000 0.0000 3.3834 0.0000

0.0000 0.0000 0.0000 0.0000

ALPHA = 0.13778113E 02

FIG. 24

MATRIX K(1)

NUMBER OF ROWS 1 NUMBER OF COLUMNS 4

EXPONENT -3

0.0000 -6.7668 0.0000 0.0000

FIG. 25

MATRIX PU(2)

TAU = 0.1000E 01 EPSILON = 0.

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

EXPONENT -3

0.0000 0.0000 0.0000 0.0000

0.0000 0.9158 0.0000 0.0000

0.0000 0.0000 4.5789 0.0000

0.0000 0.0000 0.0000 0.0000

ALPHA = 0.63890563E 01

FIG. 26

MATRIX K(2)

[illegible] 1 NUMBER OF COLUMNS 4

EXPONENT -4

0.0000 -9.1578 0.0000 0.0000

FIG. 27

MATRIX P0(3)

TAU = 0.1000E 01 EPSILON = 0.

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

EXPONENT -4

0.0000 0.0000 0.0000 0.0000
0.0000 1.2394 0.0000 0.0000
0.0000 0.0000 6.1959 0.0000
0.0000 0.0000 0.0000 0.0000

ALPHA = 0.63390564E 01

FIG. 28

MATRIX K(3)

NUMBER OF ROWS 1 NUMBER OF COLUMNS 4

EXPONENT -4

0.0000 -1.2394 0.0000 0.0000

FIG. 29

MATRIX P

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

0.	-0.	0.	0.
-0.	0.12393758E-03	0.	0.
0.	0.	0.61968793E-03	0.
0.	0.	0.	0.

MATRIX K

NUMBER OF ROWS 1 NUMBER OF COLUMNS 4

0.	-0.12393758E-03	0.	0.

MATRIX AL

NUMBER OF ROWS 1 NUMBER OF COLUMNS 1

0.63890564E 01

FIG. 30

MATRIX P1

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

-0.	-0.	-0.	-0.49876054E-00
-0.	0.24787508E-03	0.	0.
-0.	0.	0.12393754E-02	0.
-0.49876054E-00	0.	0.	0.

END OF INSTRUCTION 4

FIG. 31

PSEUDO-INVERSE OF MATRIX P

RANK = 4 FINAL PIVOTAL ELEMENT = 0.61442056E-07

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

EXPONENT 3

0.0000	0.0000	0.0000	-0.0020
0.0000	4.0343	0.0000	0.0000
0.0000	0.0000	0.3069	0.0000
-0.0020	0.0000	0.0000	0.0000

RANK	PIVOTAL ELEMENT	RHO
2	0.24876207E-00	0.61896823E-05
3	0.15360513E-05	0.25444183E-06
4	0.61442056E-07	0.22351742E-07

FIG. 32

```
MATRIX     I

NUMBER OF ROWS  4      NUMBER OF COLUMNS  4

EXPONENT   0

 1.0000   0.0000   0.0000   0.0000
 0.0000   1.0000   0.0000   0.0000
 0.0000   0.0000   1.0000   0.0000
 0.0000   0.0000   0.0000   1.0000


MATRIX    P1

NUMBER OF ROWS  4      NUMBER OF COLUMNS  4

EXPONENT  -1

 0.0000   0.0000   0.0000  -4.9876
 0.0000   0.0025   0.0000   0.0000
 0.0000   0.0000   0.0124   0.0000
-4.9876   0.0000   0.0000   0.0000
    END OF INSTRUCTION 5
```

FIG. 33

```
BEGIN
LOAD   F,PH, G, R, H,PG, D
TRANP  H,   HT,
MULT   H,HT,    D,
SAMPL  F,PG, G, R, H, D,PG,    P, K,AL
WRITE      'END OF INSTRUCTION 5'
END
```

FIG. 34

THE FOLLOWING DATA HAS BEEN STORED IN THE INDICATED MATRIX STORAGE.

MATRIX F

4 4

0.09999999E 01	0.	0.	0.	0.	0.09999999E 01
0.	0.	0.	0.	0.09999999E 01	0.
0.	0.	0.	0.09999999E 01		

MATRIX PO

4 4

0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.		

MATRIX G

1 4

0.	0.	0.	0.09999999E 01

MATRIX R

1 1

0.09999999E 01

MATRIX H

4 4

0.	0.	0.	0.50000000E 00	0.	0.
0.09999999E 01	0.	0.	0.20000000E 01	0.	0.
0.09999999E 01	0.	0.	0.		

MATRIX PC

4 1

0.20000000E 01	0.09999999E 01	0.99999999E-08	0.49999999E 01

FIG. 35

```
MATRIX  D
                    2       1
   0.                       0.09999999E 01
```


FIG. 36

MATRIX K(0)

NUMBER OF ROWS 4 NUMBER OF COLUMNS 1

EXPONENT 0

0.0000

0.0000

0.0000

0.0000

FIG. 37

MATRIX P(1)

TAU = 0.1000E 01 EPSILON = 0.

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

EXPONENT 0

0.2500 0.0000 0.0000 0.0000
0.0000 1.0000 0.0000 0.0000
0.0000 0.0000 4.0000 0.0000
0.0000 0.0000 0.0000 1.0000

ALPHA = 0.09999999E 01

FIG. 38

MATRIX P(2)

TAU = 0.1000E 01 EPSILON = 0.

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

EXPONENT 0

0.5000 0.0000 0.0000 0.0000
0.0000 2.0000 0.0000 0.0000
0.0000 0.0000 8.0000 0.0000
0.0000 0.0000 0.0000 1.5000

ALPHA = 0.47916666E-00

FIG. 39

MATRIX K(2)

NUMBER OF ROWS 4 NUMBER OF COLUMNS 1

EXPONENT -1

0.0000

0.0000

0.0000

6.0000

FIG. 40

MATRIX P(3)

TAU = 0.1000E 01 EPSILON = 0.

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

EXPONENT 1

0.0750 0.0000 0.0000 0.0000
0.0000 0.3000 0.0000 0.0000
0.0000 0.0000 1.2000 0.0000
0.0000 0.0000 0.0000 0.1600

ALPHA = 0.30835734E-00

FIG. 41

MATRIX P(4)

TAU = 0.1000E 01 EPSILON = 0.

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

EXPONENT 1

0.1000	0.0000	0.0000	0.0000
0.0000	0.4000	0.0000	0.0000
0.0000	0.0000	1.6000	0.0000
0.0000	0.0000	0.0000	0.1615

ALPHA = 0.23282313E-00

FIG. 42

MATRIX K(4)

NUMBER OF ROWS 4 NUMBER OF COLUMNS 1

EXPONENT -1

0.0000

0.0000

0.0000

6.1765

FIG. 43

```
MATRIX     P(   5)

TAU =  0.1000E 01     EPSILON =  0.

NUMBER OF ROWS  4     NUMBER OF COLUMNS  4

EXPONENT  1

0.1250  0.0000  0.0000  0.0000
0.0000  0.5000  0.0000  0.0000
0.0000  0.0000  2.0000  0.0000
0.0000  0.0000  0.0000  0.1618

                    ALPHA =  0.13847168E-00
 END OF INSTRUCTION 6
```


FIG. 44

```
        BEGIN
        LOAD   D, R, F,P1,P2,P3
        INVRS  F,    FI,
        RINT   FI,FI
        WRITE        END OF INSTRUCTION 7
        TRANP  D,    DT
        MULT   D,DT,     B
        DECOM  B,   1 S,TZ,ER,  P,  E,RK,
        RINT   RK,RNKER,ER   P,  P TZ,TZ
        DECOM  B,    IS,IX,EK,IP,  E,R2,
        RINT   R2,R2 EK,EK IP,  P
        WRITE        END OF INSTRUCTION 8
        ADD    D, D,    2D
        SUBT   2D, D,     D
        PRINT  2D,2D   D   D
        WRITE        END OF INSTRUCTIONS 9 AND 10
        MULT   S,ER,     E1
        TRANP  E1,    1E
        MULT   E1,1E,    MB
        PRINT  MB,MB   B,  B
        WRITE        END OF INSTRUCTIONS 11 AND 12
        SUBT   B,MB,     1,
        NORM   1,    N1,
        PRINT  N1,NMB
        WRITE        END OF INSTRUCTION 13
        TRACE  3,    TB,
        PRINT  TB,TB
        WRITE        END OF INSTRUCTION 14
        PIZER  P1,P2
        PSEUO  F,    IF1,R1,PRINT
        PSEUO  F,    F1,R1,PRINT
        PIZER  P3,P2
        PSEUO  F,    F1,R1,PRINT
        WRITE        END OF INSTRUCTION 15
        JUXTC  F,  R,    FR
        JUXTR  FR,FR,    RF
        PRINT  RF,RF
        WRITE        END OF INSTRUCTIONS 15 AND 17
        REW          2
        WEF          2
        REW          2
        REW          5
        WEF          5
        REW          5
        LOAD   T1,T2,T3
LOOP    SAVE         5T1,  T   F,  F,F1,F1
        SAVE         2T1,  T
        ADD    T1,T2,    T1
        IF     T3,T1,LOOP
        REW          5
        EQUAT  T3,P3
        BLOT   DT,
        BSR      3   2
        PRINT  P3,P3
        RINT    F,  F
```

FIG. 45

```
        WRITE       END OF INSTRUCTIONS 18 THROUGH 26.
        WRITE       ONE SHOULD OBSERVE IN THE FOLLOWING OPERATIONS THAT THE
        WRITE    ORIGINAL P3 WAS 1.E-7.  ALSO THAT AN F WILL BE BROUGHT OFF
        WRITE    B=5 AND STORED INTO F1.  THIS CAN BE DONE BECAUSE F1  AND
        WRITE    F HAVE THE SAME DIMENSIONS.  IN ADDITION F1 WILL BE STORED
        WRITE    IN A SYMBOLIC AREA CALLED DT.  THIS DT WILL HAVE DIMENSION
        WRITE    3 X 3 WHEREAS THE PREVIOUS DT HAD DIMENSION 3 X 4.  THERE
        WRITE    IS NO ERROR IN THIS OPERATION DUE TO THE BLOT .
        LOAD  P1
LOOP1   BRING       5 F,F1,F1,DT
        BRING       2 T,T1
        RINT  T1,   F1,F1 DT,DT
        IF    T1,P1,OUT   LOOP1
OUT     WRITE       END OF INSTRUCTIONS 28, 29 AND 30.
        END
```


FIG 46

THE FOLLOWING DATA HAS BEEN STORED IN THE INDICATED MATRIX STORAGE.

MATRIX D

4 3

0.09999999E 01 0.09999999E 01 0.09999999E 01 0.09999999E 01 0.20000000E 01 0.30000000E 01
0.49999999E 01 0.09999999E 01 0.59999999E 01 0.30000000E 01 0.40000000E 01 0.80000000E 01

MATRIX R

3 3

0.20000000E 01 0.40000000E 01 0.59999999E 01 0.80000000E 01 0.09999999E 02 0.12000000E 02
0.13999999E 02 0.16000000E 02 0.18000000E 02

MATRIX F

3 3

0.80999999E 02 0. 0. 0. 0.09999999E 05 0.
0. 0. 0.99999999E-02

MATRIX P1

1 1

0.99999999E-04

MATRIX P2

1 1

0.09999999E 01

MATRIX P3

1 1

0.99999999E-07

FIG. 47

MATRIX FI

NUMBER OF ROWS 3 NUMBER OF COLUMNS 3

0.12345678E-01	-0.	-0.
-0.	0.99999999E-04	0.
-0.	0.	0.09999999E 03

END OF INSTRUCTION 7

MATRIX RNK

NUMBER OF ROWS 3 NUMBER OF COLUMNS 1

0.30000000E 01

0.42857141E-00

0.83446503E-05

FIG. 48

MATRIX ER

NUMBER OF ROWS 4 NUMBER OF COLUMNS 3

0.09999999E 01	0.	0.
0.	0.	0.
0.	0.09999999E 01	0.
0.	0.	0.09999999E 01

MATRIX P

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

0.09999999E 01	0.	0.	0.
0.	0.	0.09999999E 01	0.
0.	0.	0.	0.09999999E 01
0.	0.09999999E 01	0.	0.

FIG. 49

MATRIX TZ

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

0.15275252E 01	0.	-0.93521950E-01	-0.18704391E-00
-0.42857137E-00	0.09999999E 01	0.14285713E-00	-0.42857143E-00
-0.	-0.	0.24409476E-00	-0.22139719E-00
0.	0.	0.	0.10599978E-00

MATRIX R2

NUMBER OF ROWS 3 NUMBER OF COLUMNS 1

0.20000000E 01

0.11561798E 02

0.61224651E 00

FIG. 50

MATRIX EK

NUMBER OF ROWS 4 NUMBER OF COLUMNS 2

0.	0.
0.	0.
0.09999999E 01	0.
0.	0.09999999E 01

MATRIX P

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

0.	0.	0.09999999E 01	0.
0.	0.	0.	0.09999999E 01
0.09999999E 01	0.	0.	0.
0.	0.09999999E 01	0.	0.

END OF INSTRUCTION 8

FIG. 51

MATRIX 20

NUMBER OF ROWS 4 NUMBER OF COLUMNS 3

EXPONENT 1

0.2000 0.2000 0.2000

0.2000 0.4000 0.6000

1.0000 0.2000 1.2000

0.6000 0.8000 1.6000

MATRIX 0

NUMBER OF ROWS 4 NUMBER OF COLUMNS 3

EXPONENT 0

1.0000 1.0000 1.0000

1.0000 2.0000 3.0000

5.0000 1.0000 6.0000

3.0000 4.0000 8.0000

END OF INSTRUCTIONS 9 AND 10

FIG. 52

MATRIX MB

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

EXPONENT 1

0.3000 0.6000 1.2000 1.5000
0.6000 1.4000 2.5000 3.5000
1.2000 2.5000 6.2000 6.7000
1.5000 3.5000 6.7000 8.9000

MATRIX B

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

EXPONENT 1

0.3000 0.6000 1.2000 1.5000
0.6000 1.4000 2.5000 3.5000
1.2000 2.5000 6.2000 6.7000
1.5000 3.5000 6.7000 8.9000
END OF INSTRUCTIONS 11 AND 12

FIG. 53

MATRIX NMB

NUMBER OF ROWS 1 NUMBER OF COLUMNS 1

EXPONENT -6

8.3447
END OF INSTRUCTION 13

MATRIX TB

NUMBER OF ROWS 1 NUMBER OF COLUMNS 1

EXPONENT 2

1.6800
END OF INSTRUCTION 14

FIG. 54

PSEUDO-INVERSE OF MATRIX F

RANK = 3 FINAL PIVOTAL ELEMENT = 0.99999997E-04

NUMBER OF ROWS 3 NUMBER OF COLUMNS 3

EXPONENT 2

0.0001 0.0000 0.0000

0.0000 0.0000 0.0000

0.0000 0.0000 1.0000

RANK	PIVOTAL ELEMENT	RHO
1	0.09999999E 09	0.65632350E-04
2	0.65610000E 04	0.37252903E-07
3	0.99999997E-04	0.22351742E-07

FIG. 55

PSEUDO-INVERSE OF MATRIX F

RANK = 1 FINAL PIVOTAL ELEMENT = 0.09999999E 09

NUMBER OF ROWS 3 NUMBER OF COLUMNS 3

EXPONENT -5

0.0000 0.0000 0.0000

0.0000 10.0000 0.0000

0.0000 0.0000 0.0000

FIG. 56

```
PSEUDO-INVERSE OF MATRIX   F

RANK =  2              FINAL PIVOTAL ELEMENT = 0.65610000E 04

NUMBER OF ROWS  3      NUMBER OF COLUMNS  3

EXPONENT  -2

 1.2346   0.0000   0.0000
 0.0000   0.0100   0.0000
 0.0000   0.0000   0.0000
    END OF INSTRUCTION 15
```

FIG. 57

MATRIX RF

NUMBER OF ROWS 6 NUMBER OF COLUMNS 6

EXPONENT 4

0.0081	0.0000	0.0000	0.0002	0.0004	0.0006
0.0000	1.0000	0.0000	0.0008	0.0010	0.0012
0.0000	0.0000	0.0000	0.0014	0.0016	0.0018
0.0081	0.0000	0.0000	0.0002	0.0004	0.0006
0.0000	1.0000	0.0000	0.0008	0.0010	0.0012
0.0000	0.0000	0.0000	0.0014	0.0016	0.0018

END OF INSTRUCTIONS 16 AND 17

THE FOLLOWING DATA HAS BEEN STORED IN THE INDICATED MATRIX STORAGE.

MATRIX T1

1 1

0.

MATRIX T2

1 1

0.09999999E 01

MATRIX T3

1 1

0.30000000E 01

FIG. 59

```
MATRIX   P3

NUMBER OF ROWS  1     NUMBER OF COLUMNS  1

EXPONENT   0

3.0000


MATRIX    F

NUMBER OF ROWS  3     NUMBER OF COLUMNS  3

   0.80999999E 02     0.                 0.

   0.                 0.99999999E 05     0.

   0.                 0.                 0.99999999E-02
   END OF INSTRUCTIONS 18 THROUGH 26.
   ONE SHOULD OBSERVE IN THE FOLLOWING OPERATIONS THAT THE
ORIGINAL P3 WAS 1.E-7.  ALSO THAT AN F WILL BE BROUGHT OFF
B-5 AND STORED INTO F1.  THIS CAN BE DONE BECAUSE F1  AND
F HAVE THE SAME DIMENSIONS.  IN ADDITION F1 WILL BE STORED
IN A SYMBOLIC AREA CALLED DT.  THIS DT WILL HAVE DIMENSION
3 X 3 WHEREAS THE PREVIOUS DT HAD DIMENSION 3 X 4.  THERE
IS NO ERROR IN THIS OPERATION DUE TO THE SLOT .
```


FIG. 60

THE FOLLOWING DATA HAS BEEN STORED IN THE INDICATED MATRIX STORAGE.

```
MATRIX  P1

                1       1

    0.20000000E 01
```

FIG. 61

MATRIX FI (0.09999999E 01)

NUMBER OF ROWS 3 NUMBER OF COLUMNS 3

0.80999999E 02	0.	0.
0.	0.09999999E 05	0.
0.	0.	0.99999999E-02

MATRIX DT (0.09999999E 01)

NUMBER OF ROWS 3 NUMBER OF COLUMNS 3

0.12343678E-01	0.	0.
0.	0.99999997E-04	0.
0.	0.	0.

FIG. 62

MATRIX FI (0.20000000E 01)

NUMBER OF ROWS 3 NUMBER OF COLUMNS 3

0.80999999E 02	0.	0.
0.	0.09999999E 05	0.
0.	0.	0.99999999E-02

MATRIX DT (0.20000000E 01)

NUMBER OF ROWS 3 NUMBER OF COLUMNS 3

0.12345678E-01	0.	0.
0.	0.99999997E-04	0.
0.	0.	0.

END OF INSTRUCTIONS 26, 29 AND 30.

FIG. 63

```
BEGIN
WRITE      THE REMAINING PROBLEM ILLUSTRATES THAT IF THERE
WRITE   IS AN ERROR IN ONE PROBLEM AS DEFINED BY THE PAIR
WRITE  OF INSTRUCTIONS BEGIN AND END, THE PROGRAM WILL
WRITE  GO ON TO THE NEXT PROBLEM.
LOAD   A, B.
 MULT  A, B,    C.
ERROR IN OPERATION        MULT
```


FIG. 64

```
BEGIN
WRITE      END OF SAMPLE PROBLEM FOR DECK C ASSEMBLED JANUARY  196
END
   END OF SAMPLE PROBLEM FOR DECK C ASSEMBLED JANUARY  196
FINAL TIME 02.927 DATE  03/09/65
```

FIG. 65

CHAPTER II

THE EXPONENTIAL ROUTINES

1. Description of the Problem: We are interested in an efficient method to integrate linear differential equations, and therefore we want to compute $\exp(F\,t)$, F = const. matrix.

2. Theory: Let us consider the linear dynamical system

$$\dot{x} = F(t)x + G(t)u(t) \tag{2.1}$$

where x is an n-dimensional vector, the state and $u(.)$ is an m-dimensional vector function, the control function.

Under very mild restrictions on F and G, this system will have unique solutions depending upon the time t, the initial time t_o, the initial state $x(t_o)$, and the control function $u(.)$. It is often convenient to exhibit this dependence explicitly; we shall therefore write a solution of (2.1) in the form

$$\varphi_u(t;\, x(t_o),\, t_o). \tag{2.2}$$

The notation implies that

$$\varphi_u(t_o;\, x(t_o),\, t_o) = x(t_o)$$

and that

$$\frac{d}{dt}\,\varphi_u(t;\, x(t_o),\, t_o) = F(t)\varphi_u(t;\, x(t_o),\, t_o) + G(t)u(t).$$

It is well known [3] that the solutions of (2.1) can be expressed by means of the formula

$$(2.3) \qquad x(t) \equiv \varphi_u(t;\, x(t_o),\, t_o) = \Phi(t,\, t_o)x(t_o) + \int_{t_o}^{t} \Phi(t,\, \tau)G(\tau)u(\tau)d\tau$$

which is valid for any $x_o = x(t_o)$, t and t_o (whether or not $t > t_o$).

The matrix $\Phi(t,\, t_o)$ occurring in (2.3) is the transition matrix of (2.1) and is uniquely determined by the requirements [4]

$$(2.4) \qquad \Phi(t,\, t) = I \quad \text{for all} \quad t$$

and

$$(2.5) \qquad \frac{d}{dt}\,\Phi(t,\, t_o) = F(t)\Phi(t,\, t_o).$$

From these properties and the uniqueness of solutions of (2.1) one can show at once that

$$(2.6) \qquad \Phi^{-1}(t,\, t_o) = \Phi(t_o,\, t) \quad \text{for all} \quad t,\, t_o;$$

$$(2.7) \qquad \Phi(t_3,\, t_2)\Phi(t_2,\, t_1) = \Phi(t_3,\, t_1) \quad \text{for all} \quad t_1,\, t_2,\, t_3.$$

As an example, consider

$$\dot{x} = \begin{bmatrix} 2t - 1 & t - 1 \\ 2 - 2t & 2 - t \end{bmatrix} x$$

where

$$\Phi(t,\, 0) = \begin{bmatrix} 2e^{\frac{t^2}{2}} - e^{t} & e^{\frac{t^2}{2}} - e^{t} \\ 2e^{t} - 2e^{\frac{t^2}{2}} & 2e^{t} - e^{\frac{t^2}{2}} \end{bmatrix}$$

For a one-dimensional system, the transition function for any linear system is given by

$$\Phi(t, t_o) = e^{\int_{t_o}^{t} F(\tau)d\tau} \tag{2.8}$$

If F is constant, the transition matrix is given by the matrix exponential e^{tF}, which can be defined by the everywhere convergent power series

$$e^{tF} = \sum_{i=0}^{\infty} \frac{(tF)^i}{i!}. \tag{2.9}$$

(To show the convergence of the series, observe that for $p > 0$

$$\left\| \sum_{i=p}^{p+q} \frac{(tF)^i}{i!} \right\| \leq \sum_{i=p}^{p+q} \frac{\|tF\|^i}{i!}$$

for any norm derived from a vector norm. This shows that the matrix series for e^{tF} converges in norm whenever the scalar series for $e^{\|tF\|}$ converges; the latter however is well known to converge uniformly for $\|tF\|$ in any bounded interval, and this is equivalent to uniform component convergence of e^{tF}.)

This matrix function is of interest in this report primarily because it is the fundamental matrix of the vector differential equation

$$\dot{x} = Fx \quad (F \text{ a constant}). \tag{2.10}$$

That is [Coddington and Levinson, 1955],

$$x(t) = e^{(t - t_o)F} x(t_o). \tag{2.11}$$

This may be proved very easily by termwise differentiation of the defining series, a valid procedure by its uniform convergence.

Some other facts which can be proved about e^{TF} are [Coddington and Levinson, 1955]

(2.12) $e^{A+B} = e^{A}e^{B}$ if and only if $AB = BA$.

(2.13) $e^{TFT^{-1}} = Te^{F}T^{-1}$ for any nonsingular T.

(2.14) $\det e^{F} = e^{\text{trace } F}$.

(2.14) shows that e^{tF} is always nonsingular. We can now paraphrase (2.11) by saying that the columns of e^{tF} are n linearly independent solutions of (2.10) and thus any solution of (2.10) can be expressed as a linear combination of the columnvectors of e^{tF}.

e^{tF} is computed in the ETPHI routine, using the series (2.9). We are often interested in the forced system

(2.15) $\dot{x} = Fx + Gu(t)$ (F, G constant).

The complete solution to (2.15) can be written as

(2.16) $$x(t) = e^{tF}x(0) + \int_0^t e^{(t-\tau)F}Gu(\tau)d\tau.$$

In sampled data systems $u(.)$ is a piecewise constant and we have

(2.17) $$x(t) = e^{tF}x(0) + \int_0^t e^{(t-\tau)F}Gd\tau\ u(0)$$

over any sampling interval.

Making the substitution $s = t - \tau$, the integral in (2.17) assumes the simpler form

$$\left[\int_0^t e^{sF}ds\right]Gu(0).$$

The integral $\int_0^t e^{sF}ds = \Gamma(t)$ is computed in the E A T routine. Its concrete definition can be obtained by term-by-term integration of the defining series for e^{tF},

$$\Gamma(t) = \sum_{i=0}^{\infty} \frac{F^i t^{i+1}}{(i+1)!}. \tag{2.10}$$

If F^{-1} exists, this may be written as $F^{-1}(e^{tF} - I)$; however $\Gamma(t)$ exists even if F is singular.

3. Computation: For computing e^{tF}, the sum of at most the first thirty-seven terms of its defining series (2.1) is used. Thus we compute

$$e^{tF} = \sum_{i=0}^{36} \frac{t^i F^i}{i!}. \tag{3.1}$$

The sum (3.1) is actually formed as follows. Let T_i be the i^{th} term of the expansion: $T_0 = I$, $T_i = Ft$, etc. The sum is accumulated and T_{i+1} is obtained as

$$T_{i+1} = \frac{tF}{i+1} T_i.$$

The following motivates why thirty-seven terms are used in (2.1) and gives a condition necessary for the result to be accurate.

In the IBM 700-7000 computers a little more than eight significant decimal degits are carried when operating in the single-precision floating point mode. This imposes limitations on the accuracy of the program. Consider a scalar cosine series. We know that for any real value of the argument, the absolute value of the function is one or less. Yet if the argument

were 20, the term $\frac{20^{20}}{20!}$ is so large that the addition to it of any number less than one has no effect. That is, if any term of the series exceeds 10^8, we know that no answer will have correct digits to the right of the decimal point. Thus if we want an answer that is correct to four decimal places, no term of the series may exceed 10^4. The largest term of the e^t series is T_j where j is the smallest integer such that $\frac{t}{j+1} < 1$; therefore $T_j = \frac{t^j}{j!}$ where $j = [t]$, the greatest integer less than t. For our purposes $\frac{t^j}{j!}$ should always be less than 10^4, which implies that a conservative bound for t is 10, since $\frac{10^{10}}{10!} \approx 2756 < 10^4$.

We can apply this analysis to the matrix case by using norms. Specifically let us define our norm as

$$\|A\| = \text{Min } \{\max_i \sum_j |a_{ij}|, \max_j \sum_i |a_{ij}|\}.$$

In this norm we have

(3.2) $$\|AB\| \leq \|A\| \, \|B\| \qquad \text{and}$$

(3.3) $$|a_{ij}| \leq \|A\|.$$

We would like to say now that if $|t| \cdot \|A\| \leq 10$ then we are assured of four decimal place accuracy. (3.2) and (3.3) guarantee that if the condition is satisfied, every element of every matrix in the sum will have absolute value less than 10^4, which was a necessary condition for four-place accuracy. In addition it is necessary that we not terminate the result too soon. The last term taken will be $\frac{(tA)^{36}}{36!}$ each element of which is less than $.27 \cdot 10^{-5}$. The remainder can be majorized by a geometric series whose sum is 10^{-6}. This begins to look as if $\|tA\| < 10$ is sufficient for four-place accuracy. This however is incorrect. Clearly since $e^{10} \simeq 22000$

We can hope for no more than three place accuracy because we only have three decimal places. But the situation is even worse than this. By the time the largest term is reached the partial sum is larger than 10^4, from here on to the end of the summation, digits less than about $.5 \cdot 10^{-3}$ will be dropped and digits of this size can occur out to the term $k = 33$.

Thus for about 25 terms we are losing digits averaging $2.5 \cdot 10^{-4}$ for a total of about .006. Thus for positive eigenvalues the error is bounded by about .006. For negative eigenvalues the problem of large accumulations does not arise; here we only have the differences of numbers about 3000 which means that we should get four decimal place accuracy.

This analysis indicates that the restrictions we apply to $\|At\|$ will suffice to give us four decimal place accuracy with (algebraically) small numbers and more than two decimal places (at least six significant figures), with large numbers.

The specified norm was chosen because, being sums of absolute values, it is simple and quick to compute and because the minimization gives a fairly small norm. This is of course desirable so as to maximize the allowable step size. We could not, however, use $\|A\| = \max |a_{ij}|$ because in this norm it is not necessarily true that $\|AB\| \leq \|A\| \, \|B\|$.

E A T is computed in much the same way except that the initial term of the series is tI, not I.

The arguments concerning the maximum value of t apply here also, with some modification. If $\frac{(Ft)^k}{k!}$ is the largest term in the exponential, then $\frac{(Ft)^{k-1}}{k!}$ will be the largest term in Γ. Again this must be less than 10^4, so

$$t \leq \frac{10^4 k!}{\|Ft\|^{k-1}}. \tag{3.4}$$

This is mechanized as follows: First T is halved until $\tau = \frac{T}{2^k} < \frac{10.001}{\|F\|}$. Then if $\|F\tau\|$ is greater than 8, we can check that $\tau \leq \frac{10^4 10}{(10.001)}$. If it is, we accept τ and proceed. If $\|F\tau\|$ is less than 8 and greater than 6 we check that $\tau \leq \frac{10^4 7!}{8^6}$. If it is, we accept τ and

proceed. This continues until, if $o < \|F\tau\| < 2$, τ must be less than 10^4, and is halved until it is less than 10^4. (See Figure 2 where $C_1 = 10^4$, $C_2 = \frac{10^4 3!}{4^2}$, $C_3 = \frac{10^4 5!}{6^2}$, $C_4 = \frac{10^4 7!}{8^6}$, and $C_5 = \frac{10^4 10!}{(10.001^9}$.)

In order to compute $\Phi = e^{TF}$, then for a T which is too large, we select $\tau = \frac{T}{k^2}$ such that $\|F\tau\| < 10.001$. To compute Φ and $\Gamma = \int_0^T e^{tF}dt$, we have additional restrictions as described above. In any case we arrive at a $\tau = \frac{T}{2^k}$ which will give us the accuracy described above. We then square $e^{\tau F}$ k times, obtaining $(e^{\tau F})^{2^k} = e^{TF}$ by (2.12). The procedure with Γ is similar. It follows from the fact that

$$\int_0^{2\tau} e^{tF}dt = \int_0^{\tau} e^{tF}dt + e^{\tau F}\int_0^{\tau} e^{tF}dt.$$

This process also is iterated k times to produce

$$\Gamma = \int_0^T e^{tF}dt.$$

The procedures have their failings of course, the multiplications and additions required to bring τ up to T introduce errors. No attempt is made to control this in the machine; however if a print is requested, the τ used is printed and will indicate how many squarings were required.

An option was found to be necessary. If we wish to compute the riccati solution over a given interval we must compute the fundamental matrix of the corresponding euler-lagrange equations over a submultiple of the interval $(t - t_o)$. This is signaled by putting an extra matrix in the calling sequence (only in E T P H I, not in E A T). When we exit from E T P H I the subinterval length $\tau = \frac{t - t_o}{2^k}$ where τ will be the maximum

value to satisfy the condition that $\tau \leq \frac{10.001}{\|F\|}$ will be in the extra matrix and the exponential with that value of τ will be in PH. At first it seems that this could be handled by the ordinary process of reducing τ for the series computation and squaring up to the correct value. This is a poor procedure however, because the euler-lagrange equations have both stable and unstable roots. Although the riccati may converge, the fundamental matrix may overflow if τ is too large.

For illustrations of most of these processes, s e Chapter I, Fig. 5-18.

The summation process terminates at the k^{th} term if every element of T_k, the k^{th} term of the exponential sum is 0. By virtue of the floating point spill routine a zero matrix can occur if every element is less than approximately 10^{-40}.

Some experiments were made in this subroutine with what we have called pseudo double precision. This involved two changes: 1) All scalar products in matrix multiplication are accumulated in double precision, rounded and stored in single precision. For instance, $c_{ij} = \sum_{k=1}^{\ell} a_{ik} b_{kj}$ products $a_{ij} b_{kj}$ would be computed as the single precision product of single precision numbers, the partial sums $\sum_{k=1}^{\ell} a_{ij} b_{kj}$, however, would be kept as the double precision sum of single precision numbers. After the double precision c_{ij} was obtained it was rounded to obtain the single precision value for storage. 2) These single precision terms are accumulated as above in double precision when the exponential is completed, the double precision matrix is rounded to single precision for storage. Theoretically of course, this procedure is meaningless, for instance if we add 10^{9}, 10^{-2}, and -10^{9}, in that order, in single precision we get zero. Using pseudo double precision we get 10^{-2}. But the immediate objection is that the errors in any single precision representation of 10^{9} are larger than 10^{-2}.

The results of a few experiments tend to corroborate these ideas. Most of the results achieved with the double precision code showed markedly greater significance, perhaps one more decimal place. However, in the most complicated check, the magnitude of one error actually increased. Tentatively one might say that when working with input numbers, this procedure is very helpful, but after a long sequence of computations, the mechanics of truncation create a situation where ordinary floating point is essentially the optimal computation procedure.

4. Checks. (A) The exponential and integral exponential were computed for the 7×7 matrix diag (-10, -4, -1, 0, 1, 4, 10), with $T = 1$. The results appear in Fig. 3-A and 4-A, the correct answers in Fig. 3-B and 4-B. Because the matrix is diagonal, the error analysis applied exactly and the answers show this very well. In the submatrix (-10), where we are not only at the limit of the acceptable range but are taking differences, we barely have four place accuracy and no correct significant figures. Where differences are not being taken as in (10) the answer is correct to seven significant figures.

One phenonemon that occurs repeatedly in series computation is that in the exponential of the positive eigenvalues, the computed value will be invariably less than the correct value because the error is caused by truncation of positive terms. With the negative eigenvalues the error does not display this characteristic.

The exponential of this matrix for $T = 1$ was also computed using the pseudo double precision code. This gave the result

$$e^F = \text{diag } [4.4255785E\text{-}5,\ 1.8315656E\text{-}2,\ 3.6787944E\text{-}1,$$

$$9.9999999E\text{-}1,\ 2.7182817,\ 5.4598150E2,$$

$$2.025454E4.]$$

(B) The exponential and integral exponential for a 15×15 Jordan block with eigenvalue zero were computed with $T = .1$, 1., and 10. The results were very good, probably because the series truncates. The answers were correct to more than seven decimal places. (See Fig. 5-10.)

(C) The exponential of the 7×7 skew-symmetric matrix.

$$\begin{matrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{matrix}$$

was computed for $T = 1$. This matrix should have been orthogonal. Multiplying by its transpose we obtained the identity matrix with errors of less than 2.10^{-7}. (See Fig. 11.)

Using the double precision code we obtained an identity matrix with errors of 2.10^{-8}.

(D) Using this orthogonal matrix we computed the exponential and integral exponential of $0A0^{-1}$ where A was the matrix diag (-10, -4, -1, 0, 1, 4, 10) used in Check (A) and 0 was the orthogonal matrix obtained in (C). The issue was confused slightly by the fact that $0'$ was not quite equal to 0^{-1}, but the accuracy was not greatly reduced. Except in the largest element, where the error was three in the seventh significant figure, the worst error was 2.5 in the fourth decimal place. (See Fig. 12.)

The exponential was computed using the double precision code and generally reduced the errors significantly, but one element, which should have been zero, actually increased in magnitude.

REFERENCES:

[1] Paul M. DeRusso, Rob J. Roy, and Charles M. Close, STATE VARIABLES FOR ENGINEERS, Wiley, 1965.

[2] Richard C. Dorf, TIME DOMAIN ANALYSIS AND DESIGN OF CONTROL SYSTEMS, Addison-Wesley, 1965.

[3] E. A. Coddington and N. Levinson, THEORY OF ORDINARY DIFFERENTIAL EQUATIONS, Chapter 3, McGraw-Hill, 1955.

[4] R. E. Kalman and J. E. Bertram, "Control system analysis and design by the second method of Lyapunov," Trans. ASME, 82 D (1960) (Journal of Basic Engineering) 371-393.

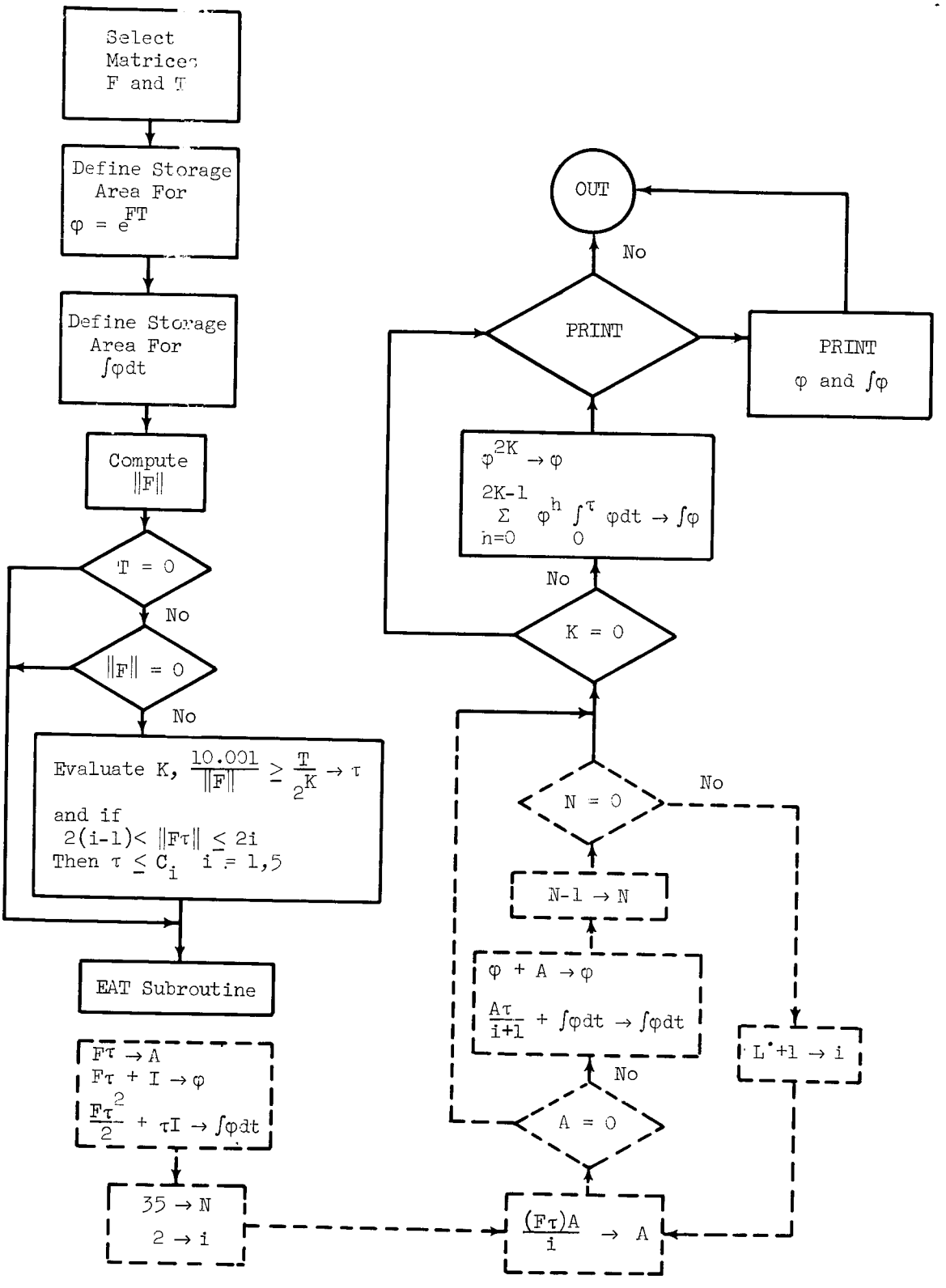
Select Matrices F and T
Define Storage Area For $\varphi = e^{FT}$
Define Storage Area For $\int \varphi dt$
Compute $\|F\|$
T = 0
No
$\|F\| = 0$
No
Evaluate K, $\frac{10.001}{\|F\|} \geq \frac{T}{2^K} \rightarrow \tau$
and if $2(i-1) < \|F\tau\| \leq 2i$ Then $\tau \leq C_i$ i = 1,5
EAT Subroutine
$F\tau \rightarrow A$
$F\tau + I \rightarrow \varphi$
$\frac{F\tau^2}{2} + \tau I \rightarrow \int \varphi dt$
$35 \rightarrow N$
$2 \rightarrow i$
$\frac{(F\tau)A}{i} \rightarrow A$
A = 0
No
$\varphi + A \rightarrow \varphi$
$\frac{A\tau}{i+1} + \int \varphi dt \rightarrow \int \varphi dt$
N-1 → N
N = 0
No
L+1 → i
K = 0
No
$\varphi^{2K} \rightarrow \varphi$
$\sum_{h=0}^{2K-1} \varphi^h \int_0^\tau \varphi dt \rightarrow \int \varphi$
PRINT
No
PRINT φ and $\int \varphi$
OUT

Fig. 1

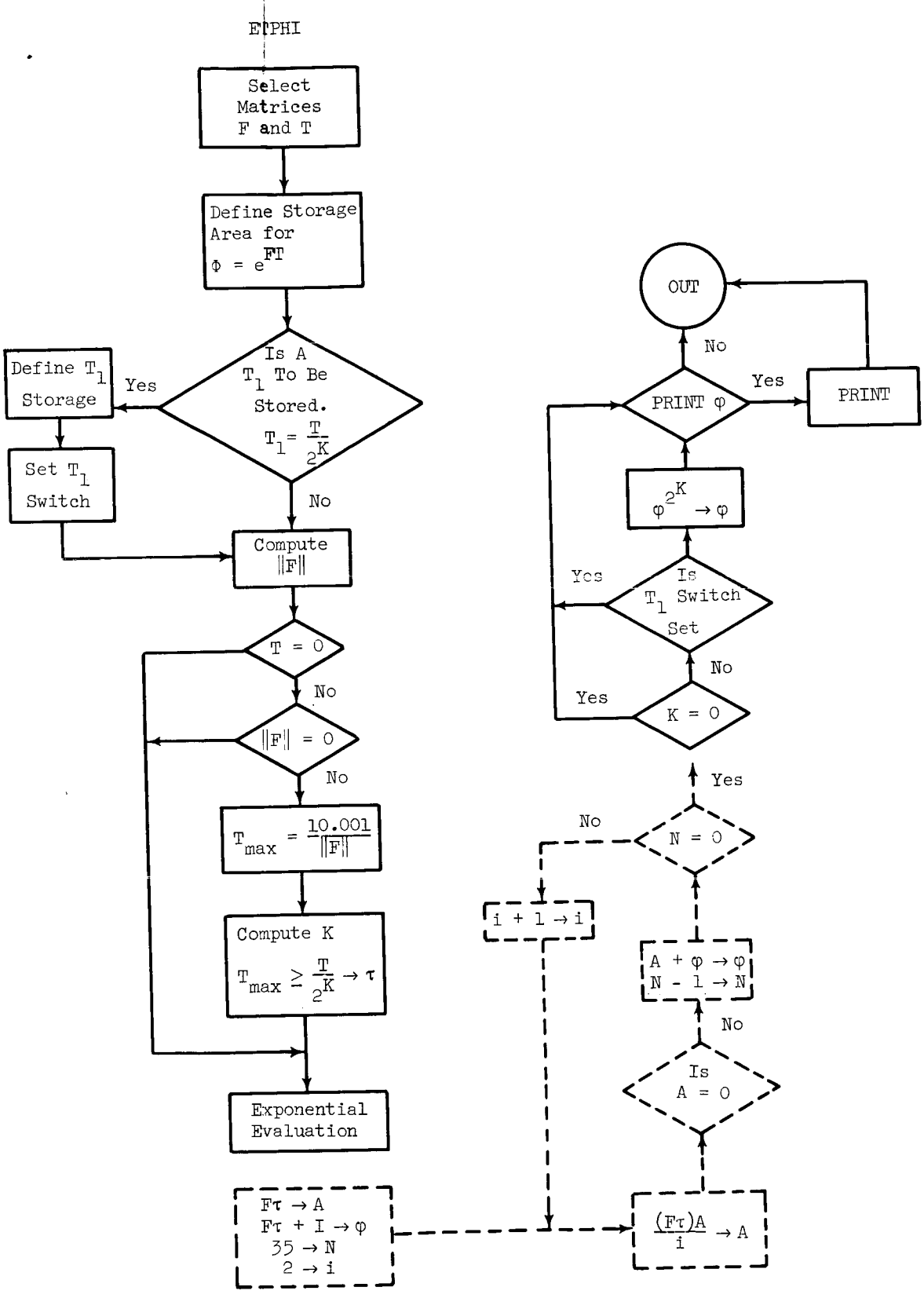
ETPHI
Select Matrices F and T
Define Storage Area for $\Phi = e^{FT}$
Is A T_1 To Be Stored. $T_1 = \frac{T}{2^K}$
Yes
Define T_1 Storage
Set T_1 Switch
No
Compute $\|F\|$
T = 0
No
$\|F\| = 0$
No
$T_{max} = \frac{10.001}{\|F\|}$
Compute K
$T_{max} \geq \frac{T}{2^K} \rightarrow \tau$
Exponential Evaluation
$F\tau \rightarrow A$
$F\tau + I \rightarrow \varphi$
$35 \rightarrow N$
$2 \rightarrow i$
$\frac{(F\tau)A}{i} \rightarrow A$
$i + 1 \rightarrow i$
N = 0
No
Yes
$A + \varphi \rightarrow \varphi$
$N - 1 \rightarrow N$
No
Is A = 0
$\varphi^{2^K} \rightarrow \varphi$
Is T_1 Switch Set
Yes
No
K = 0
Yes
PRINT φ
Yes
PRINT
No
OUT

Fig. 2

MATRIX PHI(0.09999999E 01)

NUMBER OF ROWS 7 NUMBER OF COLUMNS 7

0.53792528E-04	0.	0.	0.	0.	0.
0.					
0.	0.18315654E-01	0.	0.	0.	0.
0.					
0.	0.	0.36787941E-00	0.	0.	0.
0.					
0.	0.	0.	0.09999999E 01	0.	0.
0.					
0.	0.	0.	0.	0.27182817E 01	0.
0.					
0.	0.	0.	0.	0.	0.54598145E 02
0.					
0.	0.	0.	0.	0.	0.
0.22026461E 05					

FIG. 3-A

4.5399930E-05	0	0	0	0	0	0
0	1.8315639E-02	0	0	0	0	0
0	0	3.6787944E-01	0	0	0	0
0	0	0	1.0	0	0	0
0	0	0	0	2.7182818	0	0
0	0	0	0	0	54598150E 01	0
0	0	0	0	0	0	2.2025466E 04

Fig. 3-B

MATRIX INIT(0.09999999E 01)

NUMBER OF ROWS 7 NUMBER OF COLUMNS 7

0.99992417E-01	0.	0.	0.	0.	0.
0.	0.24542104E-00	0.	0.	0.	0.
0.	0.	0.63212050E 00	0.	0.	0.
0.	0.	0.	0.09999999E 01	0.	0.
0.	0.	0.	0.	0.17182817E 01	0.
0.	0.	0.	0.	0.	0.13399536E 02
0.	0.	0.	0.	0.	0.
0.22025461E 04					

FIG. 4-A

9.9995460E-02	0	0	0	0	0	0
0	2.4542109E-01	0	0	0	0	0
0	0	6.3212056E-01	0	0	0	0
0	0	0	1.	0	0	0
0	0	0	0	1.7182818E 00	0	0
0	0	0	0	0	1.3399538E 01	0
0	0	0	0	0	0	2.2024355E 03

Fig. 4-B

MATRIX PHJ(0.09999999E-00)

NUMBER OF ROWS 15 NUMBER OF COLUMNS 15

0.09999999E 01	0.09999999E-00	0.49999999E-02	0.16666666E-03	0.41666664E-05	0.83333328E-07
0.13888887E-08	0.19841268E-10	0.24801584E-12	0.27557316E-14	0.27557316E-16	0.25052105E-18
0.20876753E-20	0.16059041E-22	0.11470743E-24			
0.	0.09999999E 01	0.09999999E-00	0.49999999E-02	0.16666666E-03	0.41666664E-05
0.83333328E-07	0.13888887E-08	0.19841268E-10	0.24801584E-12	0.27557316E-14	0.27557316E-16
0.25052105E-18	0.20876753E-20	0.16059041E-22			
0.	0.	0.09999999E 01	0.09999999E-00	0.49999999E-02	0.16666666E-03
0.41666664E-05	0.83333328E-07	0.13888887E-08	0.19841268E-10	0.24801584E-12	0.27557316E-14
0.27557316E-16	0.25052105E-18	0.20876753E-20			
0.	0.	0.	0.09999999E 01	0.09999999E-00	0.49999999E-02
0.16666666E-03	0.41666664E-05	0.83333328E-07	0.13888887E-08	0.19841268E-10	0.24801584E-12
0.27557316E-14	0.27557316E-16	0.25052105E-18			
0.	0.	0.	0.	0.09999999E 01	0.09999999E-00
0.49999999E-02	0.16666666E-03	0.41666664E-05	0.83333328E-07	0.13888887E-08	0.19841268E-10
0.24801584E-12	0.27557316E-14	0.27557316E-16			
0.	0.	0.	0.	0.	0.09999999E 01
0.09999999E-00	0.49999999E-02	0.16666666E-03	0.41666664E-05	0.83333328E-07	0.13888887E-08
0.19841268E-10	0.24801584E-12	0.27557316E-14			
0.	0.	0.	0.	0.	0.
0.09999999E 01	0.09999999E-00	0.49999999E-02	0.16666666E-03	0.41666664E-05	0.83333328E-07
0.13888887E-08	0.19841268E-10	0.24801584E-12			
0.	0.	0.	0.	0.	0.
0.	0.09999999E 01	0.09999999E-00	0.49999999E-02	0.16666666E-03	0.41666664E-05
0.83333328E-07	0.13888887E-08	0.19841268E-10			
0.	0.	0.	0.	0.	0.
0.	0.	0.09999999E 01	0.09999999E-00	0.49999999E-02	0.16666666E-03
0.41666664E-05	0.83333328E-07	0.13888887E-08			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.09999999E 01	0.09999999E-00	0.49999999E-02
0.16666666E-03	0.41666664E-05	0.83333328E-07			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.09999999E 01	0.09999999E-00
0.49999999E-02	0.16666666E-03	0.41666664E-05			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.09999999E 01
0.09999999E-00	0.49999999E-02	0.16666666E-03			

FIG. 5

```
0.          0.          0.          0.          0.          0.
0.          0.          0.          0.          0.          0.
0.09999999E 01  0.09999999E-00  0.49999999E-02

0.          0.          0.          0.          0.          0.
0.          0.          0.          0.          0.          0.
0.          0.09999999E 01  0.09999999E-00

0.          0.          0.          0.          0.          0.
0.          0.          0.          0.          0.          0.
0.          0.          0.09999999E 01
```

FIG. 5

MATRIX INJ(0.09999999E-00)

NUMBER OF ROWS 15 NUMBER OF COLUMNS 15

0.09999999E-00	0.49999999E-02	0.16666666E-03	0.41666664E-05	0.83333328E-07	0.13888887E-08
0.19841268E-10	0.24801584E-12	0.27557316E-14	0.27557316E-16	0.25052105E-18	0.20876753E-20
0.16059041E-22	0.11470743E-24	0.76471622E-27			
0.	0.09999999E-00	0.49999999E-02	0.16666666E-03	0.41666664E-05	0.83333328E-07
0.13888887E-08	0.19841268E-10	0.24801584E-12	0.27557316E-14	0.27557316E-16	0.25052105E-18
0.20876753E-20	0.16059041E-22	0.11470743E-24			
0.	0.	0.09999999E-00	0.49999997E-02	0.16666666E-03	0.41666664E-05
0.83333328E-07	0.13888887E-08	0.19841268E-10	0.24801584E-12	0.27557316E-14	0.27557316E-16
0.25052105E-18	0.20876753E-20	0.16059041E-22			
0.	0.	0.	0.09999999E-00	0.49999999E-02	0.16666666E-03
0.41666664E-05	0.83333328E-07	0.13888887E-08	0.19841268E-10	0.24801584E-12	0.27557316E-14
0.27557316E-16	0.25052105E-18	0.20876753E-20			
0.	0.	0.	0.	0.09999999E-00	0.49999999E-02
0.16666666E-03	0.41666664E-05	0.83333328E-07	0.13888887E-08	0.19841268E-10	0.24801584E-12
0.27557316E-14	0.27557316E-16	0.25052105E-18			
0.	0.	0.	0.	0.	0.09999999E-00
0.49999999E-02	0.16666666E-03	0.41666664E-05	0.83333328E-07	0.13888887E-08	0.19841268E-10
0.24801584E-12	0.27557316E-14	0.27557316E-16			
0.	0.	0.	0.	0.	0.
0.09999999E-00	0.49999999E-02	0.16666666E-03	0.41666664E-05	0.83333328E-07	0.13888887E-08
0.19841268E-10	0.24801584E-12	0.27557316E-14			
0.	0.	0.	0.	0.	0.
0.	0.09999999E-00	0.49999999E-02	0.16666666E-03	0.41666664E-05	0.83333328E-07
0.13888887E-08	0.19841268E-10	0.24801584E-12			
0.	0.	0.	0.	0.	0.
0.	0.	0.09999999E-00	0.49999999E-02	0.16666666E-03	0.41666664E-05
0.83333328E-07	0.13888887E-08	0.19841268E-10			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.09999999E-00	0.49999999E-02	0.16666666E-03
0.41666664E-05	0.83333328E-07	0.13888887E-08			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.09999999E-00	0.49999999E-02
0.16666666E-03	0.41666664E-05	0.83333328E-07			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.09999999E-00
0.49999999E-02	0.16666666E-03	0.41666664E-05			

FIG. 6

```
 0.              0.              0.              0.              0.              0.
 0.              0.              0.              0.              0.              0.
0.09999999E-00  0.49999999E-02  0.16666666E-03

 0.              0.              0.              0.              0.              0.
 0.              0.              0.              0.              0.              0.
 0.             0.09999999E-00  0.49999999E-02

 0.              0.              0.              0.              0.              0.
 0.              0.              0.              0.              0.              0.
 0.              0.             0.09999999E-00
```

PHJ(T) AND INJ(T) ARE THE EXPONENTIAL AND INTEGRAL EXPONENTIAL OF A JORDAN BLOCK HAVING A ZERO EIGENVALUE, WITH T = .1, 1, AND 10.

FIG. 6

MATRIX PHJ(0.09999999E 01)

NUMBER OF ROWS 15 NUMBER OF COLUMNS 15

0.09999999E 01	0.09999999E 01	0.50000000E 00	0.16666666E-00	0.41666666E-01	0.83333333E-02
0.13888888E-02	0.19841269E-03	0.24801587E-04	0.27557319E-05	0.27557318E-06	0.25052107E-07
0.20876756E-08	0.16059043E-09	0.11470745E-10			
0.	0.09999999E 01	0.09999999E 01	0.50000000E 00	0.16666666E-00	0.41666666E-01
0.83333333E-02	0.13888888E-02	0.19841269E-03	0.24801587E-04	0.27557319E-05	0.27557318E-06
0.25052107E-07	0.20876756E-08	0.16059043E-09			
0.	0.	0.09999999E 01	0.09999999E 01	0.50000000E 00	0.16666666E-00
0.41666666E-01	0.83333333E-02	0.13888888E-02	0.19841269E-03	0.24801587E-04	0.27557319E-05
0.27557318E-06	0.25052107E-07	0.20876756E-08			
0.	0.	0.	0.09999999E 01	0.09999999E 01	0.50000000E 00
0.16666666E-00	0.41666666E-01	0.83333333E-02	0.13888888E-02	0.19841269E-03	0.24801587E-04
0.27557319E-05	0.27557318E-06	0.25052107E-07			
0.	0.	0.	0.	0.09999999E 01	0.09999999E 01
0.50000000E 00	0.16666666E-00	0.41666666E-01	0.83333333E-02	0.13888888E-02	0.19841269E-03
0.24801587E-04	0.27557319E-05	0.27557318E-06			
0.	0.	0.	0.	0.	0.09999999E 01
0.09999999E 01	0.50000000E 00	0.16666666E-00	0.41666666E-01	0.83333333E-02	0.13888888E-02
0.19841269E-03	0.24801587E-04	0.27557319E-05			
0.	0.	0.	0.	0.	0.
0.09999999E 01	0.09999999E 01	0.50000000E 00	0.16666666E-00	0.41666666E-01	0.83333333E-02
0.13888888E-02	0.19841269E-03	0.24801587E-04			
0.	0.	0.	0.	0.	0.
0.	0.09999999E 01	0.09999999E 01	0.50000000E 00	0.16666666E-00	0.41666666E-01
0.83333333E-02	0.13888888E-02	0.19841269E-03			
0.	0.	0.	0.	0.	0.
0.	0.	0.09999999E 01	0.09999999E 01	0.50000000E 00	0.16666666E-00
0.41666666E-01	0.83333333E-02	0.13888888E-02			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.09999999E 01	0.09999999E 01	0.50000000E 00
0.16666666E-00	0.41666666E-01	0.83333333E-02			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.09999999E 01	0.09999999E 01
0.50000000E 00	0.16666666E-00	0.41666666E-01			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.09999999E 01
0.09999999E 01	0.50000000E 00	0.16666666E-00			

Fig..7

```
0.          0.          0.          0.          0.          0.
0.          0.          0.          0.          0.          0.
0.09999999E 01  0.09999999E 01  0.50000000E 00

0.          0.          0.          0.          0.          0.
0.          0.          0.          0.          0.          0.
0.          0.09999999E 01  0.09999999E 01

0.          0.          0.          0.          0.          0.
0.          0.          0.          0.          0.          0.
0.          0.          0.09999999E 01
```

FIG. 7

MATRIX INJT 0.09999999E 01

NUMBER OF ROWS 15 NUMBER OF COLUMNS 15

0.09999999E 01	0.50000000E 00	0.16666666E-00	0.41666656E-01	0.83333331E-02	0.13888888E-02
0.19841269E-03	0.24801587E-04	0.27557318E-05	0.27557318E-06	0.25052107E-07	0.20876756E-08
0.16059043E-09	0.11470745E-10	0.76471633E-12			
0.	0.09999999E 01	0.50000000E 00	0.16666666E-00	0.41666666E-01	0.83333331E-02
0.13888888E-02	0.19841269E-03	0.24801587E-04	0.27557318E-05	0.27557318E-06	0.25052107E-07
0.20876756E-08	0.16059043E-09	0.11470745E-10			
0.	0.	0.09999999E 01	0.50000000E 00	0.16666656E-00	0.41666666E-01
0.83333331E-02	0.13888888E-02	0.19841269E-03	0.24801587E-04	0.27557318E-05	0.27557318E-06
0.25052107E-07	0.20876756E-08	0.16059043E-09			
0.	0.	0.	0.09999999E 01	0.50000000E 00	0.16666666E-00
0.41666666E-01	0.83333331E-02	0.13888888E-02	0.19841269E-03	0.24801587E-04	0.27557318E-05
0.27557318E-06	0.25052107E-07	0.20876756E-08			
0.	0.	0.	0.	0.09999999E 01	0.50000000E 00
0.16666666E-00	0.41666666E-01	0.83333331E-02	0.13888888E-02	0.19841269E-03	0.24801587E-04
0.27557318E-05	0.27557318E-06	0.25052107E-07			
0.	0.	0.	0.	0.	0.09999999E 01
0.50000000E 00	0.16666666E-00	0.41666666E-01	0.83333331E-02	0.13888888E-02	0.19841269E-03
0.24801587E-04	0.27557318E-05	0.27557318E-06			
0.	0.	0.	0.	0.	0.
0.09999999E 01	0.50000000E 00	0.16666666E-00	0.41666656E-01	0.83333331E-02	0.13888888E-02
0.19841269E-03	0.24801587E-04	0.27557318E-05			
0.	0.	0.	0.	0.	0.
0.	0.09999999E 01	0.50000000E 00	0.16666666E-00	0.41666666E-01	0.83333331E-02
0.13888888E-02	0.19841269E-03	0.24801587E-04			
0.	0.	0.	0.	0.	0.
0.	0.	0.09999999E 01	0.50000000E 00	0.16666666E-00	0.41666666E-01
0.83333331E-02	0.13888888E-02	0.19841269E-03			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.09999999E 01	0.50000000E 00	0.16666666E-00
0.41666656E-01	0.83333331E-02	0.13888888E-02			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.09999999E 01	0.50000000E 00
0.16666666E-00	0.41666666E-01	0.83333331E-02			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.09999999E 01
0.50000000E 00	0.16666666E-00	0.41666666E-01			

FIG. 8

```
0.                0.                0.                0.                0.                0.
0.                0.                0.                0.                0.                0.
0.09999999E 01    0.50000000E 00    0.16666666E-00
0.                0.                0.                0.                0.                0.
0.                0.                0.                0.                0.                0.
0.                0.09999999E 01    0.50000000E 00
0.                0.                0.                0.                0.                0.
0.                0.                0.                0.                0.                0.
0.                0.                0.09999999E 01
```

FIG. 8

MATRIX PHJ(0.09999999E 02)

NUMBER OF ROWS 15 NUMBER OF COLUMNS 15

0.09999999E 01	0.09999999E 02	0.50000000E 02	0.16666666E 03	0.41666666E 03	0.83333332E 03
0.13888888E 04	0.19841269E 04	0.24801586E 04	0.27557317E 04	0.27557317E 04	0.25052106E 04
0.20876755E 04	0.16059043E 04	0.11470744E 04			
0.	0.09999999E 01	0.09999999E 02	0.50000000E 02	0.16666666E 03	0.41666666E 03
0.83333332E 03	0.13888888E 04	0.19841269E 04	0.24801586E 04	0.27557317E 04	0.27557317E 04
0.25052106E 04	0.20876755E 04	0.16059043E 04			
0.	0.	0.09999999E 01	0.09999999E 02	0.50000000E 02	0.16666666E 03
0.41666666E 03	0.83333332E 03	0.13888888E 04	0.19841269E 04	0.24801586E 04	0.27557317E 04
0.27557317E 04	0.25052106E 04	0.20876755E 04			
0.	0.	0.	0.09999999E 01	0.09999999E 02	0.50000000E 02
0.16666666E 03	0.41666666E 03	0.83333332E 03	0.13888888E 04	0.19841269E 04	0.24801586E 04
0.27557317E 04	0.27557317E 04	0.25052106E 04			
0.	0.	0.	0.	0.09999999E 01	0.09999999E 02
0.50000000E 02	0.16666666E 03	0.41666666E 03	0.83333332E 03	0.13888888E 04	0.19841269E 04
0.24801586E 04	0.27557317E 04	0.27557317E 04			
0.	0.	0.	0.	0.	0.09999999E 01
0.09999999E 02	0.50000000E 02	0.16666666E 03	0.41666666E 03	0.83333332E 03	0.13888888E 04
0.19841269E 04	0.24801586E 04	0.27557317E 04			
0.	0.	0.	0.	0.	0.
0.09999999E 01	0.09999999E 02	0.50000000E 02	0.16666666E 03	0.41666666E 03	0.83333332E 03
0.13888888E 04	0.19841269E 04	0.24801586E 04			
0.	0.	0.	0.	0.	0.
0.	0.09999999E 01	0.09999999E 02	0.50000000E 02	0.16666666E 03	0.41666666E 03
0.83333332E 03	0.13888888E 04	0.19841269E 04			
0.	0.	0.	0.	0.	0.
0.	0.	0.09999999E 01	0.09999999E 02	0.50000000E 02	0.16666666E 03
0.41666666E 03	0.83333332E 03	0.13888888E 04			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.09999999E 01	0.09999999E 02	0.50000000E 02
0.16666666E 03	0.41666666E 03	0.83333332E 03			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.09999999E 01	0.09999999E 02
0.50000000E 02	0.16666666E 03	0.41666666E 03			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.09999999E 01
0.09999999E 02	0.50000000E 02	0.16666666E 03			

FIG. 9

```
0.          0.          0.          0.          0.          0.
0.          0.          0.          0.          0.          0.
0.09999999E 01  0.09999999E 02  0.50000000E 02

0.          0.          0.          0.          0.          0.
0.          0.          0.          0.          0.          0.
0.          0.09999999E 01  0.09999999E 02

0.          0.          0.          0.          0.          0.
0.          0.          0.          0.          0.          0.
0.          0.          0.09999999E 01
```

FIG. 9

MATRIX INJ(0.09999999E 02)

NUMBER OF ROWS 15 NUMBER OF COLUMNS 15

0.09999999E 02	0.50000000E 02	0.16666666E 03	0.41666666E 03	0.83333333E 03	0.13888888E 04
0.19841269E 04	0.24801586E 04	0.27557317E 04	0.27557317E 04	0.25052106E 04	0.20876755E 04
0.16059043E 04	0.11470744E 04	0.76471530E 03			
0.	0.09999999E 02	0.50000000E 02	0.16666666E 03	0.41666666E 03	0.83333333E 03
0.13888888E 04	0.19841269E 04	0.24801586E 04	0.27557317E 04	0.27557317E 04	0.25052106E 04
0.20876755E 04	0.16059043E 04	0.11470744E 04			
0.	0.	0.09999999E 02	0.50000000E 02	0.16666666E 03	0.41666666E 03
0.83333333E 03	0.13888888E 04	0.19841269E 04	0.24801586E 04	0.27557317E 04	0.27557317E 04
0.25052106E 04	0.20876755E 04	0.16059043E 04			
0.	0.	0.	0.09999999E 02	0.50000000E 02	0.16666666E 03
0.41666666E 03	0.83333333E 03	0.13888888E 04	0.19841269E 04	0.24801586E 04	0.27557317E 04
0.27557317E 04	0.25052106E 04	0.20876755E 04			
0.	0.	0.	0.	0.09999999E 02	0.50000000E 02
0.16666666E 03	0.41666666E 03	0.83333333E 03	0.13888888E 04	0.19841269E 04	0.24801586E 04
0.27557317E 04	0.27557317E 04	0.25052106E 04			
0.	0.	0.	0.	0.	0.09999999E 02
0.50000000E 02	0.16666666E 03	0.41666666E 03	0.83333333E 03	0.13888888E 04	0.19841269E 04
0.24801586E 04	0.27557317E 04	0.27557317E 04			
0.	0.	0.	0.	0.	0.
0.09999999E 02	0.50000000E 02	0.16666666E 03	0.41666666E 03	0.83333333E 03	0.13888888E 04
0.19841269E 04	0.24801586E 04	0.27557317E 04			
0.	0.	0.	0.	0.	0.
0.	0.09999999E 02	0.50000000E 02	0.16666666E 03	0.41666666E 03	0.83333333E 03
0.13888888E 04	0.19841269E 04	0.24801586E 04			
0.	0.	0.	0.	0.	0.
0.	0.	0.09999999E 02	0.50000000E 02	0.16666666E 03	0.41666666E 03
0.83333333E 03	0.13888888E 04	0.19841269E 04			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.09999999E 02	0.50000000E 02	0.16666666E 03
0.41666666E 03	0.83333333E 03	0.13888888E 04			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.09999999E 02	0.50000000E 02
0.16666666E 03	0.41666666E 03	0.83333333E 03			
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.09999999E 02
0.50000000E 02	0.16666666E 03	0.41666666E 03			

FIG. 10

```
0.          0.          0.          0.          0.          0.
0.          0.          0.          0.          0.          0.
0.09999999E 02  0.50000000E 02  0.16666666E 03

0.          0.          0.          0.          0.          0.
0.          0.          0.          0.          0.          0.
0.          0.09999999E 02  0.50000000E 02

0.          0.          0.          0.          0.          0.
0.          0.          0.          0.          0.          0.
0.          0.          0.09999999E 02
```

FIG. 10

MATRIX IDN

NUMBER OF ROWS 7 NUMBER OF COLUMNS 7

0.99999988E 00	0.42359716E-08	0.51181814E-08	-0.29267539E-08	0.41327439E-08	-0.14115357E-08
0.67666405E-09					
0.42359716E-08	0.99999989E 00	-0.17607817E-08	0.58207661E-09	0.28812792E-08	-0.34924597E-09
0.12223609E-08					
0.51181814E-08	-0.17607817E-08	0.99999988E 00	0.87893568E-08	0.45401976E-08	-0.51222742E-08
0.44237822E-08					
-0.29267539E-08	0.58207661E-09	0.87893568E-08	0.99999987E 00	-0.46566129E-09	0.
0.93132257E-09					
0.41327439E-08	0.28812792E-08	0.45401976E-08	-0.46566129E-09	0.99999989E 00	0.11175871E-07
0.37252903E-08					
-0.14115357E-08	-0.34924597E-09	-0.51222742E-08	0.	0.11175871E-07	0.99999988E 00
0.					
0.67666405E-09	0.12223609E-08	0.44237822E-08	0.93132257E-09	0.37252903E-08	0.
0.99999990E 00					

THIS IS THE PRODUCT OF PH AND TRANSPOSE PH, WHERE PH IS THE EXPONENTIAL OF A SKEW SYMMETRIC MATRIX. IDN SHOULD THEREFORE BE THE IDENTITY.

FIG. 11

MATRIX PHI(0.09999999E 01)

NUMBER OF ROWS 7 NUMBER OF COLUMNS 7

0.45293588E-04	0.23617882E-06	-0.40331600E-06	0.84588805E-07	-0.32967597E-06	-0.10405193E-06
0.15132519E-04					
0.33489229E-06	0.18314753E-01	0.14111196E-05	-0.79889878E-06	-0.18739047E-05	0.39375573E-06
-0.41524761E-04					
-0.59310605E-06	0.22310414E-05	0.36787488E-00	0.41776511E-05	-0.99034514E-06	0.31571835E-05
0.14841498E-03					
0.75449861E-07	-0.39555015E-06	-0.24761539E-06	0.10000050E 01	-0.79614110E-05	-0.19413419E-04
-0.36458485E-04					
0.10424555E-05	-0.46470668E-05	0.12644101E-04	-0.16149133E-04	0.27182764E 01	0.30286610E-04
0.14419109E-03					
0.61327592E-06	-0.16689301E-05	0.62286854E-05	-0.15437603E-04	0.26226044E-04	0.54598099E 02
-0.12803078E-03					
0.23484230E-04	-0.67710876E-04	0.19454956E-03	-0.10681152E-03	0.24414063E-03	-0.24414063E-03
0.22026432E 05					

THIS IS THE EXPONENTIAL OF DIAG(-10, -4, -1, 0, 1, 4, 10)
AS COMPUTED BY DOING A SIMILARITY TRANSFORMATION BEFORE AND
AFTER EXPONENTIATION.
FINAL TIME 19.561 DATE 12/01/64

FIG. 12

CHAPTER III

THE TRANSIENT PROGRAM

1. Statement of the Problem. This is a program to display solutions of a linear differential equation.

2. Theory. Having obtained a feedback control law for a system, we usually like to have some means of observing the output. We can do this most readily by computing the response of the system to unit initial conditions on the state variables.

If we have a differential system

$$\begin{aligned} \dot{x} &= Fx + Gu(t) \\ y(t) &= Hx(t) \end{aligned} \tag{2.1}$$

with control law

$$u(t) = - Kx(t)$$

then we can rewrite the system as

$$\begin{aligned} \dot{x} &= (F - GK)x \\ y(t) &= Hx(t) \end{aligned} \tag{2.2}$$

with solution

$$x(t + \tau) = e^{\tau(F - GK)}x(t).$$

On the other hand, we may have a system which is monitored at intervals τ_1, while the control is changed only at sampling intervals τ_2 (τ_2 is some positive integral multiple of τ_1.)

Furthermore the system may have a constant forcing term.

3. Computation. The Transient Routine mechanizes a set of equations which encompass these situations and provide, along with a time history of the solution, a rather detailed description of the process (see Chapter I, Fig. 16). For instance, the times when control is computed are marked with an asterisk. The equations used are:

$$x(t + \tau_1) = Px(t) + Gu(t)$$

$$u(t) = Jr - Kx(t_o + i\tau_2) \quad i\tau_2 \leq t - t_o \leq (i + 1)\tau_2.$$

Thus P is the transition matrix, G is the integral of Green's function, JR represents a forcing function and Kx is the feedback term. The state vector is monitored at intervals τ_1 and the control is constant during the sampling interval τ_2.

In certain applications the state vectors are not the observables and in such cases it is desirable to monitor the actual system output. Furthermore at most seven state variables can be printed. So instead of printing the state variables we print

$$y(t) = Hx(t)$$

wnere H is a constant matrix having at most seven rows.

4. Checks. The Transient Routine is, mathematically, an extremely simple program; but the logic is somewhat complicated. Most of the complications of the routine are involved in setting up the printout with the correct number of state variables, control variables, etc. This makes it impossible to design a single run to check every possibility. Repeated use has given us considerable confidence in its freedom from error. For a sample run see Fig. 16, Chapter I.

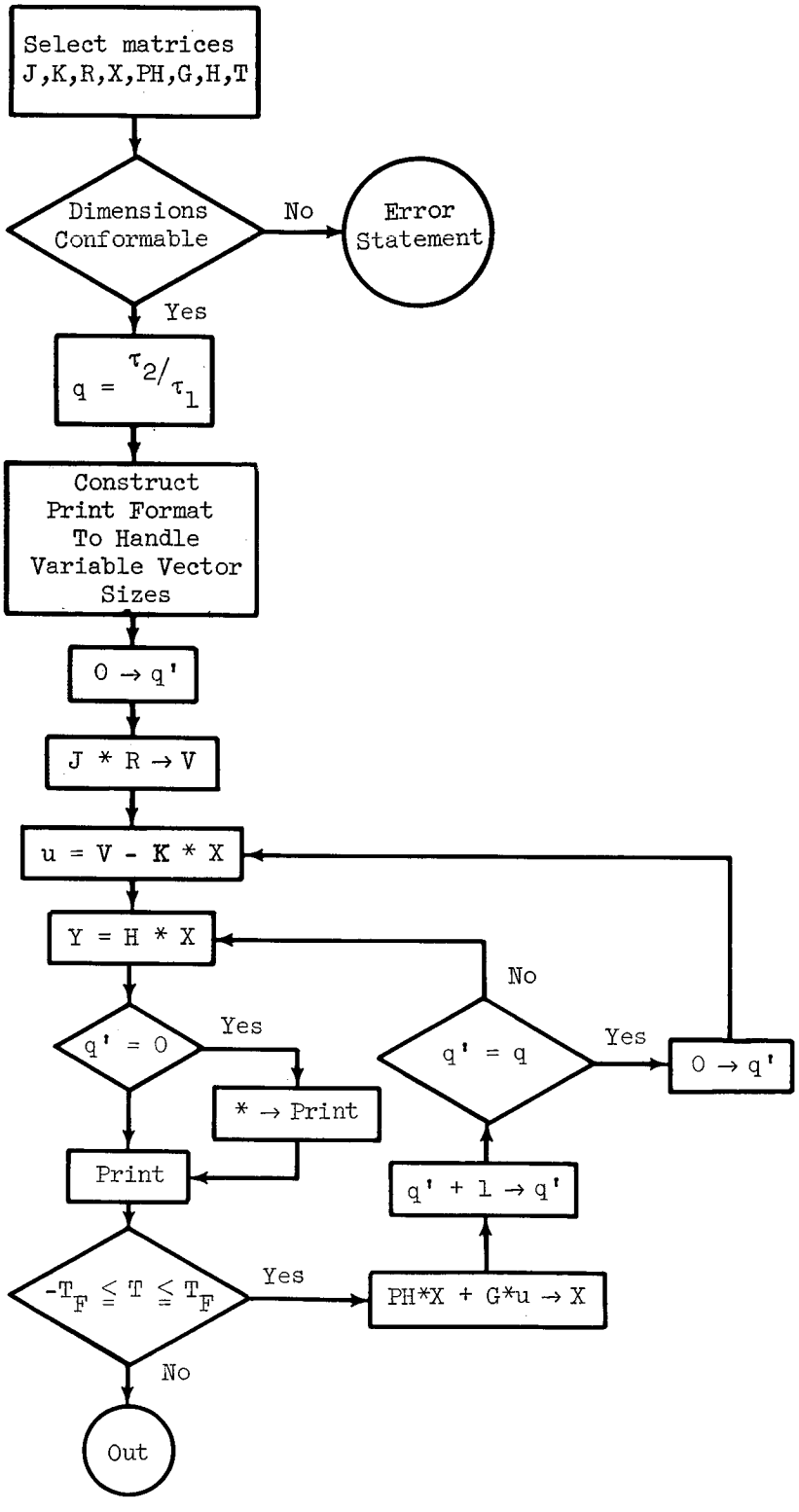
Select matrices J,K,R,X,PH,G,H,T
Dimensions Conformable
No
Error Statement
Yes
q = τ2/τ1
Construct Print Format To Handle Variable Vector Sizes
0 → q'
J * R → V
u = V - K * X
Y = H * X
q' = 0
Yes
* → Print
Print
-TF ≤ T ≤ TF
Yes
PH*X + G*u → X
No
Out
No
q' = q
Yes
0 → q'
q' + 1 → q'

Fig. 1

CHAPTER IV

THE GENERALIZED INVERSE PROGRAM

1. Description of the Problem. Many problems in optimization require computations in linear algebra which go beyond finding an inverse. For instance, determination of rank, least-squares solutions of linear equations, etc. These problems are readily handled via the concept of a pseudo-inverse (or generalized inverse) of a matrix.

2. Theory. In matrix calculus there is a frequently recurring difficulty due to the fact that the inverse of a matrix does not always exist. To prove the existence of the inverse of a given matrix is often cumbersome and difficult. Moreover, in many cases solutions of a set of linear equations exist even when the inverse of the matrix defining these equations does not.

To obviate some of these difficulties, it has been found convenient to make use of the notion of a so-called pseudo-inverse of a matrix. Roughly speaking, a pseudo-inverse must possess two properties to be useful: (i) it must always exist; (ii) when used in place of the inverse (which may not exist), it should give sensible answers to questions such as solutions of equations.

We shall present here a brief discussion of the properties of pseudo-inverses, in particular the "generalized" inverse of Penrose. Further details may be found in Reference 1.

References.

[1] R. Penrose, (1955) "A generalized inverse for matrices", Proc. Cambridge Phil. Soc., 51 406-413.

[2] R. Penrose, (1956) "On best approximate solutions of linear matrix equations", Proc. Cambridge Phil. Soc., 52, 17-19.

[3] R. V. Andrée, (1951) Am. Math. Monthly, 58, 87-92.

[4] T. N. E. Greville, (1959) "The pseudo-inverse of a rectangular or singular matrix and its application to the solution of systems of linear equations", SIAM Review, 1, 38-43.

DEFINITION. Let A denote the Moore-Penrose inverse of A. This is a matrix that satisfies the following axioms:

(i) $AA^{\#}A = A$,

(ii) $A^{\#}AA = A^{\#}$,

(iii) $(A^{\#}A)' = A^{\#}A$,

(iv) $(AA^{\#})' = AA^{\#}$.

It can be shown that $A^{\#}$ always exists and is unique (see Theorems (2.5) and (2.8) below).

The axioms are readily seen to imply that

(2.1) $A^{\#\#} = A$,

(2.2) $A^{\#} = A^{-1}$ if A^{-1} exists,

(2.3) $A'^{\#} = A^{\#'}$.

It is not generally true that $(AB)^{\#} = B^{\#}A^{\#}$.

Furthermore $A^{\#}$ is a discontinuous function of A. For instance, let

$$A = \begin{bmatrix} 1 & a \\ 1 & 1 \end{bmatrix}$$

then

$$A^{\#} = A^{-1} = \begin{bmatrix} \frac{1}{1-a} & \frac{-a}{1-a} \\ \frac{-1}{1-a} & \frac{1}{1-a} \end{bmatrix}$$

for $a \neq 1$, but

$$A^{\#} = \begin{bmatrix} 1/4 & 1/4 \\ 1/4 & 1/4 \end{bmatrix}$$

for $a = 1$. It is not true in any sense that $\lim_{a \to 1} A^{\#}(a) = A^{\#}(1)$.

(2.4) LEMMA. $A(A'QA)^{\#}A'QA = A$ for any nonsingular symmetric matrix Q.

Proof: Let $C = A(A'QA)^{\#}A'QA - A$. Then $C'QC = 0$, by use of axiom (i). This implies $C = 0$.

(2.5) THEOREM. $A^{\#}$ exists.

Proof: (a) If $A = \text{diag}\,(d_1, \ldots, d_n)$, then let

$$a^{\#}_{ii} = d_i^{-1} \quad \text{if} \quad d_i \neq 0$$

$$a^{\#}_{ij} = 0 \quad \text{otherwise.}$$

(b) If $A = A'$ is symmetric, we know that $A = U \Lambda U'$, U orthogonal, Λ diagonal. Then $A^{\#} = U\Lambda^{\#}U'$.

(c) If A is arbitrary, we let

$$A^{\#} = (A'A)^{\#}A' \tag{2.6}$$

or

$$A^{\#} = A'(AA')^{\#}. \tag{2.7}$$

Either of these satisfy axiom (i) by use of Lemma (2.4) and axioms (ii), (iii) and (iv) by use of part (b) of this theorem.

(2.8) THEOREM. $A^{\#}$ is unique.

Proof: Let X and Y be Penrose inverses of A. Then:

$$\begin{aligned} X &= XAX = A'X'X = A'Y'A'X'X \\ &= A'Y'\,XAX = A'Y'X = YAX = YX'A' \\ &= YAYX'A' = YY'A'X'A' = YY'A' = YAY = Y. \end{aligned}$$

Q.E.D.

In particular, equations (2.6) and (2.7) define the same matrix $A^{\#}$.

That hypotheses (i) - (iv) are independent (i.e., necessary for uniqueness), will be shown later.

The following theorem and its corollaries give, in increasingly general form, the main properties of the Penrose inverse. First, however, we need an elementary fact:

(2.9) ORTHOGONAL PROJECTION LEMMA. Given a normed vector space V, a constant vector $b \in V$, and a linear map $A: V \to V$. Then

$$\|Ax^{\circ} - b\| = \min_{x \in V} \|Ax - b\|$$

is equivalent to

$$(Ax, Ax^{\circ} - b) = 0 \quad \text{for all} \quad x \quad \text{in} \quad V.$$

Proof: Assume $(Ax, Ax^{\circ} - b) = 0$ for all x in V. We want to prove that $\|Ax^{\circ} - b\| \leq \|Ay - b\|$ for y in V. Any y in V can be represented as

$$y = x^{\circ} + z, \quad z \in V.$$

Therefore

$$\|Ay - b\|^2 = \|Ax^{\circ} - b + Az\|^2 = \|Ax^{\circ} - b\|^2 + 2(Az, Ax^{\circ} - b) + \|Az\|^2.$$

Since $(Az, Ax^{\circ} - b)$ is zero by hypothesis and $\|Az\|^2 \geq 0$, the desired conclusion follows.

Now assume that $\|Ax^{\circ} - b\| \leq \|Ay - b\|$ for all y in V and show that $(Ax, Ax^{\circ} - b) = 0$ for all x in V. Suppose there is an x such that $(Ax, Ax^{\circ} - b) = \alpha \neq 0$. Let

$$v = \frac{-\alpha x}{\|Ax\|^2}.$$

Then

$$\|Av + Ax^o - b\|^2 - \|Ax^o - b\|^2$$

$$= \|Av\|^2 + 2(Av,\ Ax^o - b) = \frac{\alpha^2}{\|Ax\|^2} - \frac{2\alpha^2}{\|Ax\|^2} = \frac{-\alpha^2}{\|Ax\|^2} < 0.$$

This contradiction proves the lemma.

(2.10) THEOREM (Penrose): Consider the equation $Ax = b$. Let $x^o = A^{\#}b$ and $x \neq x^o$. Then either

(a) $$\|Ax - b\| > \|Ax^o - b\|; \quad \text{or}$$

(b) $$\|Ax - b\| = \|Ax^o - b\| \quad \text{and} \quad \|x\| > \|x^o\|.$$

Furthermore the minimum value of $\|Ax - b\|$ is

(c) $$\|Ax^o - b\|^2 = \|b\|^2 - \|Ax^o\|^2 = \|b\|^2_{I-AA^{\#}}.$$

In words, x^o is the smallest (in the usual norm) vector which gives the least square error.

Proof: (a) We must show first that $(Ax,\ Ax^o - b) = 0$. In fact,

$$(Ax,\ Ax^o - b) = x'A'AA^{\#}b - x'A'b,$$

and by axiom (iv)

$$(Ax,\ Ax^o - b) = x'A'A^{\#\prime}A'b - x'A'b,$$

which is seen to be zero by equation (2.3) and axiom (i).

By Lemma (2.9) this proves

$$\|Ax - b\| \geq \|Ax^{o} - b\|.$$

(b) Suppose $\|Ax - b\| = \|Ax^{o} - b\|$, and let us write $x = x^{o} + v$, where $v \neq 0$. Then $\|Ax - b\|^2 - \|Ax^{o} - b\|^2 = 2(Ax^{o} - b, Av) + \|Av\|^2 = 0$, but it has been shown above that $(Ax^{o} - b, Av) = 0$. Therefore $Av = 0$ so that v is in the kernel of A.

$$\|x^{o}\|^2 = \|x - (x - x^{o})\|^2 = \|x\|^2 + \|x - x^{o}\|^2 - 2(x, x - x^{o})$$

$$= \|x\|^2 + \|v\|^2 - 2v'A^{\#}b - 2v'v$$

$$= \|x\|^2 - \|v\|^2 - 2b'A^{\#'}v.$$

But, by axioms (ii) and (iii)

$$A^{\#'} \equiv A^{\#'}A'A^{\#'}v = A^{\#'}A^{\#}av = 0.$$

Thus

$$\|x\|^2 = \|x^{o}\|^2 \; b \; \|v\|^2 > \|x^{o}\|^2$$

which proves part (b) of the theorem.

(c) By direct computation, we have

$$\|Ax^{o} - b\|^2 = \|(I - AA^{\#})b\|^2 = \|b\|^2 \; (I - AA^{\#})'(I - AA^{\#}).$$

But $(I - AA^{\#})' = I - (AA^{\#})' = I - AA^{\#}$ by axiom (iv). Moreover, $(I - AA^{\#})$ is idempotent by virtue of axiom (i). Hence

$$\|Ax^{o} - b\|^2 = \|b\|^2_{I-AA^{\#}} = \|b\|^2 - \|Ax^{o}\|^2. \qquad \text{Q.E.D.}$$

This theorem could have been stated as a problem in minimizing the quadratic form $\|Ax - b\|^2$. Indeed, we can extend the theorem as follows:

(2.11) COROLLARY. Consider $\|Ax - b\|^2_P$, where P is symmetric, non-negative definite. Let $x^{o} = (A'PA)^{\#}A'Pb$. Again we have alternatives (a) or (b) of Theorem (1.10), while the minimum value of $\|Ax - b\|^2_P$ is

$$\|Ax^{o} - b\|^2_P = \|b\|^2_{P-PA(A'PA)^{\#}A'P}.$$

Proof: Since P is non-negative definite symmetric there exists a symmetric matrix $Q = P^{1/2}$ such that $P = Q^2 = (P^{1/2})^2$. This allows us to write the quadratic form as $\|P^{1/2}Ax - P^{1/2}b\|^2$. Hence in view of the uniqueness theorem (2.9) all we have to show is that

$$(A'P)^{\#}A'P^{1/2} = (P^{1/2}A)^{\#}.$$

This is done by verifying directly that the left-hand side satisfies the axioms (i) - (iv). Q.E.D.

Theorem (2.10) and its corollary have the following simple geometric interpretation: Ax^{o} is the orthogonal projection (relative to P) of b on the range of A. x^{o} itself is the shortest vector whose image under A is this projection.

It is worth noting also the following: If the minimization problem is solved formally by equating derivatives to zero, the resulting formulae are rigorously correct if inverses (which may not exist) are replaced by the Penrose inverses.

A further extension of these results can be made to sums of quadratic forms.

(2.12) COROLLARY. Consider $Q(x) = \sum_{i=1}^{m} \|A_i x - b_i\|_{P_i}^2$. Let $x^o = (\sum_{i=1}^{m} A_i' P_i A_i)^{\#}(\sum_{i=1}^{m} A_i' P_i b_i)$ and $x \neq x^o$. Then either

(a) $Q(x^o) < Q(x)$ or

(b) $Q(x^o) = Q(x)$ and $\|x^o\| < \|x\|$.

Moreover,

(c) $Q(x^o) = \sum_{i=1}^{m} \|b_i\|_{P_i}^2 - \|A_i x^o\|_{P_i}^2$.

Through this proof we assume that the A_i and P_i are $n \times n$ matrices and the b_i are n vectors. There is no restriction since we may assume the addition of zero rows and columns where necessary.

Proof: This may be established directly by defining the mn dimensional vector b, the $mn \times n$ matrix A, and the $mn \times mn$ matrix P:

$$b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}, \quad A = \begin{bmatrix} A_1 \\ \vdots \\ A_m \end{bmatrix}, \quad P = \begin{bmatrix} P_1 & & \\ & \ddots & \\ 0 & & P_m \end{bmatrix}.$$

The the minimization of $Q(x)$ is the same as the minimization of $\|Ax - b\|_P^2$, so that (2.12) follows from (2.11).

(2.13) COROLLARY. Consider the quadratic form $\|Ax - b\|_P^2$. Let

$$-x^o = \{(I - A^{\#}A)[(I - A^{\#}A)R(I - A^{\#}A)]^{\#}(I - A^{\#}A)R - I\}(A'PA)^{\#}A'Pb$$

and $x \neq x^0$. Then either

(a) $$\|Ax - b\|_P > \|Ax^0 - b\|_P \quad \text{or}$$

(b) $$\|Ax - b\|_P = \|Ax^0 - b\|_P \quad \text{and} \quad \|x\|_R > \|x^0\|_R.$$

The minimum value of the form is, of course, still

(c) $$\|Ax^0 - b\|^2_{P - PA(A'PA)^{\#}A'P}.$$

Proof: First we show that the most general vector which will give the minimum value of $\|Ax - b\|_P$ is

(##) $$x^1 = (A'PA)^{\#}A'Pb + (I - A^{\#}A)v$$

where v is any vector. That x^1 gives the minimum value follows from Corollary (2.11) and the fact that (axiom (i)) $A(I - A^{\#}A) = 0$. To show that the form (##) encompasses all vectors which minimize $\|Ax - b\|^2_P$ it is sufficient to note: (i) All vectors y such that $x^0 + y$ minimizes $\|Ax - b\|_P$ are in the kernel of A; this was shown in the proof of Theorem 2.10. (ii) All vectors x in the kernel of A can be represented as $(I - A^{\#}A)x$, a fortiori as $(I - A^{\#}A)v$, where v is an arbitrary vector.

These observations simplify the original problem to minimization of $\|(A'PA)^{\#}A'Pb + (I - A^{\#}A)v\|_R$ with respect to v, which can be done by Corollary (1.11) to give the desired result.

(2.14) LEMMA. (Generalization of Lemma (2.4)). Let $B(t)$ be an $m \times n$ matrix whose elements are continuous in t in the interval $[0, T]$. Then

$$B(t) - B(t)\left[\int_0^T B'(\tau)B(\tau)d\tau\right]^{\#} \int_0^T B'(\tau)B(\tau)d\tau = 0$$

for all t in [0, T].

Proof: Let D(t) denote the matrix difference above. Using axiom (i), we find that

$$\int_0^T D'(t)D(t)dt = 0.$$

Hence

$$\int_0^T \|D(t)x\|^2 dt = 0$$

for all fixed x, which (remembering that D(t) is continuous in t) implies that $D(t) = 0$, $0 \leq t \leq T$.

B(t) may of course have piecewise constant elements and the lemma can be applied to finite sums. Note that the matrices in such a sequence need not have a constant number of rows.

An examination of (2.10) shows that in order to obtain the minimum error property only axioms (i) and (iv) are required. (ii) and (iii) then assure that the norm of x^0 is minimal; in control theory where the norm of x^0 represents the required control energy, this is a desirable property. In general, if one of the axioms is dropped, the pseudo-inverse will not be unique. For instance, let

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

then

$$A_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

satisfies (i), (ii), and (iii), but not (iv).

$$A_2 = \begin{bmatrix} 1 & 1 \\ 1/2 & 1/2 \end{bmatrix}$$

satisfies (i), (ii), and (iv) but not (iii),

$$A_3 = \begin{bmatrix} 1 & -1 \\ 1/2 & 1/2 \end{bmatrix}$$

satisfies (i), (iii), and (iv) but not (ii), and

$$Z = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

satisfies (ii), (iii), and (iv) but not (i), whereas

$$A^{\#} = \begin{bmatrix} 0 & 0 \\ 1/2 & 1/2 \end{bmatrix}$$

is the Penrose inverse.

The following corollary to Theorem (2.10) gives an alternative characterization of the Penrose inverse.

(2.15) Let $A^{\dagger} \neq A^{\#}$ be any matrix satisfying (i). Then $\|A^{\dagger}\| > \|A^{\#}\|$ where $\|M\|$ is defined as $(\text{trace } M'M)^{\frac{1}{2}} = (\sum_{i,j} m_{ij}^2)^{\frac{1}{2}}$.

Proof: To prove this, it suffices to note that the matrix equation

$$AXA = A$$

may be interpreted as a vector equation in the elements of the matrix X. Since pseudo-inverses always exist, this equation always has a solution. By Theorem (2.10), $A^{\#}$ is the "smallest" solution in the sense of the norm defined above.

A further characterization is that given by Greville [4]. Here $A^{\#}$ is defined as the unique matrix satisfying (i) and having its row space and column space the same as A'.

This characterization may be established by noting that $A^{\#}$ and A' have the same kernel.

If $A'x = 0$, then

$$A^{\#}x = A^{\#}AA^{\#}x = A^{\#}A^{\#'}A'x = 0,$$

and if $A^{\#}x = 0$, then

$$A'x = A'A^{\#'}A'x = A'AA^{\#} = 0. \qquad \text{Q.E.D.}$$

One of the most fruitful characterizations of the pseudo-inverse arises from a consideration of its mapping properties. Let A map the space V into the space W. Denote the range of A by R(A), the kernel of A by K(A), the orthogonal complement of K(A) by CK(A), and the orthogonal complement of R(A) by CR(A). Then the requirement that the

error $\|AA^{\#}b - b\|$ is minimum is equivalent to saying that $A^{\#}$ is the inverse of A on R(A) and maps CR(A) into K(A). Clearly then if we want $A^{\#}b$ to be of minimum length, we must have that $A^{\#}$ maps CR(A) into zero.

We obtain the following two formulae which are equivalent to the four axioms:

$$\text{i a)} \qquad A^{\#}AA' = A'$$

$$\text{ii b)} \qquad AA^{\#}A^{\#'} = A^{\#'}.$$

Because we have concentrated so much on the lack of uniqueness of matrices which give the solution of the minimum error problem, we should point out that there is only one such matrix if either A'A or AA' is nonsingular.

We conclude with two formulae for the pseudo-inverse,

(2.16) PROPOSITION. Let $B^{\dagger}$ be any matrix satisfying (i), then $A^{\#} = A'(AA')^{\dagger}A(A'A)^{\dagger}A'$.

This is easily proved using ia) and iia).

The following is an iterative formula; we wish to find $[A\ a]^{\#}$ where a is a vector. Unfortunately we must distinguish two cases. When $a \notin R(A)$, (rank $[A\ a]$ > rank A, $(I - AA^{\#})\ a \neq 0$).

$$(2.17) \qquad [A\ a]^{\#} = \begin{bmatrix} A^{\#}\{I - \dfrac{aa'(I - AA^{\#})}{a'(I - AA^{\#})a}\} \\ \\ \dfrac{a'(I - AA^{\#})}{a'(I - AA^{\#})a} \end{bmatrix}.$$

When $(I - AA^{\#})a = 0$

$$[A\ a] = \begin{bmatrix} A^{\#}\left(I - \dfrac{aa'a^{\#'}A^{\#}}{1 + a'A^{\dagger'}A^{\dagger}a}\right. \\ \\ \dfrac{a'A^{\#'}A^{\#}}{1 + a'A^{\#'}A^{\#}a} \end{bmatrix}$$

We must assume that a is a vector only in the sense that the number of separate cases appears to be the number of possible ranks of $(I - AA^{\#})a$.

If minimum error is the only desideratum and minumum length of the fitting vector can be ignored, (2.17) can always be used with $\dfrac{1}{a'(I - AA^{\#})a}$ set equal to zero when the denominator is zero.

3. Computation: Several proposals for computing $A^{\#}$ have been made in the literature. These algorithms are often very inefficient from the point of view of numerical computation (though they might be useful in getting exact answers in simple cases). For instance, in establishing the existence of the Penrose inverse for a symmetric matrix S we used the existence of a diagonalizing transformation, but this involved finding all the eigenvales of S which is a more difficult problem mathematically then computation of $S^{\#}$. Moreover, the algorithms proposed so far do not simplify when A^{-1} exists, whereas that given here does.

A description and proof of the algorithm follow.

Phase 1. Compute AA' or $A'A$, whichever has smaller dimension. Call the resultant $n \times n$, symmetric, nonnegative definite matrix B. It will suffice to compute $B^{\#}$ because then one can write

$$A^{\#} = (A'A)^{\#}A' \quad \text{or}$$

$$A^{\#} = (A'(AA')^{\#}.$$

Phase 2. Compute a nonsingular matrix T_1 such that $T_1 B T_1' = E$ is a diagonal matrix with elements zero or one.

The matrix T_1 is computed iteratively in at most n steps by a modification of the gaussian elimination procedure.

As a numerical example, we shall carry through the reduction of the matrix.

$$B = \begin{bmatrix} 2 & 1 & -2 & -2 \\ 1 & 25 & -8 & 6 \\ -2 & -8 & 4 & 0 \\ -2 & 6 & 0 & 4 \end{bmatrix}.$$

Let $T_1^{(0)} = I$ and $B^{(0)} = B^{(1)} = B$.

Step 1. Select the largest diagonal element (LDE) of $B^{(1)}$ and divide by its square root the corresponding row and column of $B^{(1)}$ and $T_1^{(0)}$.

In the example, the LDE of $B^{(1)}$ is 25 and occurs in the 2nd row and column. Thus

$$B^{(2)} = \begin{bmatrix} 2 & 1/5 & -2 & -2 \\ 1/5 & 1 & -8/5 & 6/5 \\ -2 & -8/5 & 4 & 0 \\ -2 & 6/5 & 0 & 4 \end{bmatrix} \quad T_1^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Having completed Step 1, we obtain

$$T_1^{(1)} B T_1^{(1)'} = B^{(2)}. \tag{3.1}$$

Step 2. Let k_1 be the row and column in which the LDE of $B^{(1)}$ occurred. Multiply the k_1-th row of $B^{(2)}$ resp. $T_1^{(1)}$ by the ik_1-th element of $B^{(2)}$ and substract the resulting row vector from the i-th row of $B^{(2)}$ resp. $T_1^{(1)}$. Do this for all $i \neq k_1$. Then set all the elements $b^{(2)}_{i_1 j}$ equal to zero, except $b^{(2)}_{k_1 k_1} = 1$.

$$B^{(3)} = \begin{bmatrix} 49/25 & 0 & -42/25 & -56/25 \\ 0 & 1 & 0 & 0 \\ -42/25 & 0 & 36/25 & 48/25 \\ -56/25 & 0 & 48/25 & 64/25 \end{bmatrix}, \quad T_1^{(2)} = \begin{bmatrix} 1 & -1/25 & 0 & 1 \\ 0 & 1/5 & 0 & 0 \\ 0 & 8/25 & 1 & 0 \\ 0 & -6/25 & 0 & 1 \end{bmatrix}.$$

We have now the identity

$$T_1^{(2)} B T_1^{(2)'} = B^{(3)}. \tag{3.2}$$

Step 3. Apply Step 1 to $B^{(3)}$ disregarding in search for the largest diagonal element the row k_1 which occurred in Steps 1-2. Now $k_2 = 4$. The result is:

$$B^{(4)} = \begin{bmatrix} 49/25 & 0 & -42/25 & -7/5 \\ 0 & 1 & 0 & 0 \\ -42/25 & 0 & 36/35 & 6/5 \\ -7/5 & 0 & 6/5 & 1 \end{bmatrix}, \quad T_1^{(3)} = \begin{bmatrix} 1 & -1/25 & 0 & 0 \\ 0 & 1/5 & 0 & 0 \\ 0 & 8/25 & 1 & 0 \\ 0 & -3/20 & & 5/8 \end{bmatrix}.$$

Step 4. Apply Step 2 to $B^{(4)}$. Then

$$B^{(5)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_1(4) = \begin{bmatrix} 1 & -1/4 & 0 & -7/8 \\ 0 & 1/5 & 0 & 0 \\ 0 & 1/2 & 1 & -3/4 \\ 0 & -3/20 & 0 & 5/8 \end{bmatrix}$$

which completes the reduction in the example considered.

At each state of the iteration we have the identity

$$T_1^{(\alpha)} B T_1^{(\alpha)'} = B^{(\alpha+1)}, \quad \alpha = 1, \ldots, 2q \leq 2n. \tag{3.3}$$

If the matrix

$$B^{(2q+1)} = E \tag{3.4}$$

for the first time, then

$$\text{rank } B = a. \tag{3.5}$$

In the example, $q = 2$.

This algorithm is a slight modification of one given by Ref. [3].

Phase 3A. If $B^{(2q+1)} = I$ = unit matrix, then B is invertible and hence

$$B = [T_1^{(2q)}]^{-1} [T_1^{(2q)'}]^{-1}$$
$$B^{\dagger} = B^{-1} = T_1^{(2q)'} T_1^{(2q)}. \tag{3.6}$$

Phase 3B. Suppose $B^{(2q+1)} = E \neq I$. Then there exists a permutation (orthogonal) matrix P such that

$$\bar{E} = PTBT'P' = PEP' = \begin{bmatrix} I_q & 0 \\ 0 & 0 \end{bmatrix}$$

That is, the first q elements on the diagonal are one and all other elements are zero. Note that $P^{-1} = P'$.

Further we have

$$\bar{E} = PTP'PBP'PT'P', \tag{3.7}$$

so that $\bar{T} = PTP'$ is the matrix given by the Andree algorithm for the matrix $\bar{B} = PBP'$.

The method of construction of T given above in the Andrée algorithm assures us that $\bar{T}$ has the form

$$\bar{T} = \begin{bmatrix} \bar{V} & 0 \\ \bar{W} & I_{n-q} \end{bmatrix},$$

where $\bar{V}$ is a nonsingular $q \times q$ matrix.

Partitioning $\bar{B}$ in the same manner we write

$$\bar{B} = \begin{bmatrix} \bar{B}_{11} & \bar{B}_{12} \\ \bar{B}'_{12} & \bar{B}_{22} \end{bmatrix}$$

where $\bar{B}_{11}$ is nonsingular.

Substituting (3.8-9) into the identity $\bar{T}\bar{B}\bar{T}' = \bar{E}$, we obtain the relations

$$\bar{V}\bar{B}_{11}\bar{V}' = I_q,$$

$$\bar{V}\bar{B}_{11}\bar{W}' + \bar{V}\bar{B}_{12} = 0 \tag{3.10}$$

$$(\bar{W}\bar{B}_{11} + \bar{B}_{12})\bar{W}' + \bar{W}\bar{B}_{12} + \bar{B}_{22} = 0.$$

These equalities imply that

$$\bar{B} = \begin{bmatrix} I_q & 0 \\ -\bar{W} & 0 \end{bmatrix} \begin{bmatrix} B_{11} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I_q & -W' \\ 0 & 0 \end{bmatrix}$$

and

$$C = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} I_q & -\bar{W}' \\ 0 & 0 \end{bmatrix} \bar{B} \begin{bmatrix} I_q & 0 \\ -\bar{W} & 0 \end{bmatrix} \tag{3.12}$$

where $D = (I_q + \bar{W}'\bar{W})B_{11}(I_q + \bar{W}'\bar{W})$.

Because the rank of the sum of two nonnegative definite, symmetric matrices cannot be less than the greater of two ranks, $(Iq + \bar{W}'\bar{W})$ and D are of rank q, hence nonsingular.

Then

$$B^{\#} = \begin{bmatrix} I & 0 \\ -\bar{W} & 0 \end{bmatrix} \begin{bmatrix} D^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I & -\bar{W}' \\ 0 & 0 \end{bmatrix} \tag{3.13}$$

as can be verified by checking the four axioms of Penrose.

And the fact that

$$B^{\#} = P'\bar{B}^{\#}P \tag{3.14}$$

follows easily.

The steps followed in the machine are simplified in that the only matrices required are $\bar{W}$ and $\bar{D}$. Moreover the permutation matrix P is not actually computed.

Step 1. Let U be the following matrix

$$\text{If } i \neq j, \quad u_{ij} = \begin{cases} -t_{ij} & \text{if } e_{ii} = 0 \\ 0 & \text{if } e_{ii} = 1 \end{cases}$$

$$u_{ii} = \begin{cases} 0 & \text{if } e_{ii} = 0 \\ 1 & \text{if } e_{ii} = 1. \end{cases}$$

In the notation used previously U can be written as

$$U = P'\bar{U}P = P' \begin{bmatrix} I_q & 0 \\ -\bar{W} & 0 \end{bmatrix} P.$$

In the example

$$U = \begin{bmatrix} 0 & 1/4 & 0 & -7/8 \\ 0 & 1 & 0 & 0 \\ 0 & -1/2 & 0 & 3/4 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Step 2. Compute

$$U'BU = C.$$

The result is

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 35.125 & 0 & -4.5625 \\ 0 & 0 & 0 & 0 \\ 0 & -4.5625 & 0 & 13.90625 \end{bmatrix}$$

which can be written as $P'\bar{C}P$.

After deleting the rows and columns corresponding to $e_{ii} = 0$ we have the same nonsingular $q \times q$ matrix which we have previously called D.

Step 3. Compute D^{-1} by means of the Andrée algorithm. As in Phase 3A, we have

$$D^{-1} = T_2'T_2.$$

Step 4. Compute

$$\bar{B}^{\#} = \bar{U} \begin{bmatrix} D^{-1} & 0 \\ 0 & 0 \end{bmatrix} \bar{U}'.$$

This completes Phase 3B.

Phase 4. We now compute $A^{\#}$ using either $A^{\#} = B^{\#}A'$ or $A^{\#} = A'B^{\#}$.

A flow chart appears as Fig. 1; if a plus sign is used in column 16 of the P S E U O instruction, implying that the argument matrix is already nonnegative definite, then the symmetrization process is not executed.

Remark: This algorithm has two very important features:

a) it takes much less time if A is invertible.

b) $A^{\#}$ is a discontinuous function of A because it depends critically on the rank of A. The Andrée algorithm and a norm error control newly developed allow to some extent control of the "apparent" rank of A. This is done as follows.

The algorithm reduction is carried out until the largest diagonal element of a non-reduced row is less than ϵ, where ϵ is the product of the largest diagonal element and a number P1 of PIZER. (P1 is 10^{-2} unless the PIZER instruction is used.) Then

$$\rho_i = \frac{\|B^{\#}B^{\#} - B\|}{\|B\|} + \frac{\|B^{\#}BB^{\#} - B^{\#}\|}{\|B^{\#}\|} \left(P_2^{\,r_i} \right)$$

is computed and the test $\rho_i \leq \rho_{i-1}$ is made where P_2 is the second number in the PIZER instruction and $i = r_i$ is the rank at the i^{th} iteration. If $\rho_i \leq \rho_{i-1}$ and $r_i < n$, the reduction is carried out on another unreduced row; this process continues until the above ρ condition is met on $r = n$

or a diagonal element is negative or zero. With sufficient knowledge of the matrix B one can choose P_1 so that the ρ computations are kept at a minimum and one can choose P_2 to control the rank.

4. Numerical Checks: The exact computation of the generalized inverse involves a very large amount of work.

A) The generalized inverse of the matrix of the numerical example of Sect. 3 has been computed by hand, using the procedure described in Sect. 3. The result is as follows:

$$10B^{\#} = \begin{bmatrix} 0.5509 & 7061 & -0.0110 & 26088 & -0.4691 & 1021 & -0.6328 & 3101 \\ -0.0110 & 26088 & 0.2973 & 7046 & -0.0755 & 12056 & 0.0975 & 64232 \\ -0.4691 & 1021 & -0.0755 & 12056 & 0.4236 & 6934 & 0.5145 & 5108 \\ -0.6328 & 3101 & 0.0975 & 64232 & 0.5145 & 5108 & 0.7511 & 1093 \end{bmatrix}$$

The machine-computed value of the generalized inverse is:

$$10B^{\#} = \begin{bmatrix} 0.5509 & 7060 & -0.0110 & 26087 & -0.4691 & 1020 & -0.6328 & 3102 \\ -0.0110 & 26087 & 0.2973 & 7044 & -0.0755 & 12051 & 0.0975 & 64228 \\ -0.4691 & 1020 & -0.0755 & 12051 & 0.4236 & 6933 & 0.5145 & 5110 \\ -0.6328 & 3102 & 0.0975 & 64228 & 0.514r & 5110 & 0.7511 & 1099 \end{bmatrix}.$$

This computation used iteration.

B) [illegible] exact generalized inverse of the matrix

$$A = \begin{bmatrix} 4 & -1 & -3 & 2 \\ -2 & 5 & -1 & -3 \\ 2 & 13 & -9 & -5 \end{bmatrix}$$

was computed in [4]. His result is

$$A^{\#} = \frac{1}{6,398} \begin{bmatrix} 608 & -362 & 130 \\ -197 & 212 & 242 \\ -431 & 204 & -250 \\ 324 & -255 & -57 \end{bmatrix}.$$

To 10 decimal places, $A^{\#}$ is given by

$$10^2 A^{\#} = \begin{bmatrix} 9.5029\ 69678 & -5.6580\ 18131 & 2.0318\ 84964 \\ -3.0790\ 87215 & 3.3135\ 35480 & 3.7824\ 32010 \\ -6.7364\ 80150 & 3.1884\ 96405 & -3.9074\ 71085 \\ 5.0640\ 85256 & -3.6730\ 22820 & -0.8909\ 03407 \end{bmatrix}.$$

The rank of A is 2 interated. The machine-computed result is (carrying 8 floating decimal digits):

$$\begin{bmatrix} 9.5029\ 695 & -5.6580\ 181 & 2.0318\ 849 \\ -3.0790\ 870 & 3.3135\ 353 & 3.7824\ 319 \\ -6.7364\ 801 & 3.1884\ 964 & -3.9074\ 711 \\ 5.0640\ 824 & -3.6730\ 227 & -0.8909\ 0338 \end{bmatrix}.$$

The error amounts to only about 2-5 parts of the last deciman digit carried by the machine. This is of course exceptionally good. As a check, one finds that the machine result satisfies (i) to about 1 part in 10^7 and (ii) to better than 5 part in 10^8.

C) In addition to this we wanted a check that would show the differences between P S E U 0, P S E U 0 +, and I N V R S. This required a positive definite symmetric matrix. For this purpose we selected a matrix introduced by H. Rutishauer and appearing in Newman and Todd, "Evaluation of Matrix Inversion Program"s, SIAM JOURNAL December 1958. This is denoted by $A_{12} = BB'$ in the reference, where

$$b_{ij} = \begin{cases} (-1)^j \begin{pmatrix} i-1 \\ j-1 \end{pmatrix} & i \geq j \\ 0 & i < j \end{cases}$$

A_{12}^{-1} is B'B because $B^2 = I$. The conditioning number $P(A_{12}) = \max_{i,j} \left|\frac{\lambda_i}{\lambda_j}\right|$ is asymptotic to $e^{2.77n}$. Hence assuming the asymptotic formula to be correct for n so small, we choose $n = 7$ since this gave $P(A_{12}) \approx 10^8$, which is about the limit that we could expect to handle. We print out A_{12} in Fig. 2, $A_{12}^{-1} = B \geq B$ in Fig. 3, the difference between A_{12}^{-1} and $A_{12}^{\#}$ as obtained by P S E U 0+ in Fig. 4, and the difference between A_{12}^{-1} and $A_{12}^{\#}$ as obtained by P S E U 0 in Fig. 5.

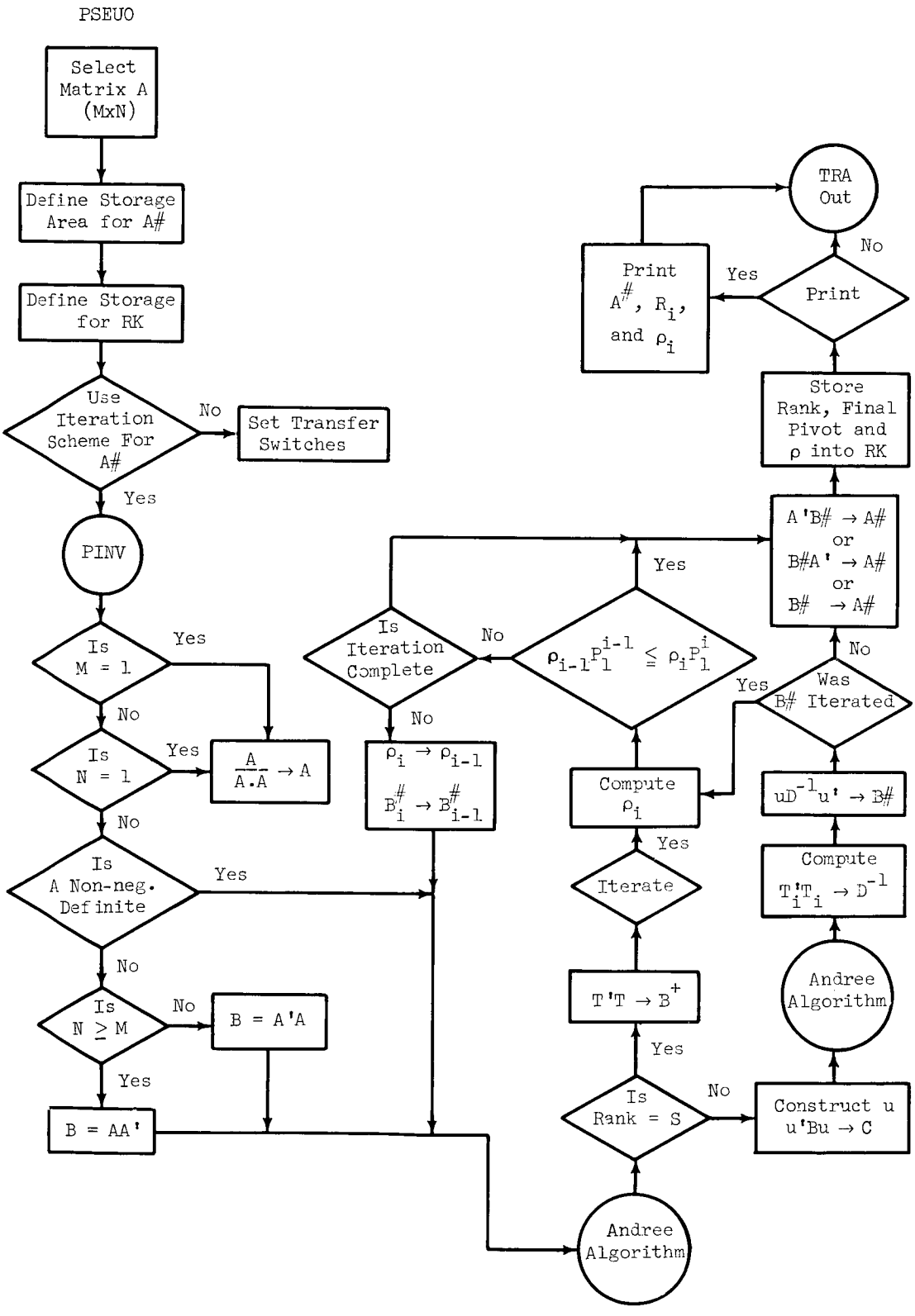
PSEUO
Select Matrix A (MxN)
Define Storage Area for A#
Define Storage for RK
Use Iteration Scheme For A#
No
Set Transfer Switches
Yes
PINV
Is M = 1
Yes
No
Is N = 1
Yes
A / A.A → A
No
Is A Non-neg. Definite
Yes
No
Is N ≥ M
No
B = A'A
Yes
B = AA'
Is Iteration Complete
No
Yes
ρi → ρi-1
B#i → B#i-1
ρi-1 P1 i-1 ≤ ρi P1 i
Yes
Compute ρi
Iterate
Yes
T'T → B+
Yes
Is Rank = S
No
Andree Algorithm
Construct u
u'Bu → C
Andree Algorithm
Compute Ti'Ti → D-1
uD-1u' → B#
Was B# Iterated
No
Yes
A'B# → A#
or
B#A' → A#
or
B# → A#
Store Rank, Final Pivot and ρ into RK
Print
No
Yes
Print A#, Ri, and ρi
TRA Out

Fig. 1

Andree Algorithm

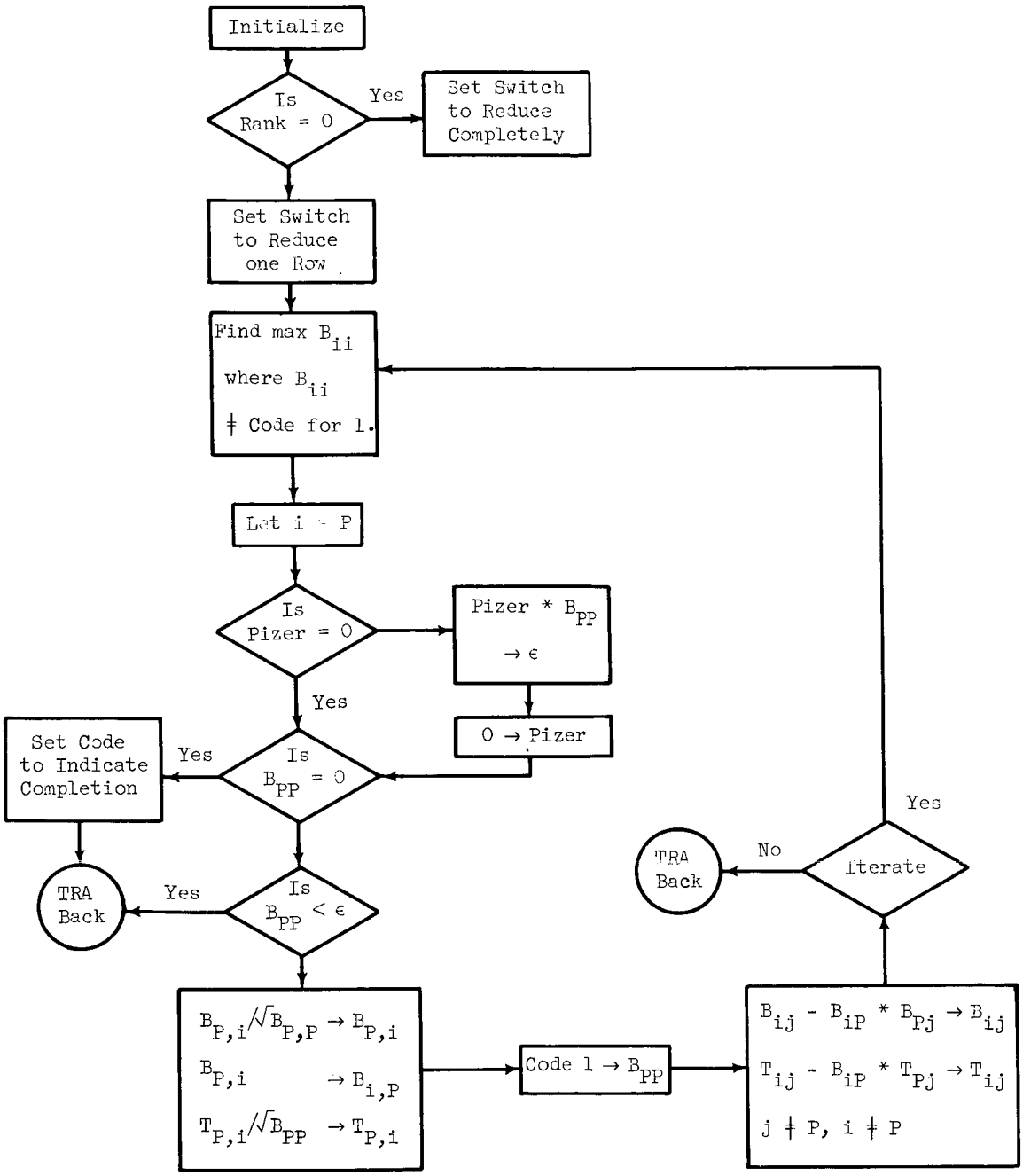

Fig. 1

MATRIX A12

NUMBER OF ROWS 7 NUMBER OF COLUMNS 7

0.09999999E 01	0.09999999E 01	0.09999999E 01	0.09999999E 01	0.09999999E 01	0.09999999E 01
0.09999999E 01					
0.09999999E 01	0.20000000E 01	0.30000000E 01	0.40000000E 01	0.49999999E 01	0.59999999E 01
0.70000000E 01					
0.09999999E 01	0.30000000E 01	0.59999999E 01	0.09999999E 02	0.15000000E 02	0.20999999E 02
0.27999999E 02					
0.09999999E 01	0.40000000E 01	0.09999999E 02	0.20000000E 02	0.34999999E 02	0.55999999E 02
0.84000000E 02					
0.09999999E 01	0.49999999E 01	0.15000000E 02	0.34999999E 02	0.70000000E 02	0.12599999E 03
0.20999999E 03					
0.09999999E 01	0.59999999E 01	0.20999999E 02	0.55999999E 02	0.12599999E 03	0.25200000E 03
0.46200000E 03					
0.09999999E 01	0.70000000E 01	0.27999999E 02	0.84000000E 02	0.20999999E 03	0.46200000E 03
0.92400000E 03					

Fig. 2

MATRIX AIN

NUMBER OF ROWS 7 NUMBER OF COLUMNS 7

0.70000000E 01	-0.20999999E 02	0.34999999E 02	-0.34999999E 02	0.20999999E 02	-0.70000000E 01
0.09999999E 01					
-0.20999999E 02	0.91000000E 02	-0.17500000E 03	0.18900000E 03	-0.11900000E 03	0.41000000E 02
-0.59999999E 01					
0.34999999E 02	-0.17500000E 03	0.37100000E 03	-0.42700000E 03	0.28100000E 03	-0.09999999E 03
0.15000000E 02					
-0.34999999E 02	0.18900000E 03	-0.42700000E 03	0.51700000E 03	-0.35399999E 03	0.13000000E 03
-0.20000000E 02					
0.20999999E 02	-0.11900000E 03	0.28100000E 03	-0.35399999E 03	0.25099999E 03	-0.95000000E 02
0.15000000E 02					
-0.70000000E 01	0.41000000E 02	-0.09999999E 03	0.13000000E 03	-0.95000000E 02	0.37000000E 02
-0.59999999E 01					
0.09999999E 01	-0.59999999E 01	0.15000000E 02	-0.20000000E 02	0.15000000E 02	-0.59999999E 01
0.09999999E 01					

Fig. 3

MATRIX D+

NUMBER OF ROWS 7 NUMBER OF COLUMNS 7

0.11402369E-03	-0.65374374E-03	0.15463829E-02	-0.19392967E-02	0.13680458E-02	-0.51558018E-03
0.81188977E-04					
-0.65374374E-03	0.37345886E-02	-0.88043213E-02	0.11026382E-01	-0.77733994E-02	0.29277802E-02
-0.46092272E-03					
0.15463829E-02	-0.88043213E-02	0.20725250E-01	-0.25943756E-01	0.18283844E-01	-0.68855286E-02
0.10840893E-02					
-0.1939296?E-02	0.11026382E-01	-0.25943756E-01	0.32470703E-01	-0.22876740E-01	0.86174011E-02
-0.13570786E-02					
0.13680458E-02	-0.77733994E-02	0.18283844E-01	-0.22876740E-01	0.16120911E-01	-0.60749054E-02
0.95701218E-03					
-0.51558018E-03	0.29277802E-02	-0.68855286E-02	0.86174011E-02	-0.60749054E-02	0.22888184E-02
-0.36072731E-03					
0.81188977E-04	-0.46092272E-03	0.10840893E-02	-0.13570786E-02	0.95701218E-03	-0.36072731E-03
0.56870282E-04					

Fig. 4

MATRIX D

NUMBER OF ROWS 7 NUMBER OF COLUMNS 7

0.47159510E 01	-0.19483182E 02	0.34996878E 02	-0.34553163E 02	0.19616211E 02	-0.60360374E 01
0.78295618E 00					
-0.19903670E 02	0.89354541E 02	-0.17499496E 03	0.18897811E 03	-0.11775946E 03	0.39934669E 02
-0.57361239E 01					
0.36016928E 02	-0.17604000E 03	0.37099806E 03	-0.42723899E 03	0.28184126E 03	-0.10057884E 03
0.15126776E 02					
-0.35550341E 02	0.18990111E 03	-0.42700113E 03	0.51680888E 03	-0.35467608E 03	0.13071819E 03
-0.20194293E 02					
0.20102470E 02	-0.11790968E 03	0.28100083E 03	-0.35397175E 03	0.25013851E 03	-0.94277938E 02
0.14824928E 02					
-0.61472701E 01	0.39808776E 02	-0.99999923E 02	0.13018229E 03	-0.94096184E 02	0.36054649E 02
-0.57412319E 01					
0.79100037E 00	-0.56917419E 01	0.14999893E 02	-0.20073193E 02	0.14772034E 02	-0.57322235E 01
0.92140007E 00					

FINAL TIME 17.638 DATE 12/07/64

Fig. 5

CHAPTER V

THE DECOMPOSITION ROUTINE

1. Statement of the Problem: It is frequently necessary to determine the rank of a matrix, factor it into two components, etc. This subroutine does such steps, without the necessity of writing a separate program around P S E U O.

2. Theory: In considering systems of the form

$$\dot{x} = Fx + u(t)$$

where $u(.)$ is a random process with a singular covariance matrix W, it is desirable to transform to a representation

$$\dot{x} = Fx + Gv(t)$$

where $v(.)$ has a nonsingular covariance matrix. If we obtain a factorization $W = G'G$ where the number of columns of G is the rank of W, then the new representation is obtained with v having a covariance matrix equal to the identity matrix.

Another application occurs in the process of investigating the equations associated with a given transfer function. It becomes evident that one of the fundamental operations is that of decomposing a matrix of rank r into the sum of r Kronecker products of vectors. That is, if A is an m by n matrix of rank r we would like to find the column vectors $\{h_i\}$ and $\{g_i\}$ such that

$$A = \sum_{i=1}^{r} h_i g_i'.$$

This can also be written as $A = HG$ where the i^{th} column of H is h_i and the i^{th} row of G is g_i'. It is in this form that the decomposition is available in the machine.

As described in Chapter I, the decomposition program at present gives a decomposition only of nonnegative definite symmetric matrices. The output provides matrices S and E where S is non-singular and E has r columns. Then if the input matrix is B, we have $B = SEE'S'$.

But given this we can generate a decomposition for any matrix A. First form AA' or A'A, whichever is smaller. We shall assume here that we have AA'. This being nonnegative definite and symmetric, we can use the D E C OM instruction to obtain S and E. Further we can use the P S E U O instruction to obtain $A^{\#}$. Then $A(A^{\#}A)' = AA^{\#}A$ by axiom (i). However, $A(A^{\#}A)' = AA'A^{\#'} = SEE'S'A^{\#'}$. Therefore if we take $H = SE$ and $G = (A^{\#}SE)'$ we have the required factorization. If we had used A'A to enter D E C O M, then the selection $H = A^{\#'}SE$ $G = E'S'$ would be used.

Notice that there can be no inconsistency about the rank of A as determined by D E C O M provided that if A'A and AA' have the same dimension then AA' is used. The reason for this is that precisely the same operations are performed in both routines. This will be made clearer in Section 3.

3. Computation. Let $B = AA'$ or $A'A$. This program uses Phase 2 of the P S E U O computation. There a non-singular matrix T is computed such that $TBT' = F$ where F is a diagonal matrix with elements 0 or 1. As indicated in Chapter V, the diagonal elements of B in phase 2 are compared with a number ϵ and the algorithm reduction ceases when $B_{ii} < \epsilon$ or rank r of B is maximal. If the code (column 18 is blank) in the D E C O M operation indicates that there is to be no iteration to compute the best decomposition ("best" as defined below), ϵ will be the product of p_1 of P I Z E R and the maximum diagonal element, and the program will exit with the resultant T and F. If colum 18 is a 1, ϵ will be the product of the maximum B_{ii} and p_1 (of the P I Z E R instruction). After B has been reduced as described above

$$\rho_j = \|(T_j^{-1}F_j)(T_j^{-1}F_j)' - B\|p_2^{r_j}$$

is computed and if this is not the first iteration, is compared against ρ_{j-1}.

If $\rho_j \geq \rho_{j-1}$, computation ceases and the (j-1)st T and F are the result. If $\rho_j < \rho_{j-1}$ another row of B is reduced and the comparison made. This process will continue until the pivotal element is non-positive or rank is maximal. An explanation of p_1 and p_2 is in order at this point. p_1 and p_2 are stored as 10^{-2} and 1, but both elements can be changed by the P I Z E R instruction. If p_1 is made smaller most of the reduction will be done the first time through the algorithm and there will be fewer ρ computations thereby saving time. The obvious danger here is that the optimal rank as defined by ρ will be bypassed. p_2 is a method of changing the rank.

Now to conform to the output notation as it appears in Chap. 1, Fig. 1 let $S = T^{-1}$. Then $B = SES'$. But E is idempotent and symmetric so $B = SEE'S' = SE(SE)'$. But SE does not conform to the requirement that the number of columns of SE equal r, the rank of B, except trivially when r is maximal and E is the identity. If rank is not maximal, however, we can delete the zero columns of E and call the resulting matrix E_r without affecting the product

$$B = SE_r(SE_r)'.$$

It is also desirable sometimes to have a matrix P such that

$$PEP' = \begin{bmatrix} I_r & 0 \\ 0 & 0_n \end{bmatrix}.$$

This is the same as having the first r rows of PEP' being linearly independent and also the first r columns.

The output to D E C O M consists of the matrices S, T, E_r, P, E and the rank r of B in that order.

4. Checks: To illustrate DECOM we have used the matrix B appearing in Check A of Chapter 5. In Fig. 2 is SE. In Fig. 3 is SE(SE) and in Fig. 4 is TBT'. These results are quite good.

To illustrate the possibilities of the PIZER instruction we ran this same matrix with $p_1 = 1$ and $p_2 = 10^8$. This gave a rank of one; $p_1 = 1$ let it pivot only once before iteration began, $p_2 = 10^8$ guaranteed that ρ would be larger for rank 2 than for rank one. SE, SE(SE)' and TBT' appear in Fig. 5, 6, and 7.

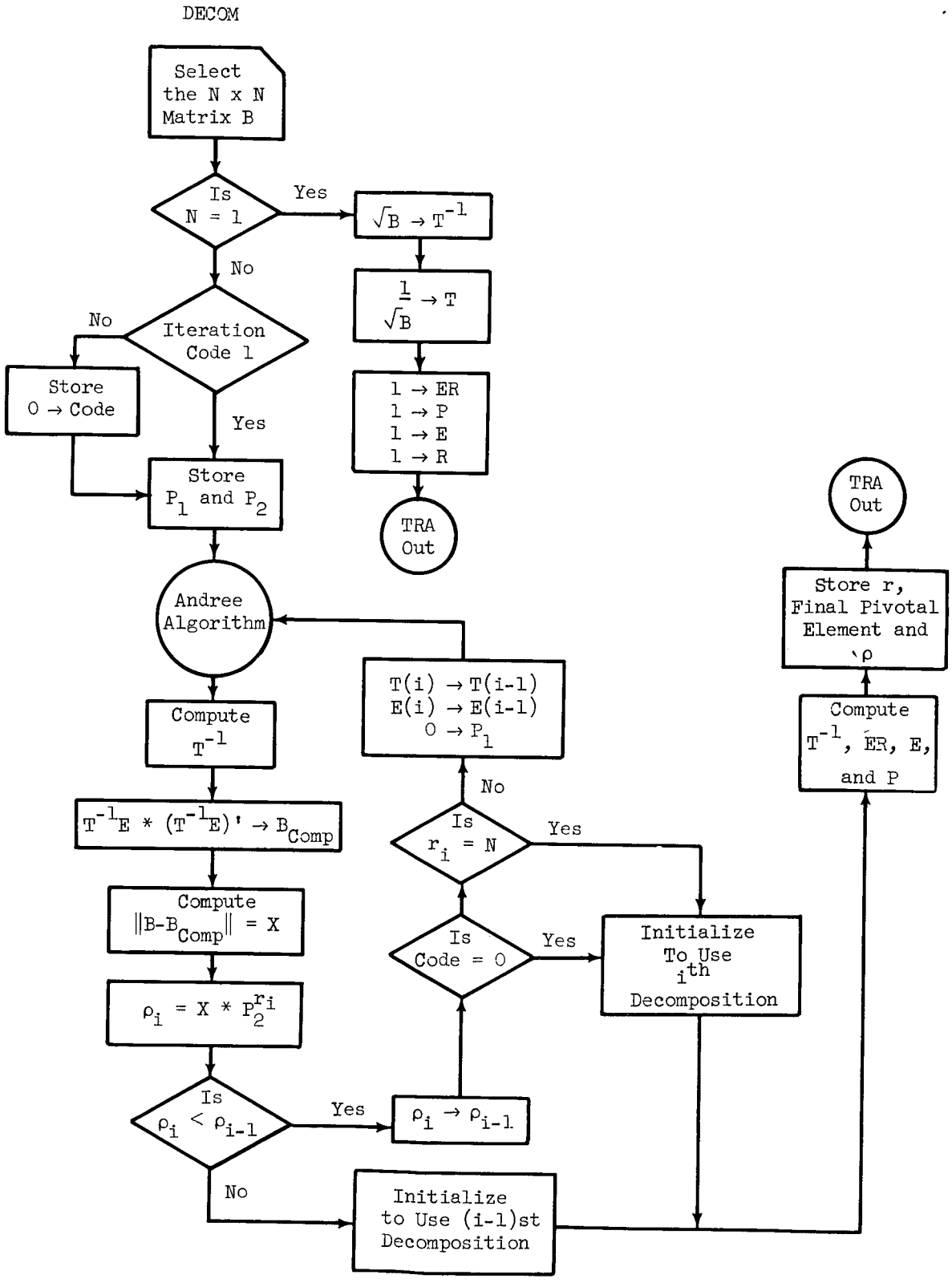
DECOM
Select the N x N Matrix B
Is N = 1
Yes
$\sqrt{B} \rightarrow T^{-1}$
No
$\frac{1}{\sqrt{B}} \rightarrow T$
Iteration Code 1
No
Store 0 → Code
Yes
$1 \rightarrow ER$
$1 \rightarrow P$
$1 \rightarrow E$
$1 \rightarrow R$
Store P_1 and P_2
TRA Out
TRA Out
Andree Algorithm
Store r, Final Pivotal Element and ρ
Compute T^{-1}
$T(i) \rightarrow T(i-1)$
$E(i) \rightarrow E(i-1)$
$0 \rightarrow P_1$
Compute T^{-1}, ER, E, and P
$T^{-1}E * (T^{-1}E)' \rightarrow B_{Comp}$
No
Is $r_i = N$
Yes
Compute $\|B-B_{Comp}\| = X$
Is Code = 0
Yes
Initialize To Use i^{th} Decomposition
$\rho_i = X * P_2^{r_i}$
Is $\rho_i < \rho_{i-1}$
Yes
$\rho_i \rightarrow \rho_{i-1}$
No
Initialize to Use (i-1)st Decomposition

Fig. 1

MATRIX SE

NUMBER OF ROWS 4 NUMBER OF COLUMNS 2

0.20000000E-00	-0.13999999E 01
0.49999999E 01	0.47683716E-07
-0.16000000E 01	0.11999999E 01
0.11999999E 01	0.16000000E 01

Fig. 2

MATRIX MB

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

0.19999998E 01	0.99999993E 00	-0.19999998E 01	-0.19999999E 01
0.99999993E 00	0.24999999E 02	-0.79999997E 01	0.59999998E 01
-0.19999998E 01	-0.79999997E 01	0.39999997E 01	0.
-0.19999999E 01	0.59999998E 01	0.	0.39999999E 01

Fig. 3

MATRIX E

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

-0.74505784E-08	0.11920928E-07	-0.26645353E-14	-0.46193598E-07
0.29802322E-07	0.99999998E 00	-0.67055225E-07	0.74505806E-08
0.26077028E-07	-0.59604644E-07	-0.52154057E-07	0.63329933E-07
-0.29802322E-07	0.59604644E-08	0.29802322E-07	0.99999997E 00

Fig. 4

MATRIX SE

NUMBER OF ROWS 4 NUMBER OF COLUMNS 1

0.20000000E-00

0.49999999E 01

-0.16000000E 01

0.12000000E 01

Fig. 5

MATRIX MB

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

0.39999998E-01	0.99999996E 00	-0.31999999E-00	0.23999999E-00
0.99999996E 00	0.24999999E 02	-0.79999997E 01	0.59999998E 01
-0.31999999E-00	-0.79999997E 01	0.25599999E 01	-0.19199999E 01
0.23999999E-00	0.59999998E 01	-0.19199999E 01	0.14399999E 01

Fig. 6

MATRIX E

NUMBER OF ROWS 4 NUMBER OF COLUMNS 4

0.19599999E 01	0.29802322E-08	-0.16799999E 01	-0.22399999E 01
0.55879354E-08	0.99999998E 00	-0.44703484E-07	0.29802322E-07
-0.16799999E 01	-0.23841857E-07	0.14399999E 01	0.19199999E 01
-0.22399999E 01	0.23841857E-07	0.19199999E 01	0.25600000E 01

FINAL TIME 02.847 DATE 12/11/64

Fig. 7

CHAPTER VI

THE I N V R S PROGRAM

1. Theory. Elementary row operations may be represented as premultiplication by nonsingular matrices, elementary column operations as postmultiplication by nonsingular matrices. Furthermore, the proper matrix may be constructed by performing the desired operation on the identity matrix.

In the André algorithm (see Chapter IV) we can obtain a pseudo inverse of M if we have matrices, R, C such that RMC = E where E is a diagonal matrix containing either zeros or ones. The matrices R and C can be obtained by setting up the equation

$$I\,M\,I = M$$

reducing the right hand M to E by elementary operations and performing the same row operations on the left hand identity and the same column operations on the right hand identity.

However if M is nonsingular, then M can be reduced to an identity by row (or column) operations alone.

It is precisely this procedure which the ASP I N V R S routine (Share subroutine U A I N V 1) uses. There is one interesting aspect, namely that the identity matrix and the matrix to be inverted are both kept in the same storage area, or rather that the matrix being inverted is systematically replaced by its inverse.

2. Computation. Column 1 is searched to find the element m_{p1} which is largest in absolute value. The remaining elements of row p are each divided by m_{p1}; the reciprocal of m_{p1} and the resulting n elements are stored in eraseable storage in the following order:

1) $$\frac{m_{p2}}{m_{p1}}, \ldots, \frac{m_{pn}}{m_{p1}}, \quad \frac{1}{m_{p1}}.$$

The first row of the matrix replaces row p in the matrix, and a record is kept of this replacement.

At this point, the only elements of interest in the original matrix are those in rows and columns two through n. Row 1 (originally row p) has been kept in sotrage and column one, rows two through n will be zero after reduction.

Remembering then that i and j range from two through n, each element m_{ij} of the $(n - 1)x(n - 1)$ submatrix is replaced by

$$m_{ij} - m_{il} \frac{m_{pj}}{m_{pl}}.$$

This reduced m_{ij} is restored in location $m_{i-1,\ j-1}$. This process is carried on in row order and, at the end of each row, the quantity $-\frac{m_{il}}{m_{pl}}$ is computed and stored in the cell that formerly held $m_{i-1,\ n}$.

After this process has been completed for rows two through n, the n elements 1) from eraseable storage are stored in those cells which originally contained the last row.

Now to obtain an idea of what has been done, let us see what the last column of the matrix is now

2) $$\frac{-m_{21}}{m_{p1}}, \ldots, \frac{-m_{11}}{m_{p1}}, \frac{-m_{p+1,\ 1}}{m_{p1}}, \ldots, \frac{1}{m_{p1}}.$$

But this column is, except for order, the pth column of the identity matrix if we performed on it the row operations which reduced the first column of M to all zeros except for a one in row p.

The foregoing process is repeated n times, the only change being that during the initial search for the largest element, the law row is ignored during

the third reduction, etc. Notice that because of the one column shift, the first column contains successively the first, second, etc. columns of the reduced matrix.

Remembering that we worked first on the pth column of the identity matrix, it becomes evident that after all the reductions are completed, the row replacements that were made are examined and the corresponding columns are interchanged. The result is now the inverse of the original matrix.

Except for the destructive computation, this operation is very much like the Andrée algorithm used in P S E U O. There are two very significant differences. One is that P S E U O preserves and takes advantage of the symmetry of the input matrix, hence we might expect to do a little better on nonnegative definite matrices. Secondly, I N V R S uses the largest element in the first column to choose which now shall be used for reduction. In P S E U O, the choice is made from the largest element on the diagonal. Thus I N V R S seems to have a slight advantage. This selection of the largest element to use in reduction is a purely numerical point and probably should be made even more sophisticated. For instance in the check problem, all elements in the first column are one, so I N V R S choose the last one it found. But unfortunately the last row was the worst one to choose as the x_1 equation.

It was actually rather surprising to find that P S E U O did a better job than I N V R S. Probably more investigation should be done, but it appears that P S E U O, with P I Z E R equal to zero, should probably be used to invert nonnegative definite, symmetric matrices.

Care must be taken in using I N V R S because the only control in the program is a test to insure that m_{ij} the new pivot element (largest element in the j^{th} row) is nonzero. If $m_{pj} = 0$ I N V R S will return with an error statement.

3. Checks. This being a Share subroutine, we did not feel that a great amount of checking was necessary. However as a comparison we ran the 7×7 matrix used in check C) for P S E U O. The difference between A_{12}^{-1} and the inverse as computed appears in Fig. 2. Notice that it is element by element, worse than P S E U O + and that the computed inverse was not symmetric.

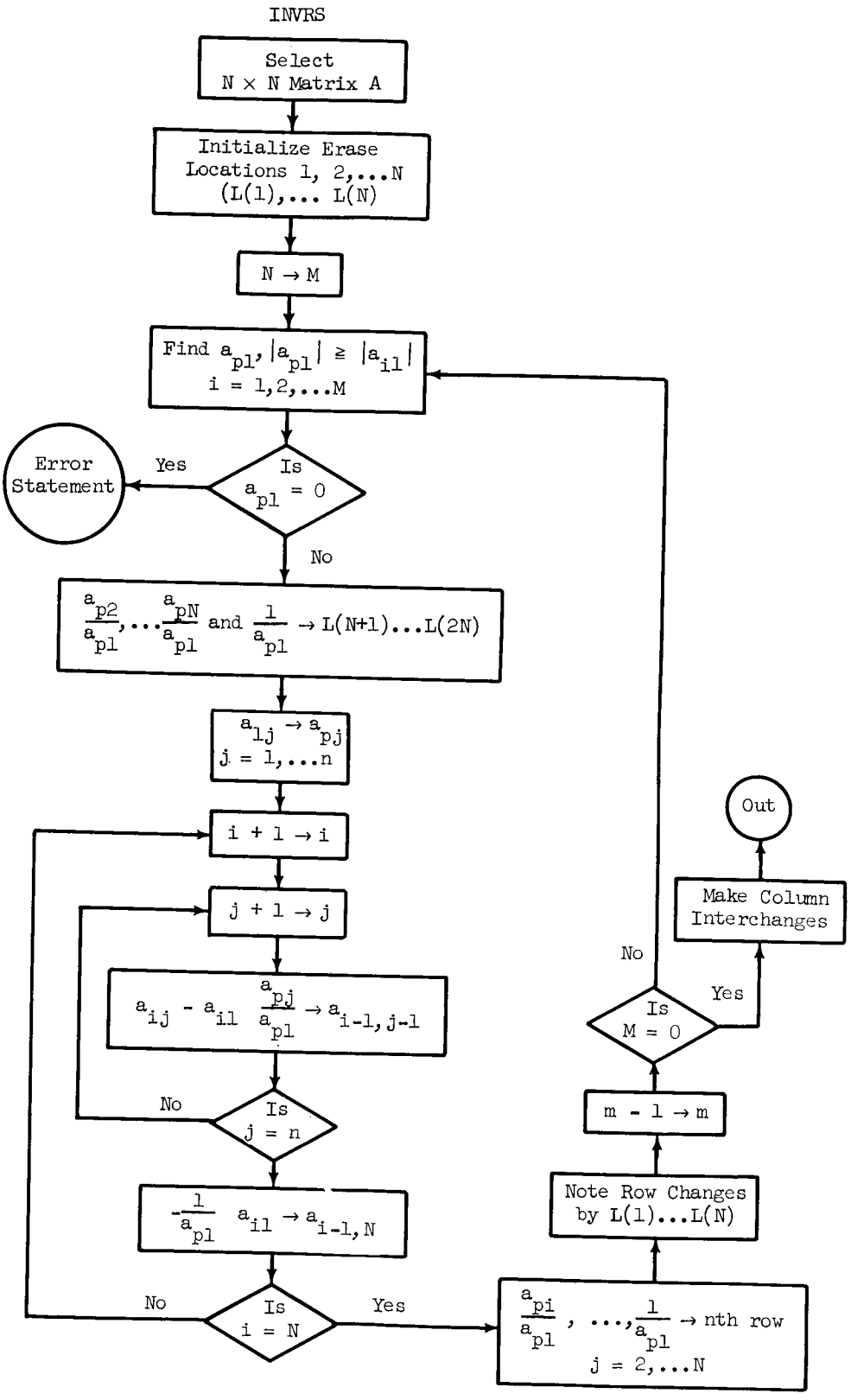
INVRS
Select N × N Matrix A
Initialize Erase Locations 1, 2,...N (L(1),... L(N)
N → M
Find a_{pl}, $|a_{pl}| \geq |a_{il}|$ i = 1,2,...M
Is $a_{pl} = 0$
Yes
Error Statement
No
$\frac{a_{p2}}{a_{pl}}, \ldots \frac{a_{pN}}{a_{pl}}$ and $\frac{1}{a_{pl}} \rightarrow L(N+1) \ldots L(2N)$
$a_{1j} \rightarrow a_{pj}$ j = 1,...n
i + 1 → i
j + 1 → j
$a_{ij} - a_{il} \frac{a_{pj}}{a_{pl}} \rightarrow a_{i-1,j-1}$
Is j = n
No
$-\frac{1}{a_{pl}} a_{il} \rightarrow a_{i-1,N}$
Is i = N
No
Yes
$\frac{a_{pi}}{a_{pl}}, \ldots, \frac{1}{a_{pl}}$ → nth row j = 2,...N
Note Row Changes by L(1)...L(N)
m - 1 → m
Is M = 0
No
Yes
Make Column Interchanges
Out

Fig. 1

MATRIX D

NUMBER OF ROWS 7 NUMBER OF COLUMNS 7

0.52267313E-03	-0.30996799E-02	0.76212883E-02	-0.99525452E-02	0.72836876E-02	-0.28356910E-02
0.45928359E-03					
-0.23298264E-02	0.13795853E-01	-0.33869771E-01	0.44227600E-01	-0.32353401E-01	0.12591839E-01
-0.20386577E-02					
0.48089027E-02	-0.28476715E-01	0.69957733E-01	-0.91320038E-01	0.66822052E-01	-0.26015282E-01
0.42132139E-02					
-0.55088997E-02	0.32627106E-01	-0.80169678E-01	0.10468292E-00	-0.76622009E-01	0.29838562E-01
-0.48334599E-02					
0.36447048E-02	-0.21586418E-01	0.53050995E-01	-0.69282532E-01	0.50722122E-01	-0.19756317E-01
0.32008806E-02					
-0.13113022E-02	0.77672005E-02	-0.19089699E-01	0.24932861E-01	-0.18256187E-01	0.71120262E-02
-0.11523366E-02					
0.19927323E-03	-0.11802912E-02	0.29009581E-02	-0.37891865E-02	0.27747154E-02	-0.10809898E-02
0.17517060E-03					

FINAL TIME 17.078 DATE 12/03/64

Fig. 2.

CHAPTER VII

THE RICCATI EQUATION

1. Description of the Problem: Although ASP is useful for many purposes other than optimization calculations, the majority of the methods are derived from the calculus of variations. We are primarily interested in the so-called "theory of the second variation", which is implemented computationally with the help of the riccati equation. This chapter, much longer than the average, constitutes a short textbook of the second variation together with a detailed treatment of the riccati equation.

2. References: There does not exist at present a complete and modern treatment of the theory of the second variation written primarily from the computing point of view. True, the computing aspect is emphasized in the papers of Arthur Bryson (and also in the forthcoming book by Bryson and Y. C. Ho). But Bryson eschews rigor to such an extent that in his exposition the classical methods seem to be far more limited than they really are.

Almost all practical results of the calculus of variations are to be found in the pages which follow. The general theory (which we discuss only very briefly), is required to justify the passage from "linear" to "local" but plays no direct role in the computations.

Good treatments, showing full awareness of the history of the subject, may be found in

[1] J. Radon, "Zum Problem von Lagrange", Hamburger Matematische Einzelschriften, No. 6, 1928.

[2] C. Carathéodory, VARIATIONSRECHNUNG, Teubner, 1935.

Mathematical research concerning quadratic variational problems is by no means finished yet; in fact, this is a field of applied mathematics where functional analysis can provide new insight as well as practical benefits. See especially

[3] M. R. Hestenes, "Applications of the theory of quadratic forms in Hilbert space to the calculus of variations", Pacific J. Math., 1 (1951) 525-581.

Our treatment here is the continuation of our expository article

[4] R. E. Kalman, "The theory of optimal control and the calculus of variations", Chapter 16, MATHEMATICAL OPTIMIZATION TECHNIQUES, edited by R. Bellman, 1963, Univ. of California Press.

See also

[5] R. E. Kalman, "Contributions to the theory of optimal control", Boletin de la Sociedad Matematica Mexicana, 1960, pp. 102-119

which contains a detailed study of the riccati equation. The modern form of the necessary conditions of Euler and Lagrange, the so-called canonical differential equations of the calculus of variations, is given in the well-known monograph

[6] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, THE MATHEMATICAL THEORY OF OPTIMAL PROCESSES, Interscience, 1962.

[7] I. M. Gel'fand and S. V. Fomin, CALCULUS OF VARIATIONS, Prentice-Hall, 1963.

[8] M. Morse, CALCULUS OF VARIATIONS IN THE LARGE, Am. Math. Soc., 1934.

[9] S. Sternberg, LECTURES ON DIFFERENTIAL GEOMETRY, Prentice-Hall, 1964.

[10] R. E. Kalman, Y. C. Ho, and K. S. Narendra, "Controllability of linear dynamical systems", Contr. to Differential Eqs., 1 (1963) 189-213.

[11] C. Carathéodory, "Über die Einteilung der Variations probleme von Lagrange nack Klassen", Comm. Helv. Mat., 5 (1933) 1-19.

[12] R. E. Kalman, "When is a Linear Control System Optimal?" Journal of Basic Engineering, March 1964.

3. The General Optimization Problem: Consider the (differential or continuous-time) dynamical system.

$$dx/dt = f(t, x, u(t)) \tag{3.1}$$

where

t = time = real number;

x = state = real n-vector; $\{x\} = \mathcal{X}$

$u(.)$ = input or control function = continuous function: $t \to u(t)$ = real m-vector; $\{u(.)\} = \Omega$

f = continuous n-vector function of t, x, u.

The general solution of (3.1) is assumed to exist for all t and is written as

$$x(t) \equiv \varphi(t; \tau, x, u(.)).$$

We wish to choose a control function $u(.)$ for each initial state x in such a way that the functional

$$V(\tau, x, T; u(.)) = \lambda(T, \varphi(T; \tau, x, u(.))) + \int_t^T L(t, \varphi(t; \tau, x, u(.)), u(t))dt \tag{3.2}$$

is minimized over Ω subject to the condition

$$x(\tau) = x,$$

(3.3)
$$\gamma_i(T, \varphi(T; \tau, x, u(x)) = 0 \quad i = 0, \ldots, q \leq n.$$

[The second half of these conditions defines a boundary manifold $\mathcal{B}$, which is a subset of $\{(t, x)\}$. Hence (3.3b) can be written also as $(T, x(T)) \in \mathcal{B}$. If $q = 0$, there are no constraints on $x(T)$. Note that T is implicitly determined by (3.3b).] We require also

(3.4)
$$u(t) \in M, \qquad t \in [\tau, T].$$

We assume that f, λ, L, γ_i are smooth functions. M is often (but by no means always) a convex set and is sometimes also assumed to be compact (to express the fact that the control signals are limited in amplitude).

[It makes little difference theoretically but it greatly simplifies our exposition if $T = \text{const}$. This we shall usually but not always assume. Then γ_i will be independent of T, and therefore $\mathcal{B}$ will be a subset of $\mathcal{X}$.]

4. The Canonical Equations: It is well known that this problem is studied with the help of the hamiltonian

(4.1)
$$H(t, x, y, u(t)) = \eta^* L(t, x, u(t)) + \langle y, f(t, x, u(t)) \rangle,$$

where $\eta^* = \text{const.}$, $y = $ real n-vector, called costate, and $\langle\ ,\ \rangle$ denotes the scalar product.

According to the "minimum principle" of Pontryagin and his collaborators [6], which is none other but the modern version of the classical "multiplier rule", we have the following necessary condition for optimality:

THEOREM. A control function $u^*(\cdot)$ and the corresponding trajectory $x^*(\cdot)$ can be optimal only if there exists a costate function $y^*(\cdot)$ and a constant η^* such that the differential equations

$$\left.\begin{aligned} dx/dt &= H_y(t, x, y, u(t)) \\ dy/dt &= -H_x(t, x, y, u(t)) \end{aligned}\right\}^{(1)} \tag{4.2}$$

are satisfied by $x^*(\cdot)$ and $y^*(\cdot)$ on $t \in [\tau, T]$, $\eta^* = 0$ or 1, and the "minimum condition"

$$H(t, x^*(t), y^*(t), u^*(t)) \leq H(t, x^*(t), y^*(t), u), \quad t \in [\tau, T] \tag{4.3}$$

holds for any $u \in M$.

Moreover, the solutions $x^*(\cdot)$ and $y^*(\cdot)$ of (4.3) must satisfy the boundary conditions

$$\begin{aligned} x(\tau) &= x \\ \langle y, z\rangle &= \eta^*\langle \lambda_x, z\rangle \quad \text{if} \quad z \in T_{\mathcal{B}}(T, x(T)), \end{aligned} \tag{4.4}$$

where $T_{\mathcal{B}}(t, x)$ = tangent manifold of $\mathcal{B}$ at (t, x). Note that

$$z \in T_{\mathcal{B}}(x) \quad \text{iff} \quad \gamma_x^i(x(T))z = 0, \quad i = 0, \ldots, q \leq n.$$

A triple or functions $\{x(\cdot), y(\cdot), u(\cdot)\}$ which satisfies Pontryagin's theorem quoted above is called a pseudo-extremal. We reserve the superscript $*$ for minimizing curves. It is by no means true that every pseudo-extremal corresponds to a minimizing curve.

A pseudo-extremal is called regular if $\eta^* = 1$ and if for each $t \in [\tau, t]$ the hamiltonian $H(t, x(t), y(t), u)$ has a unique (absolute)

(1) H_p, H_x, λ_x, etc. denote partial derivatives $\partial H/\partial p_i$, $\partial \lambda/\partial x_i$, etc.

minimum with respect to $u \in M$ (which then must be necessarily equal to $u(t)$). Thus a regular extremal is specified by the pair $\{x(.), y(.)\}$; because $u(.)$ is implicitly determined by $x(.)$ and $y(.)$ by (4.3) and the requirement of regularity.

A pseudo-extremal is called an extremal if it has the following property: If $t', t'' \in [\tau, T]$ and $|t'' - t'|$ is sufficiently small there does not exist any input $\hat{u}(.) \neq \bar{u}(.)$ such that

$$\text{(i)} \qquad \hat{x}(t'') = \varphi(t''; t', \bar{x}(t'), \hat{u}(.)),$$

$$\text{(ii)} \qquad \int_{t'}^{t''} L(t, \hat{x}(t), \hat{u}(t))dt < \int_{t'}^{t''} L(t, \bar{x}(t), \bar{u}(t))dt.$$

Note that a pseudo-extremal merely satisfies a certain necessary condition (recall that Pontryagin's equations are the generalization of Euler's equations), while an extremal has the additional property that small pieces of it are actually minimizing curves.

The main theorem of the classical calculus of variations asserts: Every regular pseudo-extremal is an extremal.

By Pontryagin's theorem, the control function $u(.)$ is optimal only if there is a pair $\{x(.), y(.)\}$ which, together with $u(.)$, is a pseudo-extremal. $x(.)$ is determined directly from $u(.)$ using (3.1) (or the first canonical equation (4.2a)); $y(.)$ then exists for abstract reasons and is generally not unique. Therefore having passed from $u(.)$ to $\{x(.), y(.), u(.)\}$, it is not obvious whether "regularity" is an intrinsic property of the "independent variable" $u(.)$ or whether it depends also on the choice of $y(.)$. It can be shown, however, that regularity is well defined independently of the possible nonuniqueness of $y(.)$.

It is not hard to prove [4] that regular pseudo-extremals satisfy the free canonical differential equations

$$\text{(4.5)} \qquad \begin{cases} dx/dt = H_y^o(t, x, y), \\ dy/dt = H_x^o(t, x, y), \end{cases}$$

together with the boundary conditions (4.4), where we have set

$$H^{o}(t, x, y) = \min_{u \in m} L(t, x, u)$$

$$= \min_{u \in M} L(t, x, u) + \langle y, f(t, x, u) \rangle. \tag{4.6}$$

This result is true if we make suitable smoothness assumptions concerning H and the terminal set $\mathcal{B}$. Note that by regularity we have set $\eta^* = 1$.

Equations (4.5) allow us to determine each regular pseudo-extremal via the solution of a two-point boundary-value problem involving ordinary differential equations. Thus the problem is reduced to examining all solution of (4.5) which satisfy the appropriate boundary conditions and then picking out those solutions (if there is one) which are actually optimal. Using the canonical equations, we can get only necessary conditions; sufficiency can be proved by adducing additional arguments.

5. The hamilton-jacobi equation: Under suitable smoothness assumptions, the family of all optimal trajectories can be determined at the same time by solving the hamilton-jacobi partial differential equation

$$V_t^o + H^o(t, x, V_x^o) = 0 \tag{5.1}$$

subject to the initial condition

$$V^o(t, x) = \lambda(t, x) \quad \text{for} \quad (t, x) \in \mathcal{B}. \tag{5.2}$$

Note the substitution of V_x^o for y in H^o. It follows that

$$V^o(\tau, x) = \min_{u \in \Omega} [\lambda(x(T)) + \int_{\tau}^{T} L(t, x(t), u(t))dt], \tag{5.3}$$

$$(T, x(T)) \in \mathcal{B}.$$

The existence of a function satisfying (5.1-5.2) is a sufficient (but in general not necessary) condition for the existence of optimal trajectories. Moreover, our notations imply that H has a unique minimum with respect to u, from which it follows that if V^o given by (5.3) satisfies (5.2), trajectory $x(.)$ starting at $x(\tau) = x$ is unique.

By slight abuse of language, let us henceforth call a regular pseudo-extremal simply an extremal. (We shall be concerned solely with regular problems, that is, the case where all pseudo-extremals are not regular and hence, by the main theorem of the classical calculus of variations, extremals.)

6. The Accessory Variational Problem. Let $\{\bar{x}(.), y(.)\}$ be an extremal, i.e. a pair of functions satisfying the canonical differential equations (4.5) with arbritary initial values $\bar{x}_o$, $\bar{y}_o$. One can show by standard arguments in the theory of differential equations see, e.g., [2]) that

$$x(t) - \bar{x}(t) = \xi(t) + o(\|x_o - \bar{x}_o\| + \|y_o - \bar{y}_o\|)$$

and

$$y(t) - \bar{y}(t) = \eta(t) + o(\|x_o - \bar{x}_o\| + \|y_o - \bar{y}_o\|),$$

where the functions $\xi(.)$ and $\eta(.)$ are determined by

$$(6.1) \qquad \begin{cases} d\xi/dt = \bar{H}^o_{yx}(t)\xi + \bar{H}^o_{yy}(t)\eta, \\ d\eta/dt = \bar{H}^o_{xx}(t)\xi - \bar{H}^o_{xy}(t)\eta, \end{cases}$$

with initial conditions

$$\xi_o = x_o - \bar{x}_o, \quad \eta_o = y_o - \bar{y}_o.$$

The $n \times n$ matrix functions $\bar{H}^o_{yx}(.)$, ... are second partial derivatives evaluated along the fixed extremal $\{\bar{x}(.), \bar{y}(.)\}$. We assume of course that F, L, and λ are so smooth that H^o is at least twice continuously differentiable.

Let us introduce the abbreviations:

$$
\begin{aligned}
\bar{F}(t) &= \bar{H}^{o}_{yx}(t) \qquad (\text{implies } \bar{H}^{o}_{xy}(t) = \bar{F}'(t)) \\
\bar{Q}(t) &= \bar{H}^{o}_{xx}(t) \\
\bar{S}(t) &= \lambda_{xx}(t) \\
\bar{G}(t)\bar{G}'(t) &= \bar{H}^{o}_{yy}(t).
\end{aligned}
\tag{6.2}
$$

The last step is legitimate only if we can show that $\bar{H}^{o}_{yy}(t)$ is nonpositive definite. This follows from minimality. (Note that G is not determined uniquely by (6.2), but this fact will be of no interest for us.) The proof is as follows:

Let $c(t, x, y)$ be the argument at the absolute minimum of H with respect to $u \in M$ for given t, x, y. Then

$$H^{o}(t, x, y) = H(t, x, y, c(t, x, y)).$$

We fix t and write H^{o} for $H^{o}(t, \bar{x}(t), \bar{y}(t))$, $\bar{x}$ for $\bar{x}(t)$, $\bar{c}$ for $c(t, \bar{x}(t), \bar{y}(t))$, etc.; y will be a free variable. We regard $H^{o}(t, \bar{x}, y)$ as a function of y and expand it by Taylor's theorem around $y = \bar{y}$. Thus for any y we have the identity

$$H^{o}(t, \bar{x}, y) = \bar{H}^{o} + \langle \bar{H}^{o}_{y}, y - \bar{y} \rangle + \theta \|y - \bar{y}\|^{2}_{\bar{H}^{o}_{yy}} \; *$$

where $0 \leq \theta \leq 1$ is a scalar dependent on y. Since

$$\bar{H}^{o}_{y} = f(t, \bar{x}, \bar{c}),$$

it follows that

$$\theta(y)\|y - \bar{y}\|^{2}_{\bar{H}^{o}_{yy}} = H^{o}(t, \bar{x}, y) - H(t, \bar{x}, y, \bar{c}) \leq 0$$

* The notation $\|x\|^{2}_{A}$ means: quadratic form with respect to the symmetric matrix A.

by the definition of H^o. Hence $\bar{H}^o_{yy}$ is nonpositive definite. (If $\theta(y) = 0$, then $H^o(t, \bar{x}, \cdot)$ is linear and so $\|y - \bar{y}\|^2_{\bar{H}^o_{yy}} = 0$.) Note the similarity of our proof to the classical arguments associated with the weierstrass "E-function".

Equations (6.1) characterize the situation near an extremal. We shall now show that these linearized equations may be regarded also as the canonical equations of a certain variational problem related to the original one. Thus each extremal $\{\bar{x}(.), \bar{y}(.))\}$ will give rise to a so-called accessory variational problem. To define this accessory problem, we must say what f, L, λ are and then we must specify appropriate boundary conditions.

We will use the hat to denote the functions defining the accessory problem, while the Greek letters ξ, η, μ will correspond to x, y, and u in the original variational problem. We introduce the following definitions

$$(6.3) \qquad \begin{cases} \hat{f}(t, \xi, \mu(t)) = \bar{F}(t)\xi + \bar{G}(t)\mu(t) \\ 2\hat{L}(t, \xi, \mu) = \|\xi\|^2_{\bar{Q}(t)} + \|\mu\|^2. \\ 2\lambda(t, \xi) = \|\xi\|^2_{\bar{S}(t)}. \end{cases}$$

Then we find that

$$(6.4) \qquad 2\hat{H}(t, \xi, \eta, \mu) = \|\xi\|^2_{\bar{Q}(t)} + 2\langle\eta, \bar{F}(t)\xi + \bar{G}(t)\mu\rangle + \|\mu\|^2$$

and

$$(6.5) \qquad 2\hat{H}^o(t, \xi, \eta) = \|\xi\|^2_{\bar{Q}(t)} + 2\langle\eta, \bar{F}(t)\xi\rangle - \|\bar{G}'(t)\eta\|^2.$$

It is clear that the accessory variational problem is always regular. (In other words, pseudo-extremals of the original variational problem which are near a regular pseudo-extremal are regular in the first approximation.

Of course, when the given pseudo-extremal of the original variational problem is not regular, we cannot define an accessory problem in the manner indicated because then equations (6.1) have no meaning.)

Using (4.5) we can write down the canonical equations of the accessory extremals:

$$
\begin{aligned}
d\xi/dt &= \hat{H}^{o}_{\eta} = \bar{F}(t)\xi - \bar{G}(t)\bar{G}'(t)\eta \\
d\eta/dt &= \hat{H}^{o}_{\xi} = -\bar{Q}(t)\xi - \bar{F}'(t)\eta.
\end{aligned}
\tag{6.6}
$$

Recalling our abbreviations (6.2), we see that these equations are the same as (6.1). Hence we have a fundamental result which is the basis of the entire theory of the so-called second variation:

THEOREM. The extremals near a (regular) extremal are obtained in the first approximation by solving the accessory variational problem defined by (6.1), (6.2), and (6.3).

Let us now specify the boundary conditions for the accessory problem in the case when T = fixed. The terminal point must belong to the tangent manifold of B, hence

$$
\bar{A}(T)\xi(T) = 0 \tag{6.7}
$$

where

$$
\bar{A}(T) = \Gamma(\bar{x}(T)), \quad \Gamma = \begin{pmatrix} \gamma^{1}_{x} \\ \cdot \\ \cdot \\ \cdot \\ \gamma^{2}_{x} \end{pmatrix}. \tag{6.8}
$$

(By convention, $\bar{A}(T) = 0$ if $q = 0$, i.e., there is no constraint on $x(T)$.) On the other hand, (4.4) gives the constraint

$$\langle \eta(T),\ z\rangle = \langle \bar{S}(T)\xi(T),\ z\rangle$$

whenever $\bar{A}(T)z = 0$; otherwise $\eta(T)$ is a free variable.

7. Digression on the Second Variation: Let us give now a derivation of the accessory variational problem, reminiscent of the way it was done in the nineteenth century. The idea is to expand the functional $V(\tau, x, T; \cdot)$ defined by (3.2) into a generalized Taylor series near a given fixed value $\bar{u}(\cdot)$ of the function $u(\cdot)$. Thus one introduces the "first variation"

$$\delta u(\cdot) = u(\cdot) - \bar{u}(\cdot),$$

and second variation $\delta^2 u(\cdot)$, etc. The term the "second variation" associated with $\delta^2 u(\cdot)$, turn out to be essentially the same as the V function associated with (6.3).

Although these imprecise ideas can be given meaning using functional analysis (see particularly [7], where a clear discussion of "strong" and "weak" first variation may be found), this is not especially fruitful from a practical point of view because it requires the apparatus of infinite-dimensional analysis, whereas the main advantage of the classical theory lies in its ability to treat finite-dimensional problems by fairly well-known methods. Since there is monstrous confusion in the engineering literature concerning the use of the purely symbolic first variation $\delta u(\cdot)$ (see Bryson, loc. cit. in Sec. 2 for a careful treatment), it is of some interest to give here a rigorous derivation of the "second variation" using only finite-dimensional analysis. It is emphasized that this derivation is included mainly for cultural reasons, since the analysis of the previous section has already shown why the accessory variational problem is important in the study of local properties of extremals.

We follow Carathéodory's treatment [2, §315-316].

In order to calculate rigorously the derivative of V with respect to $u(.)$, we assume that instead of infinitely many independent variables, namely $u(.)$, V depends only on the n-vector ρ. In other words, the family of control functions $\{u(.)\} = \Omega$ admitted into competition for the minimum of V will be an n-parameter family, indexed by the n-vector ρ.

This is a very great simplification of the original variational problem; it is justified as follows. By regularity, our extremals $\{x(.), y(.)\}$ are at most of a $2n$-dimensional family since they must satisfy the canonical equations (4.5) and are therefore uniquely determined by $2n$ initial (or end) conditions in (4.5). But the transversality conditions (4.4) impose precisely n constraints, so that the extremals may be indexed by precisely ∞^n numbers, hence by the n-vector ρ.

To show how ρ is actually determined, suppose the terminal surface $\mathcal{B}$ has $r \leq n$ dimensions ($\mathcal{B}$ is embedded in the $(n+1)$-dimensional space $X \times$ time). Since $\mathcal{B}$ is smooth, it can be parametrized by an r-vector μ. The tangent space $T_{\mathcal{B}}$ at $(T, x(T))$ is a linear space of dimension r; it follows from the transversality conditions that the vector $y(T)$ is the sum of $\lambda_x(\mu)$ and a vector y_{free}, which lies on a linear manifold of dimension $n-r$. We write $y_{free} = y(\nu)$, where ν is an $(n-r)$-vector. Then

$$\rho = (\mu, \nu). \tag{7.1}$$

(Consult [8, Chapter 2] or [9, Chapter 6] for a rigorous description of the transversality conditions for parametrized terminal surfaces. Unfortunately both authors restrict themselves to the ordinary ($\dot{x} = u(t)$) variational problem.)

$$\varphi(. \; ; T, x(T), y(T))$$

and

$$\psi(. \; ; T, x(T), y(T))$$

are the solution functions of (4.5), we may consider both φ and ψ to depend on the n-vector parameter ρ by setting

$$T = T(\mu), \quad x(T) = x(\mu), \quad y(T) = \lambda_x(\mu) + y(\nu).$$

We consider $x(t) = \varphi(t;\ \rho)$, $y(t) = \psi(t;\ \rho)$ $u(t) = c(t,\ \varphi(t;\ \rho),\ \psi(t;\ \rho)) = \omega(t,\ \rho)$; after substitution into L, H, λ, F, etc., everything becomes a function of t and ρ only. Thus, with a slight abuse of notation

$$L(t,\ \rho) = H^{\circ}(t,\ \varphi,\ \psi) - \langle \psi,\ f(t,\ \varphi,\ \omega)\rangle. \tag{7.2}$$

Differentiating with respect to ρ we get

$$\left\{\begin{aligned} L_\rho &= H^{\circ}_x\varphi_\rho + H^{\circ}_y\psi_\rho - \langle \psi,\ f_x\varphi_\rho + f_u\psi_\rho\rangle - \langle \psi_\rho,\ f\rangle \\ &= (H^{\circ}_x + \dot{\psi})\varphi_\rho + (H^{\circ}_y - f)\psi_\rho - \langle \psi,\ \varphi_\rho\rangle_t \\ &= -\langle \psi,\ \varphi_\rho\rangle_t. \end{aligned}\right. \tag{7.3}$$

We have used the following substitutions:

$$f_x\varphi_\rho + f_u\omega_\rho = \dot{\varphi}_\rho = \varphi_{\rho t}$$

$$f = H^{\circ}_y,$$

$$\dot{\psi} = -H^{\circ}_x;$$

they are true because φ and ψ satisfy the canonical equations (4.5).

Along our n-parameter family of extremals the derivative of V is given by

$$V_\rho(\tau,\ x,\ T;\ \rho) = \frac{\partial}{\partial\rho}[\lambda(T(\rho),\ \varphi(T,\ \rho)) + \int_\tau^{T(\rho)} L(t,\ \rho)dt].$$

Using (7.3) we obtain

$$V_\rho = \lambda_t(T)T_\rho + \lambda_x(T)\varphi_\rho(T) + L(T)T_\rho + \langle\psi(t),\ \varphi_\rho(t)\rangle\Big|_{T(\rho)}^{\tau}.$$

Using the transversality condition (4.5) this simplifies to

$$V_\rho = [\lambda_t(T) + L(T)]T_\rho + \langle\psi(\tau),\ \varphi_\rho(\tau)\rangle. \tag{7.4}$$

If $T = \text{const.}$, this formula shows that the initial value of y must be chosen in such a way that

$$\begin{aligned} V_\rho(\tau,\ x,\ T;\ \rho) &= \frac{\partial}{\partial\rho} V^o(\tau,\ \varphi_\rho(\tau),\ T) \\ &= \langle V_x^o(\tau),\ \varphi_\rho(\tau)\rangle; \end{aligned}$$

in other words

$$y(\tau) \approx V_x^o(\tau).$$

This is a well-known result; the substitution $y \to V_x^o$ was already encountered in connection with the hamilton-jacobi equation.

The second variation of V is now easily calculated. From (7.3) we have

$$\begin{aligned} -L_{\rho\rho} &= \frac{\partial^2}{\partial\rho\partial t}\langle\psi,\ \varphi_\rho\rangle \\ &= \frac{\partial^2}{\partial t\partial\rho}\langle\psi,\ \varphi_\rho\rangle \\ &= \langle\dot{\psi}_\rho,\ \varphi_\rho\rangle + \langle\psi_\rho,\ \dot{\varphi}_\rho\rangle + \langle\psi,\ \varphi_{\rho\rho}\rangle_t, \end{aligned}$$

or

$$L_{\rho\rho} = \|\varphi_\rho\|^2_{H^o_{xx}} - \|\psi_\rho\|^2_{H^o_{yy}} - \langle\psi,\ \varphi_{\rho\rho}\rangle_t, \tag{7.5}$$

The last step follows by differentiating the canonical equations (4.5) with respect to ρ.

Using (7.5), we obtain the following expression for the second variation of V:

$$
\begin{aligned}
(7.6) \qquad V_{\rho\rho} = {} & [\lambda_t(T) + L(T)]T_{\rho\rho} + 2\langle L_\rho(T),\, T_\rho\rangle + \|T_\rho\|^2_{\lambda_{tt}(T)} \\
& + \langle \psi(\tau),\, \varphi_{\rho\rho}(\tau)\rangle \\
& + \|\varphi_\rho(T)\|^2_{\lambda_{xx}(T)} + \int_\tau^{T(\rho)} [\|\varphi_\rho)t)\|^2_{H^o_{xx}(t)} - \|\psi_\rho(t)\|^2_{H^o_{yy}(t)}]dt.
\end{aligned}
$$

The first two lines depend on quantities associated with the endpoints τ and $T(\rho)$. The third line has the same general appearance as the accessory variational problem. We shall make this relationship precise in the next section; we summarize the discussion so far as follows:

THEOREM. If the variational problem is regular, we may confine the search for an optimal trajectory to the solutions of the "reduced" canonical equations (4.5). Taking account of the boundary conditions, we are led to consider an n-parameter family of solutions of these equations. It suffices to calculate the first and second variation of V along this family, since no other trajectories are admitted into competition for the minimum of V.

This being so, the first and second variation of V may be computed rigorously via ordinary calculus; the formulas for the derivatives of V are given by (7.4) and (7.6). These formulas show that the derivatives consist of two parts: (i) a part dependent only on the derivatives of various functions at the endpoints τ and T of a given extremal; (ii) a part (occurring in $V_{\rho\rho}$ only) which is the V function of the accessory variational problem.

Note that for the accessory variational problem (part (ii) in (7.6)) we may assume that T = const.

The formulas we have obtained must be interpreted with some care. For instance, in the engineering literature the term $\varphi_{\rho\rho}(\tau)$ is sometimes neglected as "small". This is legitimate in certain cases, but for different reasons: Usually we wish to solve the problem for arbitrary initial $x(\tau)$, so that the initial boundary surface may be regarded as free. If we parametrize this boundary surface instead of the one at $t = T$, then we may take $\rho = x(\tau)$, so that $\varphi_{\rho}(\tau)$ = unit matrix and $\varphi_{\rho\rho}(\tau) = 0$ (a three-tensor). But this parametrization may be inconvenient because we have no natural transversality conditions given at $t = \tau$ (we would have to know what $V_x^o(\tau)$ is) and therefore the boundary conditions at $t = T$ may be difficult to meet. For such reasons, it is usually more convenient to parametrize the boundary surface at which there are nontrivial boundary conditions, and in that case it is not true that $\varphi_{\rho\rho}(\tau)$ can be neglected.

The particular choice of parameters adopted here will be useful in our later studies. The reader should be clear, however, that the same technique can be used to investigate V as a function of families of curves which are not necessary extremals. In that case, we consider instead of (7.2) the expression

$$L(t, \rho) = H(t, \varphi, \psi, u) - \langle \psi, f(t, \varphi, u) \rangle$$

where φ is a solution of $\dot{x} = f$ depending parametrically on ρ since $\{u(.)\}$ is a family parametrized by ρ. In order for (7.3) to remain true, we must assume that ψ satisfies

$$\dot{y} = - H_x(t, \varphi, y, \rho); \tag{7.7}$$

then the previous considerations remain valid. If we take the family $\{u(.)\}$ such that

$$x = \varphi(\tau; \rho)$$

for all ρ, then we obtain a necessary condition for minimizing curve as $V_\rho = 0$. This is the old-fashioned way of deriving necessary conditions.

We must emphasize that formulas (7.4) and (7.6) remain valid only as long as ψ satisfies (7.7); each parametrization will in general provide a different ψ.

8. The General Quadratic Variational Problem: Most of our succeeding work will be concerned with the solution of the accessory variational problem, which is also called a quadratic variational problem. We wish to standardize the notation as much as possible, and therefore we now repeat the definition of this problem in a notation that will remain in use throughout the report.

A (continuous-time) quadratic variational problem consists of the following:

(i) Dynamics. This is a linear differential system given by

$$dx/dt = F(t)x + G(t)u(t).$$

(ii) Performance index. This is the functional

$$V(\tau, x, T) = \|x(t)\|^2_{S(T)} + \int_\tau^T [\|x(t)\|^2_{Q(t)} + \|u(t)\|^2_{R(t)}]dt.$$

(iii) Boundary conditions.

At $t = \tau$, $x(\tau) = x$ (τ = const.)

$t = T$, $A(T)x(T) = 0$ (T = const.).

(iv) Assumptions. The matrix R will always be symmetric and positive definite (for regularity) and usually $R = I$. Q and S will always be symmetric.

(v) Hamiltonian.

$$2H^0(t, x, y) = \|x\|^2_{Q(t)} + 2\langle y, F(t)x\rangle - \|G'(t)y\|^2_{R^{-1}(t)}.$$

(vi) Canonical equations.

$$dx/dt = F(t)x - G(t)R^{-1}(t)G'(t)y,$$

$$dx/dt = - Q(t)x - F'(t)y.$$

(vii) Optimal control law.

$$u(t) = - R^{-1}(t)G(t)y(t).$$

Remark. "Quadratic" means that L must be a quadratic function of x and u (the t-dependence may be more or less arbitrary). It may be useful occasionally to consider cases where L contains a term of the type u'Wx. Such problems can be reduced to our present problem by a simple change of variables. See [12]. For most of our work we prefer not to carry along this extra notation in the definition of L. In particular, the riccati equation program is written under the assumption that the term W in L is zero.

It is easy to show now that the last part of (7.6) corresponds to a quadratic variational problem.

Let $\rho_o = 0$ be the parameter corresponding to the extremal $\{\bar{x}(.), \bar{y}(.)\}$ considered in Sect. 6. Write

$$\xi(t) = \delta\varphi(t) = \varphi(t, \rho) - \bar{x}(t) \cong \varphi_\rho(t, 0)\rho$$

$$\eta(t) = \delta\psi(t) = \psi(t, \rho) - \bar{y}(t) \cong \psi_\rho(t, 0)\rho$$

$$\mu(t) = \delta u(t) = u(t, \rho) - \bar{u}(t) \cong - \omega_\rho(t, 0)\rho$$

where $\cong$ means correct "to the first approximation".

Differentiating (4.5) and making use of the abbreviations (6.2) (but dropping superscripts, etc.) we obtain

$$\begin{aligned} \dot{\varphi}_\rho &= F(t)\varphi_\rho - G(t)G'(t)\psi_\rho \\ \dot{\psi}_\rho &= - Q(t)\varphi_\rho - F'(t)\psi_\rho. \end{aligned} \tag{8.1}$$

In other words, φ_ρ and ψ_ρ are a pair of matrix solutions of the canonical differential equations of the generic quadratic variational problem. (To agree with the equations given under (vi) above we must set $R = I$.) In view of this observation we introduce a special notation which is to remain in effect throughout the report:

$$X(t) = \varphi_\rho(t), \quad Y(t) = \psi_\rho(t). \tag{8.2}$$

Then we have

$$\xi(t) = X(t)\rho, \quad \eta(t) = Y(t)\rho, \quad \mu(t) = -G'(t)Y(t)\rho$$

for each fixed ρ. Of course, $\omega_\rho = G'Y$.

Then, if $\rho = \xi(\tau)$, (7.6) shows that

$$\|\xi(\tau)\|^2_{V_{\rho\rho}} = \|\xi(T)\|^2_{S(T)} + \int_\tau^T (\|\xi(t)\|^2_{Q(t)} + \|\mu(t)\|^2)dt \tag{8.3}$$

$$+ \text{terms depending directly on } \tau, T.$$

Thus the critical part of the second variation corresponds to the evaluation of the performance index of a quadratic variational problem along its extremals.

We will see that the study of the accessory extremals provides considerable information about the solution of the two-point boundary-value problem; in fact, this aspect of our theory will turn out to be more important than the somewhat pedantic emphasis on the second variation.

9. Solution of the General Quadratic Variational Problem: Quadratic variational problems can be solved explicitly with the help of the hamilton-jacobi partial differential equation. In our case this equation is

(9.1) $$V_\tau + \langle V_x, F(t)x\rangle - \tfrac{1}{2}\|G'(t)V_x\|^2_{R^{-1}(t)} + \tfrac{1}{2}\|x\|^2_{Q(t)} = 0$$

subject to

(9.2) $$2V(T, x) = \|x\|^2_{S(T)} \quad \text{for} \quad x \in \mathcal{M}.$$

The hamilton-jacobi equation is easily solved in the special case when

(9.3) $$\mathcal{M} = \{(t, x);\ t = T = \text{fixed},\ x = \text{arbitrary}\}.$$

In the remaining cases an indirect procedure must be used. One of our objectives is a detailed explanation of the theory and computing procedure in the general case when $\mathcal{M}$ may be a linear manifold of any dimension. For the moment we shall consider only the special case when $\dim \mathcal{M} = n$.

It can be easily proved that when T = fixed, $\mathcal{M} = \mathcal{H}$ the hamilton-jacobi equation, as well as any first-order partial differential equation, has a unique C^2 solution (as long as λ is C^2 and various other natural smoothness hypotheses hold.) See, e.g., [2, §22].

We will now exhibit this solution.

We assume that

(9.4) $$2V(t, x) = \|x\|^2_{P(t)},$$

where P is a symmetric matrix.

In order that V given by (9.4) constitute the desired solution of the hamilton-jacobi equation, it is necessary that

(9.1') $$-dP/dt = F'(t)P(t) + P(t)F(t) - P(t)G(t)R^{-1}(t)G'(t)P(t) + Q(t)$$

and that

(9.2') $$P(T) = S(T) = \text{const.}$$

The first equation corresponds to the hamilton-jacobi equation and the second corresponds to the initial condition of that equation.

Conditions (9.1'-9.2') are also sufficient, by the uniqueness of solutions of the hamilton-jacobi equation. (Substitute (9.4) into (9.1) and note that V satisfies (9.1) iff P(.) satisfies (9.1'). It is vitally important for the understanding of this report that the reader be familiar with all details of this particular substitution. No tricks are involved, just matrix notation and a large amount of manipulation.).

We have then the

THEOREM. The special quadratic variational problem is solved if and only if the riccati equation (9.1') has a solution $\prod(.;\ T,\ S(T),\ Q(.))$ defined for $t = \tau$.

Note that since (9.1') is a nonlinear differential equation (the quadratic term is essential!) it may be subject to the phenomenon of finite escape time, namely a solution may be defined on an interval $(t',\ T]$ and as $t \to t'$ $P(t) = \prod(t;\ T,\ S(T),\ Q(.)) \to \infty$ (at least one element of $P \to \infty$). Then $t = t'$ is called the (left) conjugate point associated with T. If $\tau \leq t'$, then the solution of our problem does not exist, because (proof later) then $V(\tau,\ x)$ can be made arbitrarily negative by suitable choice of $u(.)$. This obviously cannot happen if both S and Q(.) are nonnegative definite.

The main objective of the theory of the second variation is to investigate the existence and nonexistence of conjugate points, as a function of Q(.), S, the boundary conditions, etc. In the course of this theoretical investigation, we will obtain explicit formulas for the solution of the riccati equation (9.1'). We will also see the role played by controllability in the investigation of conjugate points.

Our considerations are motivated by the roughly correct fact that when no conjugate point exists, the necessary conditions leading to the canonical equations are also sufficient for optimality. In other words,

$$\{\text{extremal + no conjugate point}\} = \{\text{optimal trajectory}\}.$$

We will give a precise discussion of this result, the most important theorem of the classical calculus of variations. The theory surrounding this result also motivates all methods of numerical computation of extremals.

10. Theory of the Riccati Equation Associated with a Quadratic Variational Problem: The general form of the matrix riccati equation is

$$dX/dt = AX + XB + XCX + D.$$

We will not be concerned with such complete generality (whose theory is similar to the case we shall treat) but will deal exclusively with (9.1'). The theory of this equation is very closely related to the theory of the canonical equations.

The solution of the canonical equations is equivalent to the solution of the riccati equation, provided no conjugate point exists.

We consider first the special problem.

As before let the pair $\{X(.), Y(.)\}$ be a matrix solution of the canonical equations, that is.

$$(10.1) \qquad \begin{cases} dX(t)/dt = F(t)X(t) - G(t)R^{-1}(t)G'(t)Y(t), \\ dY(t)/dt = - Q(t)X(t) - F'(t)Y(t), \end{cases}$$

and specify the initial conditions as

$$\text{(10.2)} \qquad \begin{cases} X(T) = I, \\ Y(T) = S. \end{cases}$$

We say that T has no conjugate point (in the special problem) if $\det X(t) \neq 0$, $t < T$. Then we claim

THEOREM. If T has no conjugate point, the solution of the special problem is given by

$$\text{(10.3)} \qquad 2V(t, x) = \|x\|^2_{P(t)} = \|x\|^2_{Y(t)X^{-1}(t)}, \quad t \leq T.$$

Proof. We must show: If the pair $X(.)$, $Y(.)$ satisfies (10.1) with initial conditions (10.2), then $P(.) = Y(.)X^{-1}(.)$ is a solution of the riccati equation (9.1'), with initial conditions $P(T) = H(T)X^{-1}(T) = SI^{-1} = S$.

This is proven by straightforward but tedious substitions, which the reader is again strongly urged to carry out for himself. Similar manipulations occur frequently in deriving the riccati equation for various problems.

Note that our definition of the "nonexistence of a conjugate point" is precisely that necessary and sufficient for the important formula $P = YX^{-1}$ to make sense.

Our theorem admits the following converse:

THEOREM. If the riccati equation (9.1') has a solution defined on $[\tau, T]$, then there is no conjugate point in this interval, that is, $\det X(t) \neq 0$ for $t \in [\tau, T]$. (Of course $X(.)$ is defined via the initial conditions (10.2).)

Proof. If the riccati equation has a solution, then (recall $Y = V_x$) the optimal control law is given by

$$u(t) = - R^{-1}(t)G'(t)Y(t) = - R^{-1}(t)G'(t)P(t)x(t).$$

Therefore the optimal motion $x(.)$ satisfies

$$dx/dt = H_p^o = [F(t) - R^{-1}(t)G'(t)P(t)]x. \tag{10.4}$$

We want to show that $X(.)$, which satisfies (10.1-2), is equal to $\Phi(., T)$, the transition matrix of (10.4), that is, of the optimal closed-loop system. Then the theorem follows, because transition matrices are never singular.

We define

$$\begin{cases} X(.) = \Phi(., T), \\ Y(.) = P(.)X(.). \end{cases}$$

Then $X(T) = I$ and $Y(T) = S$; moreover, the pair $\{X(.), Y(.)\}$ satisfies (10.1). The latter fact is shown by direct substitution into (9.1') and (10.4). This is the third type of substitution which the reader is urged to carry out.

In the general problem the procedure we have used here to relate the riccati equation to the canonical equations (and to relate both to questions concerning the existence of a conjugate point) requires much more delicate arguments. These will be discussed in the remaining parts of this chapter. Before doing so we wish to establish two important properties of the riccati equation:

A. Exact interpolation formula. Although the digital computer cannot compute $P(t)$ continuously in t, we can give a formula which expresses $P(t)$ exactly for any fixed t in terms of the initial values $X(T)$, $Y(T)$ of the canonical equations.

Let Θ be the transition matrix of the canonical equations; i.e., Θ satisfies

$$d\Theta/dt = \begin{bmatrix} F(t) & -G(t)R^{-1}(t)G'(t) \\ -Q(t) & -F'(t) \end{bmatrix} \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix}.$$

Then

$$P(t) = [\Theta_{21}(t, T)X(T) + \Theta_{22}(t, T)Y(T)]\times \times [\Theta_{11}(t, T)X(T) + \Theta_{12}(t, T)Y(T)]^{-1}. \tag{10.5}$$

In other words, to find explicit solutions of the riccati equation we require no more and no less than an explicit transition matrix of the canonical equations. This observation is the most important principle in the entire ASP computing procedure.

Although the right-hand side of (10.5) does not appear to be symmetric, it is known abstractly to be so if $P(T)$ is symmetric. For if $P(T) = S$ is symmetric (which is always assumed, then by (9.1') $P(T)$ = symmetric.

This is very important practically; for if P become unsymmetric due to numerical errors, the errors may propagate and lead to difficulties of various sorts. The user of ASP should always check whether the assumption of symmetry is needed in the derivation of equations to be computed. (The riccati program contains an extra operation which assures that the output is always symmetrical. This is desirable to impede error propagation due to asymmetry.)

One can also show by direct algebraic arguments that the matrix (10.5) is symmetric whenever

(10.6) $$Y(T)X^{-1}(T) = \text{symmetric.}$$

Condition (10.6) plays an important role in the general problem.

B. Symplectic character of Θ. Let

(10.7) $$z = \begin{bmatrix} x \\ y \end{bmatrix} \quad \text{and} \quad J = \begin{bmatrix} 0_n & I_n \\ -I_n & 0_n \end{bmatrix} \quad (J^{-1} = J' = -J).$$

A system of differential equations

(10.8) $$dz/dt = A(t)z$$

is said to be <u>hamiltonian</u> iff it can be written as

$$dx/dt = H_y, \qquad dy/dt = -H_x$$

where

$$H = z'M(t)z$$

and $M(t)$ is a symmetric matrix. It is easy to show that (10.8) is <u>hamiltonian iff</u> $JA(t) = M(t) =$ <u>symmetric matrix</u>. Our canonical equations (8-(vi)) are of course the principal example of a hamiltonian system.

A matrix $X(t)$ is called symplectic iff $X^{-1}(t) = J'X'(t)J$. A symplectic matrix cannot be singular; in fact $\det X(t) \equiv 1$. If X is symplectic, so is its inverse; if A, B are symplectic, then AB is also symplectic. So symplectic matrices form a multiplicative group. "Symplectic" is similar to "orthogonal" and has interesting geometric interpretations (for which see Mal'cev, LINEAR ALGEBRA).

We are interested in symplectic matrices because of the following well-known result:

THEOREM. The transition matrix of a hamiltonian system is symplectic.

Proof. If $\Theta(t, \tau)$ is the transition matrix of a hamiltonian system, then

$$d\Theta/dt = - JM(t)\Theta, \quad M(t) = \text{symmetric}.$$

Let $\Gamma(t, \tau) = \Theta'(\tau, t)$ be the adjoint of Θ. Then Γ satisfies

$$d\Gamma/dt = M(t)J'\Gamma$$

or

$$d[J'\Gamma J]/dt = - JM(t)[J'\Gamma J].$$

Since $J'\Gamma(t, t)J = J'J = I$, $J'\Gamma J$ is a transition matrix; and since it satisfies the same differential equation as Θ, we have

$$J'\Theta'(\tau, t)\ J = \Theta(t, \tau) = \Theta^{-1}(\tau, t).$$

So $\Theta(\tau, t)$ is symplectic and therefore also $\Theta^{-1}(\tau, t) = \Theta(t, \tau)$. Q.E.D.

Using the fact that Θ is symplectic we obtain the identity

$$\Theta(t, \tau) = \begin{bmatrix} \Theta_{11}(t, \tau) & \Theta_{12}(t, \tau) \\ \Theta_{21}(t, \tau) & \Theta_{22}(t, \tau) \end{bmatrix} = \begin{bmatrix} \Theta'_{22}(\tau, t) & - \Theta'_{12}(\tau, t) \\ -\Theta'_{21}(\tau, t) & \Theta'_{11}(\tau, t) \end{bmatrix}.$$

Later it will be useful to know:

PROPOSITION. Let $\{X(.), Y(.)\}$ be matrix solutions of the linear equations (10.1). If

$$X'(t)Y(t) = Y'(t)X(t)$$

for some $t = \tau$, then the same relation holds for all t.

Proof. Write

$$X(t) = \Theta_{11}(t, \tau)X(\tau) + \Theta_{12}(t, \tau)Y(\tau)$$

$$Y(t) = \Theta_{21}(t, \tau)X(\tau) + \Theta_{22}(t, \tau)Y(\tau).$$

Then,

$$\begin{aligned} X'(t)Y(t) = {} & X'(\tau)\Theta'_{11}(t, \tau)\Theta_{21}(t, \tau)X(\tau) \\ & + Y'(\tau)\Theta'_{12}(t, \tau)\Theta_{21}(t, \tau)X(\tau) \\ & + X'(\tau)\Theta'_{11}(t, \tau)\Theta_{22}(t, \tau)Y(\tau) \\ & + Y'(\tau)\Theta'_{12}(t, \tau)\Theta_{22}(t, \tau)Y(\tau). \end{aligned}$$

Now

$$\begin{aligned} \Theta'_{11}(t, \tau)\Theta_{21}(t, \tau) &= \Theta_{22}(\tau, t)\Theta_{21}(t, \tau) && \text{(by symplectic)} \\ &= \Theta_{21}(\tau, t)\Theta_{11}(t, \tau) && \text{(composition property of transition matrices} \\ &= \Theta'_{21}(t, \tau)\Theta_{11}(t, \tau) && \text{(by symplectic).} \end{aligned}$$

The other terms may be rearranged similarly. Thus a term-by-term comparison shows that the proposition is true.

Similarly, we can prove that $P(t)$ given by (10.5) is a symmetric matrix iff $P(t) = Y(T)X^{-1}(T)$ is symmetric.

11. n-Parameter Families of Extremals: Let ρ be a constant n-vector. Given any matrix pair $\{X(.), Y(.)\}$ satisfying (10.1), we define an n-parameter family of vector solutions of (10.1) by setting

$$\begin{cases} \xi(t) = X(t)\rho \\ \eta(t) = Y(t)\rho. \end{cases} \tag{11.1}$$

(We are applying the considerations of Sec. 7 to the general quadratic variational problem. According to the preceding theory every extremal of a quadratic variational problem must be of the form (11.1).

Conversely, we may ask the question: Is every pair $\{\xi(.), \eta(.)\}$ of solutions of (8-(vi)) a (regular pseudo-) extremal of the associated quadratic variational problem?

The answer is well known to be NO. There is no difficulty in satisfying the minimum requirement (4.3) (since u_{opt} is a function only of y and since the relation $u_{opt}(t) = -R^{-1}(t)G'(t)y(t)$ has been built into (10.1)). However, there is an additional constraint imposed by the transversality conditions (4.4).

PROPOSITION. An n-parameter family of solutions of (8-(vii)) defined by (11.1) is a family of extremals of a quadratic variational problem if and only if

$$X'(T)Y(T) = Y'(T)X(T). \tag{11.2}$$

In view of the fact that (8-(vi)) is a hamiltonian system, relation (11.2) guarantees that $X'(t)Y(t) = Y'(t)X(t)$ for all t.

Proof. The transversality condition imposes on $X(T)$ and $Y(T)$ the requirements

(11.3) $$A(T)X(T)\rho = 0 \quad \text{for all} \quad \rho,$$

(11.4) $\langle Y(T)\rho, z\rangle = \langle S(T)X(T)\rho, z\rangle$ for all ρ and all z satisfying $A(T)z =$

Therefore any row vector of the type

$$z'[Y(T) - S(T)X(T)]$$

is zero; since by (11.3) $X(t)\rho = z$ satisfies $A(T)z = 0$,

$$\rho'[X'(T)Y(T) + X'(T)S(T)X(T)] = 0$$

for all ρ, which proves (11.2).

We will now derive a canonical set of coordinates which will enable us to state the constraints (11.3-4) in a particularly simple form. In other words, we translate the transversality conditions into a specification of the pair $X(T)$, $Y(T)$.

Let $n - r$ be the rank of $A(T)$ (hence $r = \dim \mathcal{B}$). By changing coordinates we can exhibit $A(T)$ in the canonical form

(11.5) $$A(T) = [0 \quad I_{n-r}].$$

I.E., we rotate the terminal manifold $\mathcal{B}$ so that it is given by

$$\gamma^i(x) = x^i = 0, \quad i = 1, \ldots, r.$$

Then x can be written as (x^1, x^2), where x^1 is an r-vector and x^2 is an $(n-r)$-vector; x belongs to the terminal manifold iff $x^2 = 0$. As in Sect. 7, we parametrize the terminal manifold by the r-vector μ. Then

$x \cdot$ belongs to the terminal manifold iff $x = (\mu, 0)$. μ must be part of the parameter ρ which determines our n-parameter family of extremals. Hence we set (cf. Sect. 7)

$$\rho = (\mu, \nu).$$

Let us determine the special form of the matrices $X(T)$ and $Y(T)$ in the coordinate system introduced above, writing them in such a way that the condition (11.2) is automatically satisfied.

Since $\mathcal{M}$ is to be parametrized by μ and since every $x(T) = X(T)\rho$ must belong to $\mathcal{M}$, i.e., $x(T) = (\mu, 0)$, it is clear that

$$X(T) = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}.$$

Condition (11.4) shows that

$$Y(T) = \begin{pmatrix} S_{\mu\mu} & 0 \\ B & C \end{pmatrix}$$

where B, C are arbitrary matrices. We set $\Sigma = S_{\mu\mu}$ for the sake of a simpler notation. Then

$$y(T) = Y(T)\rho = (S_{\mu\mu}\mu,\ B\mu + C\nu).$$

The second term is an arbitrary $(n-r)$-dimensional parameter. We lose no generality if we set $B = 0$, $C = I$. Thus

$$Y(T) = \begin{bmatrix} \Sigma & 0 \\ 0 & I \end{bmatrix}.$$

It is clear that (11.6) and (11.7) define the most general n-parameter family of solutions of (8-(vii)), which are extremals.

12. Two Important Formulas: To complete our discussion of the parametric representation of extremals, let us calculate the optimal performance index.

The first formula is

$$\int_t^T L(t, \xi(t), u_{opt}(t))dt = \langle \xi(t), \eta(t) \rangle \Big|_T^\tau. \tag{12.1}$$

To prove it, write

$$L(t, x, u_{opt}) = 2H^o(t, x, y) + \|G'y\|^2_{R^{-1}} - 2y'Fx.$$

By homogeneity of H^o,

$$L(t, x, u_{opt}) = H_x x + H_y y + \|G'y\|^2_{R^{-1}} - 2y'Fx$$

Using the canonical equations we get

$$\begin{aligned} L(t, x, u_{opt}) &= -x'\dot{y} + y'\dot{x} - 2y'(Fx - G'R^{-1}G_y) \\ &= -y'\dot{x}), \end{aligned}$$

which proves (12.1).

Now suppose that $\det X(\tau) \neq 0$. Then we can write the right-hand side of (12.1) as a function of ρ, as follows:

$$\xi(\tau) = X(\tau)\rho$$

$$\rho = X^{-1}(\tau)\ (\tau).$$

On the other hand,

$$\langle \xi(T),\ \eta(t) \rangle = \rho' X'(T) Y(T) \rho$$

$$= \|\mu\|_{\Sigma}^2$$

by (11.7). Thus (12.1) becomes

$$\|\mu\|_{\Sigma}^2 + \int_{\tau}^{T} L(t,\ \xi(t),\ u_{opt}(t))dt = \|\xi(\tau)\|^2_{Y(\tau)X^{-1}(\tau)} \tag{12.2}$$

which is our second important formula.

Suppose now that we have not merely $\det X(\tau) \neq 0$, but

$$\det X(t) \neq 0 \quad \text{for all} \quad [\tau,\ T). \tag{12.3}$$

Then according to Sect. 10 we know that $Y(t)X^{-1}(t)$ satisfies the hamilton-jacobi partial differential equation on the interval $[\tau,\ T)$. (Note that the right endpoint $T = t$ is not included in this statement since $X(T)$ is singular when $r < n$.) But the existence of solutions of the hamilton-jacobi equation is a sufficient condition for optimality. Hence we have our

FIRST MAIN THEOREM: <u>If an n-parameter family of extremals satisfies (12.3), then</u> $X(.)\rho$ <u>is an optimal trajectory for each</u> ρ.

We now embark on a detailed investigation of the conjugate point condition (12.3). It should be remarked right away that (12.3) is not a <u>necessary</u> condition for optimality; the degree to which it fails to be optimal is related to the controllability properties of the family of extremals.

13. The Boundary-Value Problem: Let us return now for a moment to the original (not accessory) variational problem. If this problem has a solution in terms of a (regular pseudo-) extremal, then this extremal must be determined using the free canonical equations (4.5). This means that we must know either the two initial conditions, $x(\tau)$ and $y(\tau)$ or the two final conditions, $x(T)$ and $y(T)$. $x(\tau)$ is given as part of the specification of the problem, while $y(\tau)$ must be determined by solving a so-called "two-point boundary-value problem". Usually neither $x(T)$ nor $y(T)$ is given.

Let us consider some typical examples. Recall that $\varphi(t; \tau, x(\tau), y(\tau))$ and $\psi(t; \tau\, x(\tau), y(\tau))$ is the solution pair of (4.5) corresponding to the initial values $x(\tau)$, $y(\tau)$.

(1) The endpoint is free, i.e., $-\beta = \mathfrak{X}$. Then we must satisfy the condition (obtained from the transversality condition)

$$
\begin{aligned}
(13.1) \qquad y(T; \tau, x(\tau), y(\tau)) &= y(T) \\
&= \lambda_x(T, x(T)) \\
&= \lambda_x(T, \varphi(T; \tau, x(\tau), y(\tau))).
\end{aligned}
$$

This is an implicit relationship which determines $y(\tau)$ as a function of $x(\tau)$. Once $y(\tau)$ is determined, the extremals can be computed by (4.5). If (13.1) has a solution, $x^0(\tau)$ and $p^0(\tau)$, i.e., if we have determined a particular extremal*, then using the implicit function theorem we can solve (13.1) provided the relevant jacobian is $\neq 0$. This jacobian can take several different forms, depending on the variables which are designated as unknowns. In our present case, it is convenient to choose not $p(\tau)$ but $x(T)$ as the unknown. Then we get the condition

$$
(13.2) \qquad \varphi(\tau; T, x(T), \lambda_x(T, x(T))) = x(\tau)
$$

(equivalent to (13.1)), from which we are to determine $x(T)$ as a function of $x(\tau)$. It is well known in the theory of differential equations that the jacobian

*We may do this as follows: We pick $x(T)$ arbitrarily, set $p(T) = \lambda_x(T, x(T))$, and then integrate (4.5) from T toward τ.

$$\left[\frac{\partial x_i(\tau)}{\partial x_j(t)}\right] = \left[X_{ij}(\tau)\right]$$

where $X(.)$ is that solution of our canonical equations (10.1) which corresponds to the given boundary condition and $\Sigma = \lambda_{xx}$. In other words, the local solution of equation (13.2) (in the approximate sense, considering only the linear term in the taylor series of the left-hand side of (13.2)) requires knowledge of the solution of the accessory problem. Moreover, a solution is possible if (usually also only if) $\det X(\tau) \neq 0$. So we see that the conjugate point condition is closely related to the solution of the "boundary-value problem" (13.2).

The engineering literature contains many misleading statements concerning the solution of the boundary-value problem. In effect this problem is merely a part of the classical theory of the accessory problem. We hasten to add that the entire theory is local. (If one had any mathematical idea concerning the global solution of the boundary-value problem, that would be nearly as much as a theory of the global accessory problem, that is to say, a global theory of the original nonlinear variational problem.)

The choice of $t = T$ as the starting point for the solution of the canonical equations (10.1) is quite arbitrary. We could have started also at $t = \tau$. Which of these choices is more sensible depends on the practical features of the problem.

Let us illustrate what would happen in the second case. (We let T play the role of τ.)

(2) The endpoint is completely constrained, i.e. $\mathcal{B} = \{0\}$. Now we must satisfy the condition

$$\varphi(\tau;\ T,\ 0,\ y(T)) = x(\tau). \tag{13.3}$$

In this case $y(T)$ = free. The jacobian is again given by

$$\left[\frac{\partial x_i(\tau)}{\partial p_j(T)}\right] = \left[X_{ij}(\tau)\right]$$

(X suitably defined).

Both (13.2) and (13.3) can be expressed by a single formula with the aid of our parametrization. Thus in the first example we have that

$$(x(T),\ \lambda_x(T,\ x(T)) = (\mu,\ \Sigma\mu)$$

and in the second case

$$(0,\ y(T)) = (0,\ \nu).$$

In the general case, we have

(13.4) $$(x(T),\ y(T)) = (X(T)\rho,\quad Y(T)\rho).$$

This gives our

SECOND MAIN THEOREM: The jacobian matrix

$$\left[\frac{\partial x_i(\tau)}{\partial \rho_j}\right]$$

of the general variational problem is given by the corresponding matrix $X(\tau)$.

An important reason for our parametric representation of extremals is to have a simple expression for the jacobian.

14. Two Important Lemmas: Now we wish to investigate the necessity of the condition (12.3).

It is clear that if this condition is not satisfied, then the problem is not well posed. For if $\det X(t_1) = 0$ for some $t_1 \in [\tau, T)$, then it is impossible to connect some point (t_1, ξ_1) with the terminal set $\mathcal{B}$ extremally, i.e.,

$$\xi_1 \notin \text{range } [X(t_1)].$$

However, even in this case it is conceivable that we might be able to connect a point (t_1, ξ_1) with the terminal set $\mathcal{B}$ in some nonextremal and therefore nonoptimal way. It turns out that this is not the case. In other words,

{reachability of terminal surface} = {optimal reachability}

provided there is no conjugate point in the interval $[\tau, t)$.

Let us now make these notions precise.

LEMMA 1. If $X(t_0)\rho_0 = 0$ for some $\rho = \rho_0 = (0, \nu_0)$, $\nu_0 \neq 0$, $\xi(t) = X(t)\rho_0 \not\equiv 0$ on $[t_0, T)$ iff the boundary value problem is well posed over the interval $[t_0, T)$. [That is, given any state x there is a control $u(\cdot)$ which connects (t_0, x) with (T, z), $z \in \mathcal{B}$.]

Proof. Suppose $\xi(t) = X(t)\rho_0 \equiv 0$ on $[t_0, T)$. Then

$$\dot{\eta} = -F'(t)\eta.$$

This equation can be explicitly solved; the answer is

$$\eta(t) = \Phi'(T, t)Y(T)\rho_0.$$

Then, since $X(T)\rho_0 = (\mu_0, 0) = 0$, the equation

$$\dot{\xi} = F(t)\xi - G(t)R^{-1}(t)G'(t)\Phi'(T, t)Y(T)\rho_0$$

has the solution

$$\xi(T) = \Phi(T, t_o)\xi(t_o) = [\int_{t_o}^{T} \Phi(T, t)G(t)R^{-1}(t)G'(t)\Phi'(T, t)dt]Y(T)\rho_o,$$

$$= W(T, t_o)Y(T)\rho_o,$$

$$= 0.$$

$W(T, t_o)$ is the reachability matrix; a point (T, z), $z \in \mathcal{B}$ is reachable from (t_o, x) iff $z - \Phi(T, t_o)x \in$ range $W(T, t_o)$ (see [5]). That is,

$$x = \Phi(t_o, T)(w + z), \quad w \in \text{range } W, \quad z \in \mathcal{B}.$$

Thus the problem is well posed only if

(14.2) $$\text{range } W + \mathcal{B} = \mathcal{X}.$$

Now $(0, \nu_o) \perp \mathcal{B}$.

On the other hand, (14.1) and (11.7) show that

$$WY(T)\rho_o = W(0, \nu_o) = 0$$

so that $(0, \nu_o) \in$ null space W. Since W is symmetric, this means that $(0, \nu_o) \perp$ range W. Hence $(0, \nu_o) \perp [\text{range } W + \mathcal{B}]$, and since $\nu_o \neq 0$ we have a contradiction to (14.2). This proves Lemma 1.

If the hypotheses of the lemma hold, we say that t_o is a <u>conjugate point</u> (precise definition) of T. According to the lemma, a conjugate point is characterized by the fact that at least two distinct extremals connect the points $(0, t_o)$ and $(0, T)$. $\xi(t) \equiv 0$ is always an extremal; it corresponds to $\rho = 0$. By the lemma the extremal $\xi(t) = X(t)\rho_o$ is distinct from $\xi(t) \equiv 0$.

If $\det X(t_o) \neq 0$, we may have also the case where $X(t_o)\rho = 0$ and $\rho_o = (\mu_o, 0)$, $\mu_o \neq 0$. In this case the extremal $\xi(t) \equiv 0$ is obviously different from $\xi(t) = X(t)\rho_o$ because $x(T) = (\mu_o, 0) \neq 0$. When there is a ρ_o with the latter properties, we say that t_o is a <u>focal point</u> (precise definition) of T. Slightly imprecisely a focal point may be regarded as a generalization of a conjugate point and includes the latter as a special case. The optical intuition leading to the term "focal point" should be quite clear.

Another definition of the focal [conjugate] point is: there exist two distinct extremals which terminate at $x(t_o) = 0$ [and begin at $x(T) = 0$]. In view of our lemma, this is equivalent to the statement: there exists a $\rho_o \neq 0$ [$\rho_o \neq 0$ with $\nu_o \neq 0$] such that $X(t_o)\rho_o = 0$.

LEMMA 2. <u>Suppose that the boundary-value problem is well posed over any interval</u> $[\tau, T)$, $\tau < T$. (<u>Refer Lemma 1 for detailed statement.</u>) <u>Then there is no focal point in</u> $[\tau, T)$ <u>if</u> τ <u>is taken to be sufficiently small.</u>

<u>Proof</u>. If there is a focal point at $t = \tau$, then there is a $\rho_o \neq 0$ such that

$$x(\tau) = X(\tau)\rho_o = 0.$$

Let γ_o be the extremal which corresponds to ρ_o. In view of formulas (14.1-2), it is clear that the performance index V along γ_o is zero (since $x_o(\tau) = 0$).

Now we define the scalar function

$$V(t, x) = \|x\|^2_{P(t)} \tag{14.3}$$

by letting $P(.)$ be given by (10.5), where

$$X(T) = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}, \quad Y(T) = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix}.$$

It is clear that $P(t)$ will be well defined (will exist) for t near T because $\det X(T) = 1$ and therefore by continuity $\det X(t) \neq 0$ for t near T. According to the theory discussed in Sects. 5 and 9, V is a solution of the hamilton-jacobi equation corresponding to a new variational problem P_2; P_2 is the same as the original problem P, (for which there exists a focal point) except for the fact that the boundary surface for P_2 is $\mathcal{M} = \mathcal{K}$. Hence γ_o is an extremal also for P_2 because it satisfies the boundary condition (trivially) for P_2. So the performance index V for γ_o is 0 also for P_2.

The function V defines a unique control law, which is linear. Hence in P_2 the optimal trajectory γ_∞ starting at $x(\tau) = 0$ is the curve identically zero. Along this curve the performance index is obviously zero.

By Carathéodory's lemma (see [4]) the existence of a solution of the hamilton-jacobi equation implies that the optimal trajectories are unique. Both γ_o and γ_∞ are optimal because their performance index is 0. We have a contradiction to uniqueness.

This completes the proof of the lemma.

We note also the

COROLLARY. Under the hypotheses of Lemma 1, the focal points of T are isolated.

Proof. If not, let $[t_o, t_1]$ be an interval of focal points. Since $X'(t_1)Y(t_1)$ is symmetric, we can repeat the analysis of the lemma, choosing t_1 as T. Then the lemma shows that there are no focal points in the interval $[\tau, t_1]$ if τ is near t_1. Contradiction.

Notice also that our proof of Lemma 2 implies the

PROPOSITION. If $t_1 < T$ is the focal point nearest to T in problem P_1 and t_2 is the analogous focal point in P_2, then $t_1 \leq t_2 < T$.

15. Summary: We can now state our final result, which shows that the computational treatment of the general boundary conditions is identical with the treatment of the special case when $x(T)$ is free

THIRD MAIN THEOREM. Assume that the boundary-value problem is well posed over any interval $[t, T)$ (where $t \in [\tau, T)$) and that τ is chosen so small that no focal point exists in the interval $[\tau, T)$ (that is, $\det X(t) \neq 0$ on $[\tau, T)$).

Then the solution of the general problem may be obtained using the function V defined via (9.4) by $P(t)$ which is given by (9.1'), where $X(T)$ and $Y(T)$ are determined by (10.1) and (10.2). This choice of initial conditions for (10.1) assumes that the terminal surface consists of the first r of the n (orthogonal) coordinate axes. The function V so computed satisfies the relevant hamilton-jacobi equation for all $t \in [\tau, T)$; if $t = T$, then $V(T, x) = \|x\|_{\Sigma}^2$ on $\mathcal{B}$.

The statement: "the boundary-value problem is well posed over every interval $[t, T)$" is satisfied, e.g., when the system (3.1) is completely controllable over every such interval. Our italicized statement is in fact equivalent in classical terminology to the statement "the problem is normal in every interval".

The theory can be carried out without such an assumption, as follows.

Suppose the problem is well posed merely with respect to the fixed interval $[\tau, T)$. Consider the linear space $\Gamma(t)$ of all points (t, x) from which $\mathcal{B}$ may be reached at $t = T$. By definition of "well posed at $t = \tau$" we have that $\Gamma(\tau) = \mathcal{X}$. But for $t > \tau$ $\Gamma(t)$ may be a proper subspace of $\mathcal{X}$.

By the theory of controllability, it follows very easily [10] that $\operatorname{Dim} \Gamma(t)$ is a piecewise integer-valued function, which cannot be decreased as $t \to \tau$. Moreover, there are but finitely many intervals on which this function has different values (because $\mathcal{X}$ is finite-dimensional). Hence let us

subdivide $[\tau, T]$ into maximal disjoint intervals on which $\dim \Gamma(t) = \text{constant}$. Each interval is closed on the left and open on the right. We repeat the entire analysis (which is quite independent of the specification of the initial state $x(\tau)$: all of our analysis holds for all initial states for which the boundary-value problem makes sense) for each interval. Putting together the pieces, we get a result analogous to the third main theorem.

The possibility of such a procedure was pointed out a long time ago by Carathéodory [11], but his formulas are not explicit, contain some mistakes (rectified in Chapter 18 of [2]), and are not easy to read. Our present treatment is similar to Caratheodory's exposition of the theory of "second variation", [2, Chapter 15] which was concerned solely with the simple variational problems ($F = 0$, $G = I$). As the reader has seen, the Hamilton-Jacobi-Carathéodory-Bellman approach can give us the solution of the most general problem, and has the advantage in addition of yielding a single formula, (9.4), which covers all computing problems.

It is hardly necessary to point out that the theoretical part of our investigation is completed by applying the theory of the accessory problem to the formulas developed in Sect. 7. Most of the classical sufficiency proofs of the calculus of variations are done in this style. We omit the details, since they have no bearing on the computational problem.

16. Computation: The transition matrix for (10.1) can be computed easily only when all matrices are constants. In that case

$$\Theta(t, \tau) = \exp[(t - \tau)Z], \tag{16.1}$$

where

$$Z = \begin{pmatrix} F & -GR^{-1}G' \\ -Q & -F \end{pmatrix}.$$

The program is supplied with (16.1) and then computes $P(t)$ by substituting the four submatrices of Θ into one of the formula (10.5). In this way one obtains a stepwise solution of the riccati equation.

At each step P is symmetrized before proceeding by replacing it with

$$\frac{P(t) + P'(t)}{2}.$$

Symmetrization is absolutely essential because otherwise uncontrollable roundoff errors may accumulate in the antisymmetric part of $P(t)$.

The input to the program consists of the matrices Θ, $R^{-1}G$, a symmetric $P(\tau)$, a sampling period T at which intervals P will be computed, a convergence criterion number ϵ, a final time TF, and various printing codes.

The problem is terminated in one of two ways. Either the final time TF is reached or the convergence criterion is satisfied.

The convergence criterion is that

$$\frac{\sum_{i=1}^{n} |p_{ii}(t + T) - p_{ii}(t)|}{\sum_{i=1}^{n} |p_{ii}(t + T)|}$$

be less than the input number ϵ.

When $P(t)$ is computed, the matrix

$$K(t) = R^{-1}G'P(t)$$

specifying the control law can be computed and printed. Print controls enable one to print K or P at every N steps and at the final step.

17. Checks: A) The Program was run with

$$F = \begin{bmatrix} 0_6 & I_6 \\ 0 & 0'_6 \end{bmatrix}$$

where 0_6 = six dimensional zero column vector and $I_6 = 6 \times 6$ identity matrix

$$G = \begin{bmatrix} 1 \\ \\ 0_6 \end{bmatrix}, \quad Q = H'H = 7 \times 7 \text{ matrix}$$

$$R = [0.6075]$$

$$P(0) = 0.6075\ I,$$

$$T = -0.2.$$

This was iterated for ten steps and the result compared with a hand-computed result expressed exactly in four place decimals. P appears in Fig. 2. The hand-computed P is:

$$P = \begin{bmatrix} 0.2025 & 0.4050 & 0.4050 & 0.2700 & 0.1350 & 0.0540 & 0.0180 \\ 0.4050 & 1.4175 & 2.0250 & 1.7550 & 1.0800 & 0.5130 & 0.1980 \\ 0.4050 & 2.0250 & 3.8475 & 4.1850 & 3.1050 & 1.7280 & 0.7650 \\ 0.2700 & 1.7550 & 4.1850 & 5.8275 & 5.4450 & 3.7170 & 1.9680 \\ 0.1350 & 1.0800 & 3.1050 & 5.4450 & 6.6375 & 5.8410 & 3.8730 \\ 0.0540 & 0.5130 & 1.7280 & 3.7170 & 5.8410 & 6.8319 & 5.9178 \\ 0.0180 & 0.1980 & 0.7650 & 1.9680 & 3.8730 & 5.9178 & 6.8623 \end{bmatrix}.$$

B) The Program was run with

$$F = \begin{bmatrix} 0_6 & 0 \\ -I_6 & 0_6 \end{bmatrix} \qquad H = [1 \quad 0_6']$$

$$G = R^{-1} = 7 \times 7 \text{ zero matrix}$$

$$Q = 2.0250$$

$$P(0) = -0.2I$$

This was iterated for ten steps and the results compared with a hand-computed result which was expressed exactly in four place decimals. P appears in Fig. 3, K is of course zero. The hand-computed result is:

$$P = 10. \begin{bmatrix} 2.6935 & -1.9759 & 1.2990 & -0.6720 & 0.2790 & -0.0900 & 0.0180 \\ -1.9750 & 2.2869 & -1.9710 & 1.2870 & -0.6480 & 0.2430 & -0.0540 \\ 1.2990 & -1.9710 & 2.2725 & -1.9350 & 1.2150 & -0.5400 & 0.1350 \\ -0.6720 & 1.2870 & -1.9350 & 2.1825 & -1.7550 & 0.9450 & -0.2700 \\ 0.2790 & -0.6480 & 1.2150 & -1.7550 & 1.8225 & -1.2150 & 0.4050 \\ -0.0900 & 0.2430 & -0.5400 & 0.9450 & -1.2150 & 1.0125 & -0.4050 \\ 0.0180 & -0.0540 & 0.1350 & -0.2700 & 0.4050 & -0.4050 & 0.2025 \end{bmatrix}$$

C) To obtain a problem whose answer we knew for all time, we ran the following

$$F = \text{diag } (.1, .2, .5, 1, 2)$$

$$H'QH = \begin{bmatrix} .1 & .2 & .3 & .4 & .5 \\ .2 & .2 & .4 & .5 & .6 \\ .3 & .4 & .3 & .6 & .7 \\ .4 & .5 & .6 & .4 & .8 \\ .5 & .6 & .7 & .8 & .5 \end{bmatrix}$$

$GR^{-1}G' = 5 \times 5$ zero matrix $= P(0)$.

P was computed at various times, we show in Fig. 4 and Fig. 5 the computed results at $t = .75$ and $t = 3.7$. The answers are accurate in the seventh significant figure. The correct results are

$$p_{ij}(t) = q_{ij} \frac{e(\lambda_i + \lambda_j)t - 1}{\lambda_i - \lambda_j}.$$

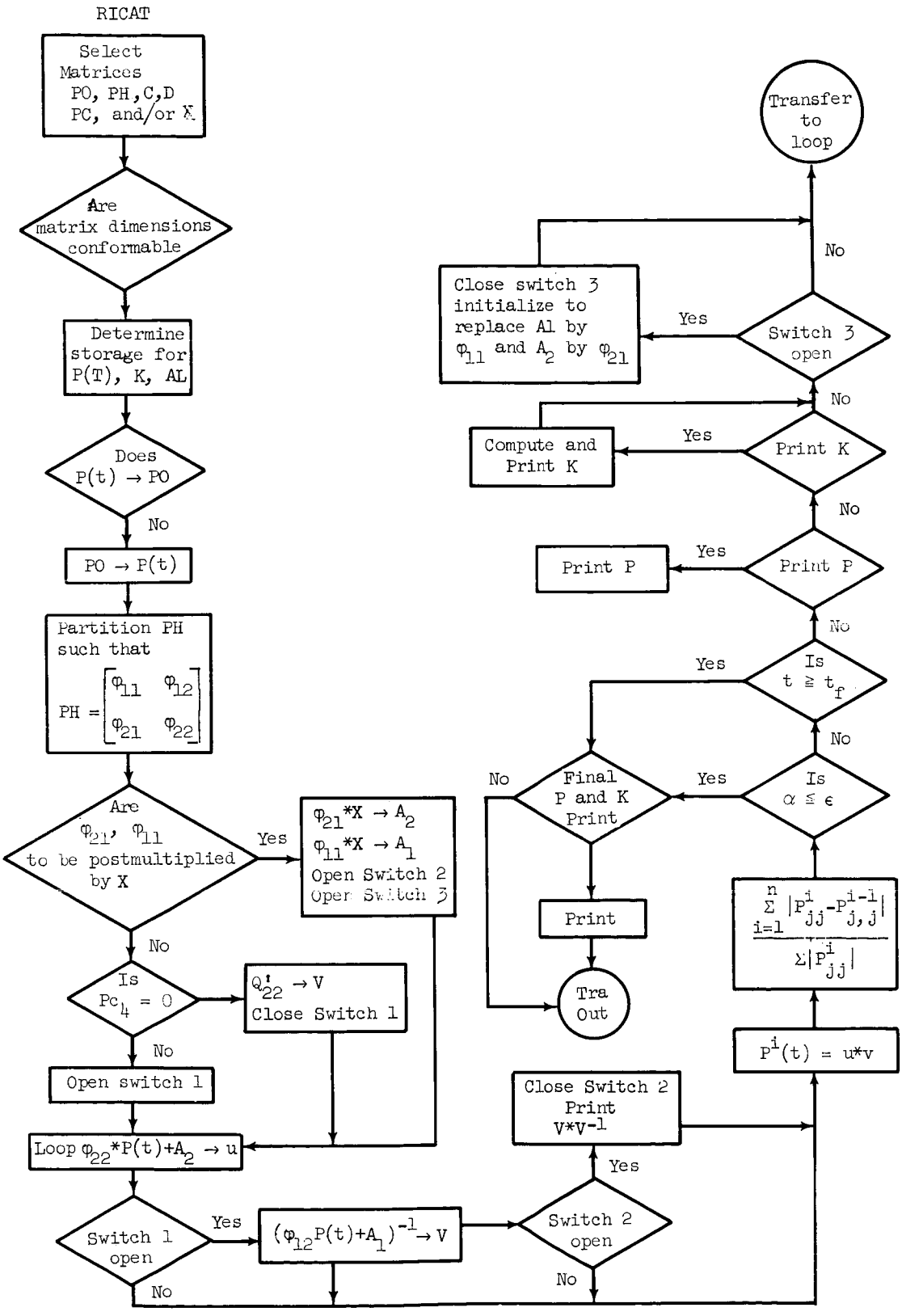
RICAT
Select Matrices PO, PH, C, D PC, and/or X
Are matrix dimensions conformable
Determine storage for P(T), K, AL
Does P(t) → PO
No
PO → P(t)
Partition PH such that
PH = [φ11 φ12; φ21 φ22]
Are φ21, φ11 to be postmultiplied by X
Yes
φ21*X → A2
φ11*X → A1
Open Switch 2
Oper Switch 3
No
Is Pc4 = 0
Q'22 → V
Close Switch 1
No
Open switch 1
Loop φ22*P(t)+A2 → u
Switch 1 open
Yes
(φ12P(t)+A1)^-1 → V
No
Switch 2 open
Yes
Close Switch 2
Print
V*V-1
No
P^i(t) = u*v
Σ_{i=1}^{n} |P^i_jj - P^{i-1}_j,j| / Σ|P^i_jj|
Is α ≤ ε
Yes
No
Is t ≥ t_f
Yes
No
Print P
Yes
No
Print K
Yes
No
Compute and Print K
Switch 3 open
Yes
No
Close switch 3 initialize to replace A1 by φ11 and A2 by φ21
Transfer to loop
Final P and K Print
No
Yes
Print
Tra Out

Fig. 1

MATRIX P10

NUMBER OF ROWS 7 NUMBER OF COLUMNS 7

0.20249999E-00	0.40499996E-00	0.40499996E-00	0.26999997E-00	0.13499998E-00	0.53999992E-01
0.17999997E-01					
0.40499996E-00	0.14174999E 01	0.20249997E 01	0.17549997E 01	0.10799998E 01	0.51299991E 00
0.19799996E-00					
0.40499996E-00	0.20249997E 01	0.38474994E 01	0.41849993E 01	0.31049994E 01	0.17279997E 01
0.76499987E 00					
0.26999997E-00	0.17549997E 01	0.41849993E 01	0.58274991E 01	0.54449991E 01	0.37169994E 01
0.19679996E 01					
0.13499998E-00	0.10799998E 01	0.31049994E 01	0.54449991E 01	0.66374990E 01	0.58409990E 01
0.38729993E 01					
0.53999992E-01	0.51299991E 00	0.17279997E 01	0.37169994E 01	0.58409990E 01	0.68318988E 01
0.59177990E 01					
0.17999997E-01	0.19799996E-00	0.76499987E 00	0.19679996E 01	0.38729993E 01	0.59177990E 01
0.68622989E 01					

Fig. 2

MATRIX P10

NUMBER OF ROWS 7 NUMBER OF COLUMNS 7

0.26934986E 02	-0.19757989E 02	0.12989993E 02	-0.67199971E 01	0.27899990E 01	-0.89999975E 00
0.17999996E-00					
-0.19757989E 02	0.22868989E 02	-0.19709991E 02	0.12869995E 02	-0.64799979E 01	0.24299993E 01
-0.53999989E 00					
0.12989993E 02	-0.19709991E 02	0.22724991E 02	-0.19349993E 02	0.12149996E 02	-0.53999987E 01
0.13499998E 01					
-0.67199971E 01	0.12869995E 02	-0.19349993E 02	0.21824993E 02	-0.17549996E 02	0.94499981E 01
-0.26999996E 01					
0.27899990E 01	-0.64799979E 01	0.12149996E 02	-0.17549996E 02	0.18224996E 02	-0.12149998E 02
0.40499996E 01					
-0.89999975E 00	0.24299993E 01	-0.53999987E 01	0.94499981E 01	-0.12149998E 02	0.10124999E 02
-0.40499997E 01					
0.17999996E-00	-0.53999989E 00	0.13499998E 01	-0.26999996E 01	0.40499996E 01	-0.40499997E 01
0.20249999E 01					

Fig. 3

MATRIX P (0.75000000E 00)

NUMBER OF ROWS 5 NUMBER OF COLUMNS 5

0.80917116E-01	0.16821513E-00	0.28415607E-00	0.46613841E-00	0.91208123E 00
0.16821513E-00	0.17492939E-00	0.39454786E-00	0.60816789E 00	0.11473579E 01
0.28415607E-00	0.39454786E-00	0.33509998E-00	0.83208665E 00	0.15458290E 01
0.46613841E-00	0.60816789E 00	0.83208665E 00	0.69633775E 00	0.22633958E 01
0.91208123E 00	0.11473579E 01	0.15458290E 01	0.22633958E 01	0.23856918E 01

Fig. 4

MATRIX P (0.37000000E 01)

NUMBER OF ROWS 5 NUMBER OF COLUMNS 5

0.54796769E 00	0.13562386E 01	0.41036645E 01	0.20929799E 02	0.56368339E 03
0.13562386E 01	0.16964725E 01	0.70455819E 01	0.34906216E 02	0.93488632E 03
0.41036645E 01	0.70455819E 01	0.11834189E 02	0.10249499E 03	0.29129970E 04
0.20929799E 02	0.34906216E 02	0.10249499E 03	0.32699680E 03	0.17645368E 05
0.56368339E 03	0.93488632E 03	0.29129970E 04	0.17645368E 05	0.33455532E 06

Fig. 5

CHAPTER VIII

THE SAMPLED-DATA RICCATI PROGRAM

1. Description of the Problem: For many purposes, the variational theory presented in the preceding chapter is more conveniently applied in discrete-time. This is the topic of the present chapter.

2. Theory: Consider the linear dynamical system

$$x_{t+1} = \Phi_t x_t + \Gamma_t u_t, \tag{2.1}$$

where

t = integer,

x_t = $(p \times 1)$ the state vector,

Φ_t = $(p \times p)$ transition matrix, a function of t,

Γ_t = $(p \times q)$ input matrix, a function of t,

u_t = $(q \times 1)$ the control vector.

Such a system may arise in a variety of ways. Let us assume only that we are given a linear system, that is Φ_t and Γ_t are independent of x_t or u_t.

The control problem we will consider may be stated as follows.

Given the desired value x^d of the state vector, manipulate the sequence of control vectors $\{u_t\}$ in such a way as to bring the state vector x_t rapidly as close as possible to x^d, and then keep the state vector near x^d at all times.

Let Q be any nonnegative definite symmetric matrix. As before $\|x - x^d\|_Q^2$ will denote the quadratic form $(x - x^d)'Q(x - x^d)$. This pseudo-metric (metric if Q is positive definite) will be our definition of the distance between states. (We deviate intentionally from the point of view taken in Chapter VII in order to emphasize certain additional aspects of optimization.)

By static optimization we mean selecting a constant value u^o of the control vector giving an equilibrium state x^* of (1) lying as close as possible to x^d. Then x^* and u^o are characterized by the relations

$$x^* = \Phi x^* + \Gamma u^o \tag{2.2}$$

and

$$\|x^* - x^d\|_Q^2 = \text{minimum}.$$

(We have assumed that Φ and Γ are independent of t, because otherwise the problem would not be well defined.)

Such a u^o exists, but knowing it may not provide a complete answer to our basic problem, because the equilibrium state x^* may be unstable. Consider the scalar system

$$x_{t+1} = 2x_t + u.$$

Assume $x^d = 0$. Then u^o must be zero. But no initial state will go to zero if no control is used. Even if the initial state is $x_o = 0$, because of the instability of the system small perturbations about zero will eventually become arbitrarily large. So the equilibrium state $x^* = 0$ will not be maintained in practice. Note also that even if x^* is globally asymptotically stable, there is no assurance that the approach to x^* will be in any sense "rapid".

Hence static optimization is inadequate and we must consider dynamic optimization. We proceed to formulate the latter problem.

A performance index for the system (2.1) may be defined as:

$$(2.3) \qquad \mathcal{P}_{T,t_o}[x_{t_o}] = \sum_{t=t_o+1}^{t_o+T} [\|x_t - x^d\|^2_{Q_t} + \|u_{t-1}\|^2_{R_t}], \quad t_o,\ t,\ T = \text{integers}.$$

Usually we normalize the time scale so that $t_o = 0$ (and then we suppress t_o as an argument of $\mathcal{P}$.)

We now define our control problem to mean that the control variables u_t be selected in such a way as to minimize (2.3). This is what we shall mean by dynamic optimization.

We must find a sequence of vectors $\{u_t\}$ $t = 0, \ldots, T-1$ for which (2.3) assumes its minimum. At first sight it would appear that to obtain the optimal set $\{u_t\}$ we must optimize (2.3) with respect to all u_t simultaneously, which is an elementary but very complicated job, at least for large T.

Fortunately, we can achieve a decisive simplication by making use of the following intuitively obvious but convenient observation emphasized by R. Bellman in the 1950's (but known at least since the 17th century).

PRINCIPLE OF OPTIMALITY. An optimal sequence $\{u_t^o\}$ of control variables has the property that, whatever the initial tate x_o and the initial choice u_o^o of control vectors, the remaining terms $u_1^o,\ u_2^o, \ldots$ must constitute an optimal sequence with respect to the state x_1 resulting from the choice of u_o^o.

Using the Principle of Optimality, we can obtain various expressions for the theoretical study and practical determination of the optimal control sequence. (Methods derived from the Principle of Optimality are known by the generic name of dynamic programming. Of course, dynamic programming is simply the discrete-time counterpart of the Hamilton-Jacobi theory discussed in Sect. 5 of the preceding chapter.) The reasoning used here and the

equations obtained are for the most part but slightly different from their continuous-time counterparts in the classical calculus of variations.

We observe first that (2.3) can be written in the form:

$$ {}_{T}[x_o] = \{\|x_1 - x^d\|^2_{Q_1} + \|u_o\|^2_{R_1} + {}_{T-1}[x_1]\}. $$

Now let ${}^{o}_{T}[x_o]$ be the value of the performance index when the optimal sequence is used. Invoking the Principle of Optimality, we obtain a functional equation for ${}^{o}_{T}$:

$$ \begin{aligned} {}^{o}_{T}[x_o] &= \min_{u_o,\, u_1,\, \cdots} {}_{T}[x_o] \\ &= \min_{u_o} \{\|x_1 - x^d\|^2_{Q_1} + \|u_o\|^2_{R_1} + {}^{o}_{T-1}[x_1]\}. \end{aligned} \tag{2.5} $$

This functional equation is solved by an iterative procedure. The successive optimal (minimum) performance indices ${}^{o}_{t}$ are connected by recursion relations:

$$ \begin{aligned} {}^{o}_{1}[x_{T-1}] &= \min_{u_{T-1}} \{\|x_T - x^d\|^2_{Q_T} + \|u_{T-1}\|^2_{R_T}\}, \\ {}^{o}_{k+1}[x_{T-k-1}] &= \min_{u_{T-k-1}} \{\|x_{T-k} - x^d\|^2_{Q_{T-k}} + \|u_{T-k-1}\|^2_{R_{T-k}} + {}^{o}_{k}[x_{T-k}]\}. \end{aligned} \tag{2.6} $$

Substituting (2.1) into (2.5) we have

$$ {}^{o}_{1}[x_{T-1}] = \min_{u_{T-1}} \{\|\phi_{T-1}x_{T-1} + \Gamma_{T-1}u_{T-1} - x^d\|^2_{Q_T} + \|u_{T-1}\|^2_{R_T}\}. \tag{2.7} $$

From the theory of the pseudo-inverse, we know that the minimum of this expression is attained when

$$
\begin{aligned}
u^{o}_{T-1} &= -\left(\Gamma'_{T-1}Q_T\Gamma_{T-1} + R_T\right)^{\#}\left(\Gamma'_{T-1}Q_T(\Phi_{T-1}x_{T-1} - x^d)\right) \\
&= -K_o x_{T-1} + J_o x^d,
\end{aligned} \tag{2.8}
$$

where # denotes the Penrose-inverse of a matrix.

Substituting (2.8) into (2.7), we get

$$
{}^{o}_{1}[x_{T-1}] = \|x_{T-1}\|^2_{P_1} - 2x'_{T-1}U_1x^d + \|x^d\|^2_{S_1},
$$

where

$$
\begin{aligned}
P_1 &= (\Phi_{T-1} - \Gamma_{T-1}K_o)'Q_T(\Phi_{T-1} - \Gamma_{T-1}K_o) + K_o'R_TK_o, \\
U_1 &= -(\Phi_{T-1} - \Gamma_{T-k}K_o)'Q_T(T_{T-1}J_o - I) + K_o'R_TJ_o, \\
S_1 &= (\Gamma_{T-1}J_o - I)'Q_T)\Gamma_{T-1}J_o - I) + J_o'R_TJ_o.
\end{aligned} \tag{2.9}
$$

Now we claim that

$$
{}^{o}_{k}[x_{T-k}] = \|x_{T-k}\|^2_{P_k} - 2x'_{n-k}U_kx^d + \|x^d\|^2_{S_k}.
$$

We will prove the assertion by induction. The statement has been already shown true for $k = 1$. We assume it is true for $k - 1$. Using (2.6) we write:

$$
\begin{aligned}
{}^{o}_{k}[x_{T-k}] = \min_{u_{T-k}} \{ &\|x_{T+1-k} - x^d\|^2_{Q_{T+1-k}} + \|u_{T-k}\|^2_{R_{T+1-k}} \\
&+ \|x_{T+1-k}\|^2_{P_{k-1}} - 2x'_{n+1-k}U_{k-1}x^d + \|x^d\|^2_{S_{k-1}} \}.
\end{aligned}
$$

Applying (2.1), this becomes

$$
\begin{aligned}
{}^{o}_{k}[x_{T-k}] = \min_{u_{T-k}} \{ & \|\Phi_{T-k}x_{T-k} + \Gamma_{T-k}u_{T-k} - x^{d}\|^{2}_{Q_{T+1-k}} \\
& + \|u_{T-k}\|^{2}_{R_{T+1-k}} \\
& + \|\Phi_{T-k}x_{T-k} + \Gamma_{T-k}u_{T-k}\|^{2}_{P_{k-1}} \\
& - 2(\Phi_{T-k}x_{T-k} + \Gamma_{T-k}u_{T-k})' U_{k-1}x^{d} + \|x^{d}\|_{S_{k-1}} \}.
\end{aligned}
$$

The minimum of this expression is attained when

$$
u^{o}_{T-k} = - [\Gamma'_{T-k}(P_{k-1} + Q_{T+1-k})\Gamma_{T-k} + R_{T+1-k}]^{\#}
$$

$$
\times [\Gamma'_{T-k}(P_{k-1} + Q_{T+1-k})\Phi_{T-k}x_{T-k} - \Gamma'_{T-k}(U_{k-1} + Q_{T+1-k})x^{d}]
$$

$$
= - K_{k-1}x_{T-k} + J_{k-1}x^{d}.
$$

The fact that u^{o}_{T-k} is a linear combination of x_{T-k} and x^{d} completes the proof, and we can obtain the following recursion formulae for P_k, U_k, and S_k:

$$
\left.
\begin{aligned}
P_k &= (\Phi_{T-k} - \Gamma_{T-k}K_{k-1})'(P_{k-1} + Q_{T+1-k})(\Phi_{T-k} - \Gamma_{T-k}K_{k-1}) \\
&\qquad + K'_{k-1}R_{k-1}K_{k-1} \\
- U_k &= (\Phi_{T-k} - \Gamma_{T-k}K_{k-1})'[(P_{k-1} + Q_{T+1-k})\Gamma_{T-k}J_{k-1} - (U_{k-1} + Q_{T+1-k})] \\
&\qquad - K'_{k-1}R_{k-1}J_{k-1} \\
S_k &= (\Gamma_{T-k}J_{k-1} - I)'Q_{T+1-k}(\Gamma_{T-k}J_{k-1} - I) + J'_{k-1}R_{T+1-k}J_{k-1} \\
&\quad + S_{k-1} + J'_{k-1}\Gamma'_{T-k}P_{k-1}\Gamma_{T-k}J_{k-1} - J'_{k-1}\Gamma'_{T-k}U_{k-1} - U'_{k-1}\Gamma_{T-k}J_{k-1}.
\end{aligned}
\right\} \quad (2.10)
$$

Considerable simplification is possible after expansion and use of the pseudo-inverse lemma, $A^{\#}AA = A^{\#}$.

$$(2.11)\qquad P_k = \Phi'_{T-k}[(P_{k-1} + Q_{T+1-k}) - (P_{k-1} + Q_{T+1-k})\Gamma_{T-k}\{\Gamma'_{T-k}(P_{k+1} + Q_{T+1-k})\Gamma_{T-k} + R_{T+1-k}\}^{\#}\Gamma'_{T-k}(P_{k-1} + Q_{T+1-k})\Phi_{T-k}$$

$$(2.12)\qquad U_k = (\Phi_{T-k} - \Gamma_{T-k}K_{k-1})'(U_{k-1} + Q_{T+1-k}).$$

$$(2.13)\qquad K_k = [\Gamma'_{T-k-1}(P_k + Q_{T-k})\Gamma_{T-k-1} + R_{T-k}]^{\#}\Gamma'_{T-k-1}(P_k + Q_{T-k})\Phi_{T-k-1}$$

$$(2.14)\qquad J_k = [\Gamma'_{T-k-1}(P_k + Q_{T-k})\Gamma_{T-k-1} + R_{T-k}]^{\#}\Gamma'_{T-k-1}(U_k + Q_{n-k}).$$

From the formulae for P, and U, we see that $P_o = U_o = 0$.

Although the preceding manipulations have been rather tortuous, we now have quite explicit formulae for computing the control. We observe the important result:

Under the specified assumptions the optimal control vector is a linear (but time-dependent) function of the actual and desired states of the system.

Examination of the derivation shows that the control to be used in minimizing $\mathcal{J}_T[x_o]$ is the sequence

$$u_o^o = -K_{T-1}x_o + J_{T-1}x^d$$

$$u_1^o = -K_{T-2}x_1 + J_{T-2}x^d$$

$$\cdots$$

$$u_{T-k}^o = -K_{k-1}x_{n-k} + J_{k-1}x^d$$

$$u_{T-1}^o = -K_o x_{T-1} + J_o x^d.$$

Note that the subscripts t of x and u indicate the actual time, while the subscripts k of K, J, P, etc. are "time to go" until T.

At this point it is appropriate to discuss the very important case where we wish to minimize the performance index over an infinite number of steps.

From the derivation we see that P_1 is computed using the transition relations and weighting coefficients applicable to the last interval considered in the performance index. But this has no meaning when we are optimizing over an infinite number of intervals. Taking a practical point of view, we choose some T which in relation to the system parameters seems to be sufficiently large to allow the system to come to rest long before $t = T$. If the system is periodic over the entire range of optimization and completely controllable then we would expect that K_k would tend to be periodic as $k \to T$.

If the parameters of the system and performance index are constant, we have a much simpler situation. In this case, after using the initial control, we have exactly the same minimization problem since we must still optimize over an infinite number of steps. Thus we would expect to obtain in the limit $k \to \infty$ a constant value of the feedback matrices K and J. If the system is completely controllable and x^d is a possible equilibrium state of the system, then P_i and U_i approach limits and this expectation is fulfilled.

Under these conditions we can generalize further. Let us assume that $x^d \neq 0$ and asymptotic approach to x^d is desired. This is impossible unless x^d is in the range of $(I - \Phi)^{-1}\Gamma$. Even then if $R \neq 0$, x will not approach x^d, but a balance will be reached minimizing the sum

$$\|x - x^d\|_Q^2 + \|u\|_R^2.$$

Thus asymptotic approach to x^d implies that we must ignore the cost of control because as $x \to x^d$, $u \to \Gamma^{\#} (I - \Phi)x^d \neq 0 = \text{const.}$ It seems therefore that under certain circumstances it might be desirable to use a performance index of the form

$$\|x - x^d\|_Q^2 + \|u - u^d\|_R^2$$

where

$$u^d = \Gamma^{\#}(I - \Phi)x^d.$$

If we do this, we find that

$$u = - Kx + Jx^d + Lu^d,$$

where K and J are defined above and

$$L = [\Gamma'(P + Q)\Gamma]^{\#}\Gamma'MR$$

with

$$M = - (I - \Phi' + K'\Gamma')^{-1}\Phi'(P + Q)\Gamma[\Gamma'(P + Q)\Gamma + R]^{\#}R.$$

There are certain differences between the discrete and continuous control problems to which attention should be drawn. These follow mainly from the fact that in the discrete case R may be singular. For instance, in order to take the state of a continuous system to the origin in finite time the ordinary quadratic performance index method cannot be used; instead the "Minimum Energy" trajectory is utilized. The difference between the two riccati equations is solely a matter of terminal conditions, it is true; nevertheless special treatment must be used. In the sampled case, these problems do not arise. If s is positive definite, Q and R

zero matrices, and the system completely controllable, P will be zero after at most n (dim x = n) steps. (See Sect. 8 of previous chapter for notations.) This is the so-called "Dead Beat Control".

Physically the requirement that control be piecewise constant restricts the system response. In the continuous problem, sufficiently large control can make the state error arbitrarily small. This is no longer true in the discrete system. In the continuous system sufficiently large control can take the system to zero ("Minimum Energy Trajector") in arbitrarily small time. In the discrete system a certain number of sampling instants is required and no amount of extra control can reduce this number.

3. Computation: For the moment we will consider only the computation of P_k and K_k.

Let us define

$$P_k^* = P_k + Q_{T-k};$$

then equation (2.11) can be written

$$P_k^* = \Phi'_{T-k}[P_{k-1} - P_{k-1}\Gamma_{T-k}\{\Gamma'_{T-k}P^*_{k-1}\Gamma_{T-k} + R_{T+1-k}\}^{\#}\Gamma'_{T-k}P^*_{k-1}]\Phi_{T-k}.$$

Then by (2.13)

$$K_k = [\Gamma'_{T-k-1}P^{*T}_{k}\Gamma_{T-k-1} + R_{T-k}]^{\#}\Gamma'_{T-k}]\Gamma'_{T-k-1}P^*_k\Phi_{T-k-1}.$$

At the risk of being pedantic, we will outline very carefully the phasing of indices in P and K.

P_o is always zero.

P_1 is the value of the performance index obtained by using optimal control u^o_T over the last interval only.

$P_1^* = P_1 + Q_{T-1}$, namely P_1 plus the weighting matrix at the $(T-1)^{th}$ point.

$u_{T-1}^o = - K_o x_{T-1} + J_o x^d$ where K_o and J_o are computed using P_o^*, not P_1^*.

$P_T^* = P_T + Q_o$, where Q_o has not been defined (see (2.3)). This is purely a computational problem and merely requires that some matrix Q_o be provided for the program.

$P_k = P_k^* - Q_{T-k}$, so if the performance index matrix P_k is required, Q_{T-k} must be subtracted from the program output P_k^*.

If comparisons are made between the discrete and continuous performance indices, we must use the redefinition

$$(3.1) \qquad {}_T[x_o] = \sum_{k=1}^{T} (t_k - t_{k-1}) [\|x_k - x^d\|^2_{Q_k} + \|u_{k-1}\|^2_{R_k}].$$

Computation of U_k does not fall into the pattern given. However if K_k converges to K and U_k to U then from (1.9) we have

$$U = (\Phi - \Gamma K)'(U + Q).$$

Since we have K available we can compute U and thus J. Notice that although we need not have assumed x^d constant our program can obtain J only if U converges, which probably will not occur unless x^d is constant.

The form of the equation used in the machine is described in Chapter I, page 11 and Fig. 1.

Some transformations are required. In terms of Chapter I, Fig. 1, we let

$$F_{MACH} = \Phi'_{SYSTEM}$$

$$PO_{MACH} = P^*_{SYSTEM}$$

$$G_{MACH} = \Gamma'_{SYSTEM}$$

$$R_{MACH} = R_{SYSTEM}$$

$$Q_{MACH} = Q_{SYSTEM}$$

$$K_{MACH} = K'_{SYSTEM}.$$

Indexing of parameters in a nonstationary problem must be handled by proper programming of ASP, see, for instance, Problem C in Chapter IX.

Under certain circumstances P^* will be independent of time. A relevant theory is:

THEOREM: Let $R = 0$, $\Gamma'P^*_o\Gamma$ be nonsingular and rank $\Gamma'P^*_o\Gamma$ = rank P^*_o. Then $P^*_o\Gamma(\Gamma'P^*_o\Gamma)^{-1}\Gamma'P^*_o = P^*_o$, implying that $P^*_k = Q = P^*_o$ for all k.

Proof. $P^*_o = Q = AA'$, $\Gamma'A$ nonsingular (we make the trivial assumption that Γ has maximal rank), then

$$P^*_o\Gamma(\Gamma'P^*_o\Gamma)^{-1}\Gamma'P^*_o$$

$$= AA'\Gamma(\Gamma'AA'\Gamma)^{-1}\Gamma'AA = AA' = P^*_o = Q.$$

An important special case occurs when Γ is a vector and rank $Q\Gamma$ = rank Q.

An analogous result states:

THEOREM: If Γ is nonsingular, $P^*_k = Q$.

Proof. Consider $P^*_o - P^*_o\Gamma(\Gamma'P^*_o\Gamma)^{\#}\Gamma'P^*_o$. Since Γ is nonsingular, multiply by Γ' and Γ. Then

$$\Gamma'P^*_o\Gamma - \Gamma P^*_o\Gamma(\Gamma'P^*_o\Gamma)^{\#}\Gamma'P^*_o\Gamma;$$

by the first pseudo inverse lemma, this is the zero matrix.

All of these results are alike in saying that if k states are penalized and there are k independent control variables which can independently "reach" the k weighted states, then the performance index is zero. Clearly if $R \neq 0$, this cannot be true.

PSEUO in SAMPL does not iterate. It uses Pl which is ordinarily 10^{-2}. Therefore unless R is a scalar Pl should be changed by PIZER to something like 10^{-6} or 10^{-7}.

4. Checks: Input conforms to Chapter I notation, not to the system (2.1) described above.

$$F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad G' = \begin{bmatrix} 0 & 0 \\ 2 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Q = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 1 \\ & \\ 1 & 2 \end{bmatrix}.$$

$K(0)$, $P(1)$, $K(1)$, and $P(2)$ appear as Fig. 2, 3, 4, and 5 and are correct to the given number of places.

5. Filtering Theory: In Sect. 2 we have set up a control problem for a discrete deterministic system and derived the appropriate equations for the optimal feedback and the minimum performance index. Through the Principle of Duality we know that these same equations represent the solution of an optimal filtering problem.

For pedagogical reasons, however, we prefer to derive the equations for the optimal filter ab initio.

Consider the system

$$x_{t+1} = \Phi_t x_t + u_t$$

$$y_t = H_t x_t + v_t \tag{5.1}$$

where

t = integer

x_t = the state vector

Φ_t = transition matrix, a function of t

u_t = an element of a vector-valued independent gaussian random sequence

H_t = $(q \times p)$ output matrix, a function of t.

v_t = $(q \times 1)$ an element of a vector-valued independent gaussian random sequence.

We assume that $Eu_t = 0$, $Ev_t = 0$, $Eu_t v_{t'}' = 0$, $Eu_t u_{t'}' = Q_t \delta_{tt'}$ and $Ev_t v_{t'}' = R_t \delta_{tt'}$, where Q_t and R_t are respectively $(p \times p)$ and $(q \times q)$ matrix, functions of t; we also assume that $[\Phi', H']$ is completely controllable, i.e. (5.1) is completely observable.

The filtering problem we will consider may be stated as follows.

Given the values of the output at T times $\{y_0, y_1, \ldots, y_T\}$, determine an estimate $\hat{x}_{T+1}$ of x_{T+1} such that $E\|x_{T+1} - \hat{x}_{T+1}\|^2$ is minimum.

This statement has a formal resemblance to the usual least square fitting problem which may in turn be posed as a problem in orthogonal projections in an inner product space (see for instance the Orthogonal Projection Lemma in Chapter IV). Let us digress to show that this resemblance is more than formal.

Consider the real-valued random variables $y_0, y_1, \ldots, y_T$. The set of all real linear combinations of these form a vector space Y_T. (Remember y_t does not denote the sample obtained at t but the random variable at t

that is the set of all possible samples at t with some distribution function.) Then any linear combination of the y_t is also a random variable.

Furthermore, in this vector space, an inner product may be defined $(y_t, y_{t'}) = Ey_t y_{t'}$. But now that we are in an inner product space we can apply fourier analysis to our problem. The orthogonal projection lemma from Chapter IV tells us that $E\|x_{T+1} - \hat{x}_{T+1}\|^2$ will be minimum if $\hat{x}_{T+1}$ is the orthogonal projection of x_{T+1} on Y_T, provided we restrict $\hat{x}_{T+1}$ to be a linear functional on Y_T.

It is interesting to have the following theorem which tells us that the orthogonal projection on Y_T in the case we consider, actually gives the minimum value of $E\|x_{T+1} - \hat{x}_{T+1}\|^2$.

THEOREM: Let $\{x_i\}$, $\{y_i\}$ be random sequences with zero mean. We observe $y_0, y_1, \cdots, y_T$. If either

(A) the random sequences are gaussian; or

(B) the optimal estimate is restricted to be a linear function on Y_T and the function to be minimized is $E\|x_{T+1} - \hat{x}_{T+1}\|^2$ then the optimal estimate of x_{T+1} given $y_0, \cdots, y_T$ is the orthogonal projection of x_{T+1} on Y_T.

In the sequel, we shall be dealing mainly with vector valued random variables. In that case we have only to remember that Y_T is an $m(T+1)$ dimensional space and Y_{T+1} is $m(T+2)$ dimensional. This is actually the only novelty to the analysis -- at each step we increase by m the dimension of the space into which we are projecting. The notation to be used in this case will be explained at each step. Furthermore we will minimize not just $E\|x_{T+1} - \hat{x}_{T+1}\|^2$ but even $E(x_{T+1} - \hat{x}_{T+1})_i^2$ that is, minimize each component of the error $E\|x - \hat{x}\|^2$ separately. Let us assume that the $\{y_t\}$ have been orthonormalized to $\{\epsilon_t\}$. In R^m we mean by this that $(\epsilon_t, \epsilon_{t'}) = \delta_{tt'}$. However the j^{th} component ϵ_{tj} of ϵ_t is a random variable and it is these that must be orthonormalized, i.e. $E\epsilon_{tj}\epsilon_{t'\ell} = \delta_{tt'k\ell} = 0$ unless $i = j = k = \ell$. This is conveniently expressed by writing $E\epsilon_t\epsilon_{t'} = \delta_{tt'}I_m$.

Purely algebraically, we know that

$$\hat{x}_{T+1} = \sum_{t=0}^{T} (x_{T+1}, \epsilon_t)\epsilon_t. \tag{5.2}$$

By this formula we mean that every component of x_{T+1} is projected into Y_T

$$(\hat{x}_{T+1})_j = \sum_{t=0}^{T} ((x_{T+1})_j, \epsilon_t)\epsilon_t. \tag{5.3}$$

Since ϵ_t is an m-vector, we can write (5.2) more explicitly as

$$(\hat{x}_{T+1})_j = \sum_{t=0}^{T} \sum_{k=1}^{m} ((x_{T+1})_j, \epsilon_{tk})\epsilon_{tk} \tag{5.3'}$$

which has a clearer representation as

$$\hat{x}_{T+1} = \sum_{t=0}^{T} E(x_{T+1}\epsilon_t')\epsilon_t. \tag{5.4}$$

In other words, $\hat{x}_{T+1}$ is a random variable expressed as the sum of random variables ϵ_t with coefficients determined by the expected values $E(x_{T+1}\epsilon_t')$.

Using (5.1) in (5.2) we get

$$\begin{aligned}\hat{x}_{T+1} &= \sum_{t=0}^{T} ((\Phi_T x_T + u_T), \epsilon_t)\epsilon_t \\ &= \Phi_T \sum_{t=0}^{T} (x_T, \epsilon_t)\epsilon_t + \sum_{i=0}^{T} (u_T, \epsilon_t)\epsilon_t).\end{aligned}$$

Since $\{u_t\}$ is uncorrelated, we have $(u_T, \epsilon_t) = 0$ for $i = 0, \ldots, T$, by (5.1). Thus

$$\hat{x}_{T+1} = \Phi_T[\hat{x}_T + (x_T, \epsilon_T)\epsilon_T].$$

Now what is ϵ_T? It is the component of y_T orthogonal to Y_{T-1}, normalized. Let us observe that $(y_T - H_T\hat{x}_T)$ is orthogonal to Y_{T-1}. We know that

$$E(x_T - \hat{x}_T)y_t' = 0 \qquad t = 1, \ldots, T-1.$$

So $E(Hx_T - H\hat{x}_T)y_T' = 0$ and $E(y_T - Hx_T)y_t' = Ev_Ty_t' = 0$, $t = 1, \ldots, T-1$. And since $(y_T - H\hat{x}_T)$ lies in Y_T it must be a multiple of ϵ_T. In the usual analysis where y_T is one-dimensional, we would say a scalar multiple, but here we can say only that

$$(y_T - H\hat{x}_T) = A\epsilon_T,$$

where A is nonsingular.

We have, therefore, shown that $\hat{x}_T$ is generated in a system

$$\hat{x}_{T+1} = \Phi\hat{x}_T + K_T(y_T - H\hat{x}_T).$$

Now let us determine K_T.

We know that $\tilde{x}_{T+1} = x_{T+1} - \hat{x}_{T+1}$ is orthogonal to Y_T. From the discussion above we know that this means

$$E\tilde{x}_{T+1}y_t' = 0 \qquad \text{for} \quad t = 1, \ldots, T.$$

Hence

$$\Phi_TE\tilde{x}_Ty_t' - K_TH_TE\tilde{x}_Ty_t' + Eu_Ty_t' - K_TEv_Ty_t' = 0.$$

Again we have $(u_T, y_t) = 0$ for $t = 0, \ldots, T$. By optimality of $\hat{x}_T$, $(\tilde{x}_T, y_t) = 0$ for $t = 0, \ldots, T-1$. Because $v_Tx_t' = 0$ for all t and $v_Tv_t = R_T\delta_{tT}$, we have $Ev_Ty_t' = R_T$. So we finally arrive at

(5.5)

$$\Phi_T E\tilde{x}_T y_T' = K_T(H_T E\tilde{x}_T y_T + R_T)$$

$$E\tilde{x}_T y_T' = E\tilde{x}_T \tilde{x}_T H_T' + E\tilde{x}_T v_T + E\tilde{x}_T \hat{x}_T H_T'.$$

We know from fourier series that $(\tilde{x}_T, \hat{x}_T) = 0$.

$$E\tilde{x}_T y_T' = E\tilde{x}_T x_T H_T' + E\tilde{x}_T v_T'$$

$$E\tilde{x}_T v_T = E\tilde{x}_T v_T H_T' + Ex_T v_T'$$

$$= E\hat{x}_T v_T'.$$

But $\hat{x}_T$ is orthogonal to ϵ_T, so

$$(\hat{x}_T, (y_T - H_T\hat{x}_T)') = 0$$

$$= (\hat{x}_T, (H_T(x_T - \hat{x}_T) + v_T$$

$$= \hat{x}_T v_T' = 0.$$

Let us denote $E\tilde{x}_T\tilde{x}_T$ by P_T and attempt to find a recursion scheme for P_T.

$$P_{T+1} = \Phi_N - K_N H_N)P_N(\Phi_N - K_N H_N)' + (\Phi_N - K_N H_N)\tilde{x}_N u_N$$

$$- (\Phi_N - K_N H_N)\tilde{x}_N v_N K_N' R_N K_N'$$

$$= (\Phi_N - K_N H_N)P_N(\Phi_N - K_N H_N)' + Q_N + K_N R_N K_N'.$$

After a similar reduction we obtain

$$P_{n=1} = \Phi_N[P_N - P_N H_N'[H_N P_N H'_N + R_N]]^{\#} H_N P_N \Phi_N' + Q_N$$

$$K_N = \Phi_N P_N H_N'[H_N P_N H_N' + R_N]^{\#}$$

$$\hat{x}_1 = \Phi_o \hat{x}_o + K_o(y_o - H_o \hat{x}_o).$$

Since (5.5) must be satisfied exactly, not in the least squares sense, it is necessary to justify the use of the pseudo-inverse in the formula for K_T. The justification is that a solution always exists since

$$\text{Range } [HPH' + R] \supset \text{Range } HP$$

provided P and R are nonnegative definite symmetric. This assertion is proved in the appendix to the chapter on approximation of an impulse response.

These formulae require initial conditions P_o and $\hat{x}_o$ which may be otained for instance as follows.

By our assumption of complete observability

$$[H', \Phi'H', \Phi'^2H', \ldots, \Phi'^{n-1}H']$$

has maximal rank. Let q be the number such that

$$\Lambda' = [H', \Phi'H', \ldots, \Phi'^{q-1}H']$$

has maximal rank.

Then

$$Y = \begin{bmatrix} y_o \\ y_1 \\ \vdots \\ y^{q-1} \end{bmatrix} = \Lambda x_o + \Omega$$

where

$$\Omega = \begin{bmatrix} v_o \\ v_1 + Hu_o \\ v_2 + H\Phi u_o + Hu_1 \\ v_3 - H\Phi^2 u_o + H\Phi u_1 + Hu_2 \end{bmatrix}.$$

Our primary concern in an initial estimate $\hat{x}_o$ is that it be unbiased, i.e.

$$E(x_o - \hat{x}_o) = 0$$

$$E(x_o - KY) = E(I - K\hat{\Lambda})x_o - EK\Omega$$

$$= E(I - K\Lambda)x_o.$$

In order to guarantee that this is zero, we must choose $K = \Lambda^{\#}$. It is then possible to compute $E\tilde{x}_o\tilde{x}_o' = \Lambda^{\#}E\Omega\Omega'\Lambda^{\#'}$, and, letting

$$\hat{x}_o = \Phi^k\Lambda^{\#}Y$$

and

$$P_o = \Phi^k\Lambda^{\#}E\Omega\Omega\Lambda^{\#}\Phi'^k$$

we are ready to start our recursion scheme.

This brief analysis brings out the importance of an unbiased initial $\hat{x}_o$ since the derivation of the minimum variance estimate requires the previous $\tilde{x}$ to be unbiased.

As an example to help those who are confused by the outer products of Euclidean vectors, e.g. Exx', which appear to be inner products in the probability space, let us point out the following precise analogy.

Let $f' = (f_1, \ldots, f_n)$ be a row vector of n functions which are Lebesgue square integrable on $[0, 2\pi]$. We desire the m^{th} Fourier approximant for f. The easiest way to write the solution is by representing the coefficients of the Fourier series as an outer product:

$$\hat{f} = \frac{1}{2\pi} \left\{ \int_0^{2\pi} \frac{1}{2} f dt + \sum_{i=1}^{m} \left[\int_0^{2\pi} f(\cos it)dt, \int_0^{2\pi} f(\sin it)dt \right] \begin{bmatrix} \cos & it \\ \sin & it \end{bmatrix} \right\}.$$

The significant property here is that in both the example and the problem itself there is not merely a minimization of

$$\|\hat{f} - f\|^2 \quad \text{or} \quad E\|\hat{x} - x\|^2$$

but minimization is taking place component-wise! That is, we are actually minimizing each

$$(\hat{f}_i - f_i)^2 \quad \text{and} \quad E(\hat{x} - x)_i^2 \qquad i = 1, \ldots, n.$$

Thus each component of x is projected into the probability space and at each step more than one new dimension is being added to the space into which we are projecting. In the Fourier approximation we are projecting each component of f into the function space spanned by

$$\{\cos it, \sin it\} \qquad i = 0, \ldots, m$$

and at each step we add two to the dimension of the fitting space. (It is not universally known, but the m^{th} Fourier approximant is "best" in the least squares sense.)

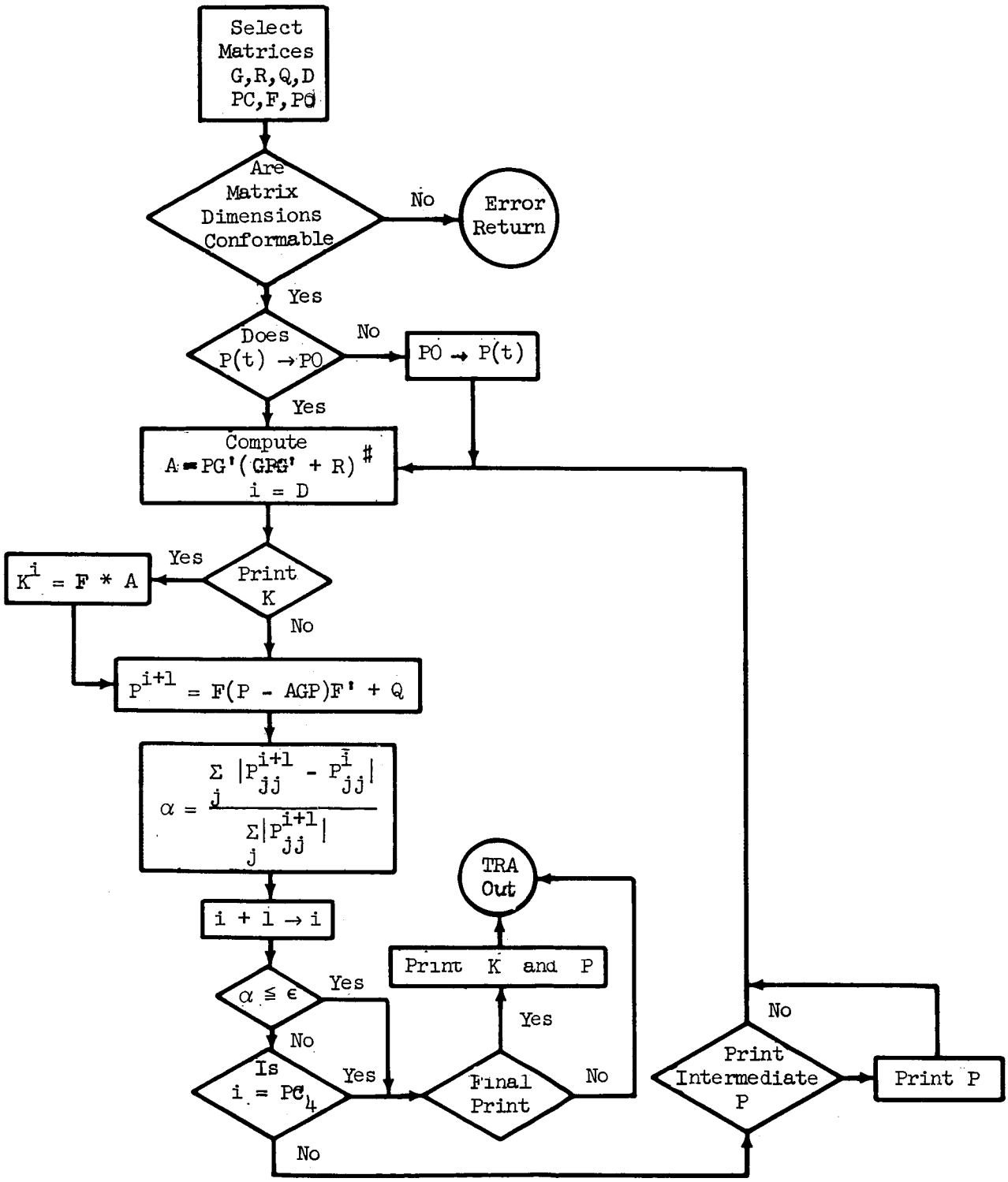
Select Matrices G,R,Q,D PC,F,P0
Are Matrix Dimensions Conformable
No
Error Return
Yes
Does P(t) → P0
No
P0 → P(t)
Yes
Compute A = PG'(GPG' + R)# i = D
Print K
Yes
K^i = F * A
No
P^{i+1} = F(P - AGP)F' + Q
α = Σ_j |P^{i+1}_jj - P^i_jj| / Σ_j |P^{i+1}_jj|
i + 1 → i
α ≦ ε
Yes
No
Is i = PC4
Yes
No
Final Print
Yes
Print K and P
TRA Out
Print Intermediate P
No
Print P

Fig. 1

MATRIX K(0)

NUMBER OF ROWS 3 NUMBER OF COLUMNS 2

EXPONENT -1

4.2857 -1.4286

0.0000 0.0000

-1.4286 7.1429

Fig. 2

MATRIX P(1)

TAU = 0.1000E 01 EPSILON = 0.

NUMBER OF ROWS 3 NUMBER OF COLUMNS 3

EXPONENT 0

3.1429	1.0000	0.2857
1.0000	1.0000	0.0000
0.2857	0.0000	3.5714

ALPHA = 0.35185185E-00

Fig. 3

MATRIX K(1)

NUMBER OF ROWS 3 NUMBER OF COLUMNS 2

EXPONENT 0

0.4149 -0.0745
0.0000 0.0000
-0.2660 1.3298

Fig. 4

MATRIX P(2)

TAU = 0.1000E 01 EPSILON = 0.

NUMBER OF ROWS 3 NUMBER OF COLUMNS 3

EXPONENT 0

3.1702	1.0000	0.5319
1.0000	1.0000	0.0000
0.5319	0.0000	5.7872

ALPHA = 0.22527475E-00

Fig. 5

CHAPTER IX

COMPUTATION OF TIME HISTORIES

1. Description of the Problem: The entire manual is concerned with finite dimensional linear dynamical systems. That is, processes determined by linear differential or difference equations. This being the case, one of the most important problems is to obtain the time history of the state.

As we have seen in Chapter II, this involves computation of the transition matrix -- ordinarily a very straightforward process. In this chapter we shall consider several numerical difficulties which can arise. In general these belong to two categories: poor conditioning and non-stationarity.

ASP has two formats for displaying transient behavior. We will illustrate both of these and discuss their respective virtues.

2. Theory and References: Theoretical background for this material is contained in Chapter II and Acrivos, see 3.B) below. Acrivos is concerned with discrete systems but in the digital computer we are always constrained to discretize so that much of this material is pertinent to both continuous and discrete systems.

We shall now consider several distinct problems (A, B, C below) to familiarize the reader with techniques which are repeatedly used in later applications.

3A. The Specific Problem: We begin by computing the transition matrix of

$$(3A.1) \qquad \dot{x} = \begin{bmatrix} 0 & \frac{\pi}{10} \\ -\frac{\pi}{10} & 0 \end{bmatrix} x, \quad \Phi(t) = \begin{bmatrix} \cos \frac{\pi}{10}t & \sin \frac{\pi}{10}t \\ -\sin \frac{\pi}{10}t & \cos \frac{\pi}{10}t \end{bmatrix}.$$

In the computer we have no method of functional expression, so we must compute Φ at selected points in time. For convenience we ask for printout at fixed intervals, say, $\tau = .5$.

4A. Equations and Procedure: We will compute

$$\Phi(.5) = e^{.5F}$$

and then use the relationship

$$\Phi(.5k) = \Phi^{k}(.5).$$

5A. Results: The ASP program used to compute such a transiton matrix may be seen in Fig. 1. Results are good, as would be expected in such a trivial problem. The errors are larger than can be legimately blamed on the fact that $\pi/10$ can not be accurately entered in the machine. For instance, the error in sine at $k = 10$, where the argument of trigonometric terms should be $\pi/2$, is $.24 \cdot 10^{-6}$, but the errors in $\pi/10$ should be less than $1 \cdot 10^{-8}$ and the derivative of sine at this point is nearly zero. So it appears that the errors are largely accumulated computational errors.

The rather large sampling period makes it difficult to check phase precisely. Nevertheless it is possible to say that these are genuine errors in the computation of Φ rather than phase errors caused by incorrect values of the argument. In the first place 0.5 enters the machine exactly. Secondly, the errors increase linearly with time and the value of the determinant decreases linearly. If a phase error were contributing the cosine term would not have errors increasing linearly with time and the determinant would remain constant. Notice that there is an extremely small phase error as measured by the sine terms being nonzero.

t	Φ	element error	determinant error
5	$\begin{bmatrix} .5588.10^{-6} & .99999976 \\ -.99999976 & .5588\cdot10^{-6} \end{bmatrix}$	$.24\cdot10^{-6}$	$.5\cdot10^{-6}$
50	$\begin{bmatrix} -.99999756 & .5476\cdot10^{-6} \\ -.5476\cdot10^{-6} & -.99999756 \end{bmatrix}$	$2.43\cdot10^{-6}$	$4.8\cdot10^{-6}$
100	$\begin{bmatrix} .99999513 & -.1051\cdot10^{-5} \\ .1051\cdot10^{-5} & .99999513 \end{bmatrix}$	$4.87\cdot10^{-6}$	$9.7\cdot10^{-6}$
200	$\begin{bmatrix} .99999024 & -.2023\cdot10^{-5} \\ .2023\cdot10^{-5} & .99999513 \end{bmatrix}$	$9.75\cdot10^{-6}$	$19.5\cdot10^{-6}$.

One interesting aspect of the computation was that throughout this fairly long transient the numbers which should have been equal were equal.

6A. Numerical Considerations: Observe that the previous F matrix was in good form for computation. If we take the same problem and change only the units in which the components of x are expressed, we could obtain the mathematically identical system

$$\dot{y} = \begin{bmatrix} 0 & \frac{\pi}{10} \cdot 10^{6} \\ -\frac{\pi}{10} \cdot 10^{-6} & 0 \end{bmatrix} y, \quad \Phi(t) = \begin{bmatrix} \cos \frac{\pi}{10} t & 10^{6} \sin \frac{\pi}{10} t \\ -10^{-6} \sin \frac{\pi}{10} t & \cos \frac{\pi}{10} t \end{bmatrix}.$$

In this case $.5 \|F\| > 10$ so τ will be halved until $\frac{.5}{2^k} \|F\| < 10$ (see Chapter II) and the exponential squared k times. In this example k was 14 and τ was $\frac{.5}{16,384} \approx 3 \cdot 10^{-5}$. Because of this all the significance was contained in the sine term since the cosine to eight decimal places was one. This seriously affected the accuracy. For one thing notice that terms which should be equal are not precisely so.

t	Φ	element errors	determinant errors
5	$\begin{bmatrix} .2608\cdot10^{-7} & .99770785\cdot10^{6} \\ -.99770787\cdot10^{-6} & .3353\cdot10^{-7} \end{bmatrix}$	$.22\cdot10^{-2}$	$.44\cdot10^{-2}$
50	$\begin{bmatrix} -.97731353 & .2617 \\ -.2647\cdot10^{-12} & -.97731354 \end{bmatrix}$	$2.2\cdot10^{-2}$	$4.4\cdot10^{-2}$
100	$\begin{bmatrix} .95514177 & -.5332 \\ .5542\cdot10^{-12} & .94414181 \end{bmatrix}$	$4.4\cdot10^{-2}$	$8.8\cdot10^{-2}$
200	$\begin{bmatrix} .91229575 & -1.123 \\ 1.135\cdot10^{-12} & .91229574 \end{bmatrix}$	$8.8\cdot10^{-2}$	$17.6\cdot10^{-2}$

Not only does this cause loss of accuracy but it also wastes time in the exponential routine. This problem is a very serious numerical one, it is difficult to tell whether a number is small because it expresses an unimportant coupling or because the system is given in a poorly chosen basis.

In this case $(\frac{\pi}{10}) \cdot 10^{-6}$ is vital to the system although it is much smaller than the other matrix elements. Usually we will try to equalize elements as much as possible by using diagonal transformations. No more complicated transformations are attempted as a rule.

7A. Remarks: In this simple case where we know the transition matrix, we can consider the problem to be discrete and load $\Phi(.5)$ into the machine. Clearly this is the most machine-efficient process as it avoids the exponential computation. But it means also that the programmer must compute sine $\frac{\pi}{20}$. Since the last step is laborious, it is better to load F into the machine and let it compute $\Phi(.5)$.

Another decision to be made when computing time histories is whether or not to use TRNSI. TRNSI cannot be used with nonstationary systems. If a complete homogeneous solution is required TRNSI should not be used; it is inefficient. If a complete solution, including control variables is required TRNSI should not be used if $n > 7$. In fact the use of TRNSI should be confined to the case where only one or two initial vectors are to be considered and the number of outputs is less than seven. A drawback of TRNSI is that it has only a four-decimal-place print format; if the system is well-conditioned this should not be a serious objection. In programmed transients, however, we can compute in a well-conditioned basis and retransform for printing with no accumulating error due to poor conditioning. Clearly this cannot be done in TRNSI.

To cut down on the paper volume of output, we did not use the Fig. 1 ASP Program, but instead that appearing in Fig. 2. This presents the output in a "packed" form which uses the output routine, and hence writes a new page less frequently. Since Fortran uses blocked output, there is no significant saving of machine time, and it makes output somewhat more difficult to read, but the factor of 10 decrease in pages is worth come inconvenience.

3B. The Specific Problem: As a second example let us consider a system given by Andreas Acrivos, "The Transient Response of Stagewise Processes", Jour. SIAM, 4 (March 1956):

$$(3B.1) \qquad \dot{x} = \begin{bmatrix} 0 & 435\cdot21 & 355\cdot54 \\ -2.0892\cdot10^{-2} & -2.4507 & 0.6577 \\ -0.0386\cdot10^{-2} & 1.4507 & -2.7634 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u.$$

Notice that the units (of physical quantities) have been poorly chosen. We can easily transform to the system

$$(3B.2) \qquad \dot{y} = \begin{bmatrix} 0 & 4.3521 & 3.5554 \\ -2.0892 & -24507 & 0.6577 \\ -0.0386 & 1.4507 & -2.7634 \end{bmatrix} y + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u.$$

using the transformation

$$(3B.3) \qquad y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix} x.$$

We will run a transient response of the system to $u = 10.$ in both the transformed and untransformed system to analyze the effects of the transformation. In addition we will obtain a few points of the frequency responseϕ using the transformed system (3B.2).

The eigenvalues are given as

$$\lambda_1 = -3.7778$$

$$\lambda_{2,3} = -0.7180 \pm i3.0179$$

making the shortest time constant about .27 and the largest about 1.4.

Let us use a print interval then of .05 and a final time of 7.5.

4B. Equations and Procedure: To obtain the transient response, we let $\Phi = \Phi(.05)$,

$$\Gamma = \int_0^{0.05} \Phi(\tau)d\tau \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

and compute

$$x(.05(k+1)) = \Phi x(.05k) + \Gamma u.$$

To obtain a point on the frequency response, we enlarge the system by a harmonic oscillator feeding into x_1 and run the free motion. After all transients have decayed, the amplitude and phase may be calculated.

$$F = \begin{bmatrix} 0 & 4.3521 & 3.5554 & 1 & 0 \\ -2.0802 & -2.4507 & 0.6577 & 0 & 0 \\ -0.0386 & 1.4507 & -2.7634 & 0 & 0 \\ 0 & 0 & 0 & 0 & \omega \\ 0 & 0 & 0 & -\omega & 0 \end{bmatrix}$$

The program used to do this appears in Fig. 4.

The transient response could also have been computed (probably a little less efficiently) by this same method, using a fourth order system (adding an eigenvalue = 0).

5B. Results: If we consider only the particular solution, this appears to be an ideal situation in which to use TRNSI. We need only one vector solution of three components and the solution provided by Acrivos is accurate

only to three significant figures. However, we wish to compare results obtained for systems (which is (3B.1) and (3B.2), so we will use the program in Fig. 3 (which is virtually the same as that in Fig. 2) in order to obtain more significant figures in the output.

There was very little difference in the computation using scaled and unscaled values, both were accurate to the number of significant figures given by Acrivos. At $t = 7.75$ we had:

x transformed	x untransformed
1.5908521	1.5908545
-1.5968528	$-1.5968528 \cdot 10^{-2}$
-0.86253438	$-0.86253363 \cdot 10^{-2}$.

The correct results are (transformed):

$$x = \begin{bmatrix} 1.600 \\ -1.595 \\ -0.860 \end{bmatrix} - \begin{bmatrix} 5.641 \cdot 10^{-2} \\ 0.300 \\ -0.427 \end{bmatrix} e^{-3.7778t}$$

$$+ e^{-.718t} \begin{bmatrix} 1.544 \\ -1.895 \\ -0.433 \end{bmatrix} \cos 3.0179t + \begin{bmatrix} 2.876 \\ 0.075 \\ 0.637 \end{bmatrix} \sin 3.0179t$$

The computed response of x_1 is graphed in Fig. 5, where the largest time constant (1.39) and the frequency may be checked easily.

The transfer function for x_1 is $\dfrac{s^2 + 5.2141s + 5.81814}{(s + 3.7778)((s + 0.718)^2 + (3.0179)^2)}$

which aids in checking the frequency response. The initial response to an input of $\sin t$ is graphed in Fig. 6. From the transfer function we compute the amplitude to be .207 with a phase lead of about .40 radians. The digital output after only one full cycle of the input indicates an amplitude of .208 and a phase lead of .40 radians.

As in the transient response, we see here the prime virtue of the computer with no intellectual effort and for modest computer charges we can obtain quite reasonably accurate quantitative information about the system response, such as overshoot and terminal value in the transient, phase shift and amplitude in the frequency response -- even for a much larger system than one for which we could practically carry out the required hand computation.

It hardly seems necessary here to illustrate the use of ASP in obtaining the time history of a discrete system because, as we have seen, all systems are discretized before consideration by the computer. A system which is initially discrete deletes the exponentiation operation but otherwise requires the same treatment.

3C. The Specific Problem: It is advisable however, to consider also a time-varying system. Probably the simplest and most easily checked is the Bessel function $J_0(t)$. This function is generated by the equation

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -1 & -\frac{1}{t} \end{bmatrix} x;$$

if $x_1(0) = 1$, $x_2(0) = 0$, then $x_1(t) = J_0(t)$. To avoid the pole at $t = 0$ we will begin the trajectory at $t = .1$, where $x_1 = .99750623$, $x_2 = -.04975034$.

We made this run for several meshes on the F matrix to show how the accuracy improves as the mesh becomes smaller. Accuracy will be checked against the Jahnke-Emde four decimal place tables.

First we set up a t mesh such that in any interval, the sample used does not differ by more than k percent in norm from the actual F.

This is most simply done by letting ASP compute the system matrices. Fig. 7 shows the program used to compute the F matrices and run the transient when $k = 20$.

The trajectory was run for $k = 1$, 10, and 20. The results for $k = 10$ and 20, along with J_o, are graphed in Fig. 8. The graph for $k = 1$ was indistinguishable from J_o.

```
       BEGIN
       LOAD    T,Z1,ND, I, F,
       ETPHI   F, T,PH,
       RINT   Z1,     I,PHI
HEAD 1MULT    PH, I,     1,
       ADD     T,Z1,    Z1,
       RINT   Z1,     1,PHI
       EQUAT   1, I,
       IF     ND,Z1,HEAD 1
       END
     T         1       1
              .5
    Z1         1       1
               0
    ND         1       1
           201.
     I         2       2
              1.                  0                  0                  1.
     F         2       2
               0      .3141592654       -.3141592654       0
```

FIG. 1

```
        BEGIN
        LOAD    T,Z1,ND,PR,  I,  F,
        ETPHI   F,  T,PH,
        RINT    Z1,      I,PHI
HEAD    1MULT   PH,  I,      1,
        MULT    PH,  1       2,
        MULT    PH,  2       3,
        MULT    PH,  3       4,
        MULT    PH,  4       5,
        MULT    PH,  5       6,
        MULT    PH,  6       7,
        MULT    PH,  7       8,
        MULT    PH,  8       9,
        MULT    PH,  9       I,
        JUXTR    1,  2,     A1,
        JUXTR   A1   3,     A2,
        JUXTR   A2   4,     A3,
        JUXTR   A3   5,     A4,
        JUXTR   A4   6,     A5,
        JUXTR   A5   7,     A6,
        JUXTR   A6   8,     A7,
        JUXTR   A7   9,     A8,
        JUXTR   A8   I,     A9,
        ADD     Z1,PR,      Z1,
        RINT    Z1,     A9,PHI
        IF      ND,Z1,HEAD 1
        END
             1       1
          .5
             1       1
            0
             1       1
          201.
             1       1
             5.
             2       2
            1.          0                   0                  1.
             2       2
         0           .3141592654         -.314159264            0
```

FIG. 2

```
      BEGIN
      LOAD    F, G, T,ND,PR,Z1, I,
HEAD 2EAT     F, T,PH,IN,PRINT
      RINT   PH,PHI IN,INT
      MULT   IN, G,    GM,
      RINT   Z1,      I,PHI
HEAD 1MULT   PH, I,     1
      ADD     1,GM,     1
      MULT   PH, 1,     2
      ADD     2,GM,     2
      JUXTR   1, 2,     3
      MULT   PH, 2,     1
      ADD     1,GM,     1
      JUXTR   3, 1,     4
      MULT   PH, 1,     2
      ADD     2,GM,     2
      JUXTR   4, 2,     5
      MULT   PH, 2,     1
      ADD     1,GM,     1
      JUXTR   5, 1,     6
      ADD    Z1,PR,    Z1
      RINT   Z1,      6,PHI
      EQUAT   I, I
      IF     ND,Z1,HEAD 1
      END
   F        3       3
            0        435.21          355.54          -.020892        -2.4507
      .6577          -.000386        1.4507          -2.7634
   G        3       1
         10.         0               0
   T        1       1
         .05
  ND        1       1
         7.5
  PR        1       1
          .25
  Z1        1       1
            0
   1        3       1
            0       0               0
```

FIG. 3

```
      BEGIN
      LOAD    F, T,X0,PR,Z1,ND,
      ETPHI   F, T,PH,
      RINT   Z1,    X0,X0
HEAD 1MULT   PH,X0,     1,
      MULT   PH, 1,     2,
      JUXTR   1, 2      3
      MULT   PH, 2      1
      JUXTR   3  1      4
      MULT   PH, 1,    X0,
      JUXTR   4,X0      5
      ADD    Z1 PR     Z1,
      RINT   Z1,     5, X
      IF     ND,Z1,HEAD 1
      END
 F          5      5
              0          4.3521      3.5554              1.          0
      -2.0892          -2.4507        .6577              0           0
      -.0386         1.4507          -2.7634              0           0
                0                0              0          0          1.
                0                0              0         -1.         0
 T          1      1
           .1
 X0         5      1
              0                0              0          0          1.
 PR         1      1
             .4
 Z1         1      1
              0
 ND         1      1
           7.5
```

FIG. 4

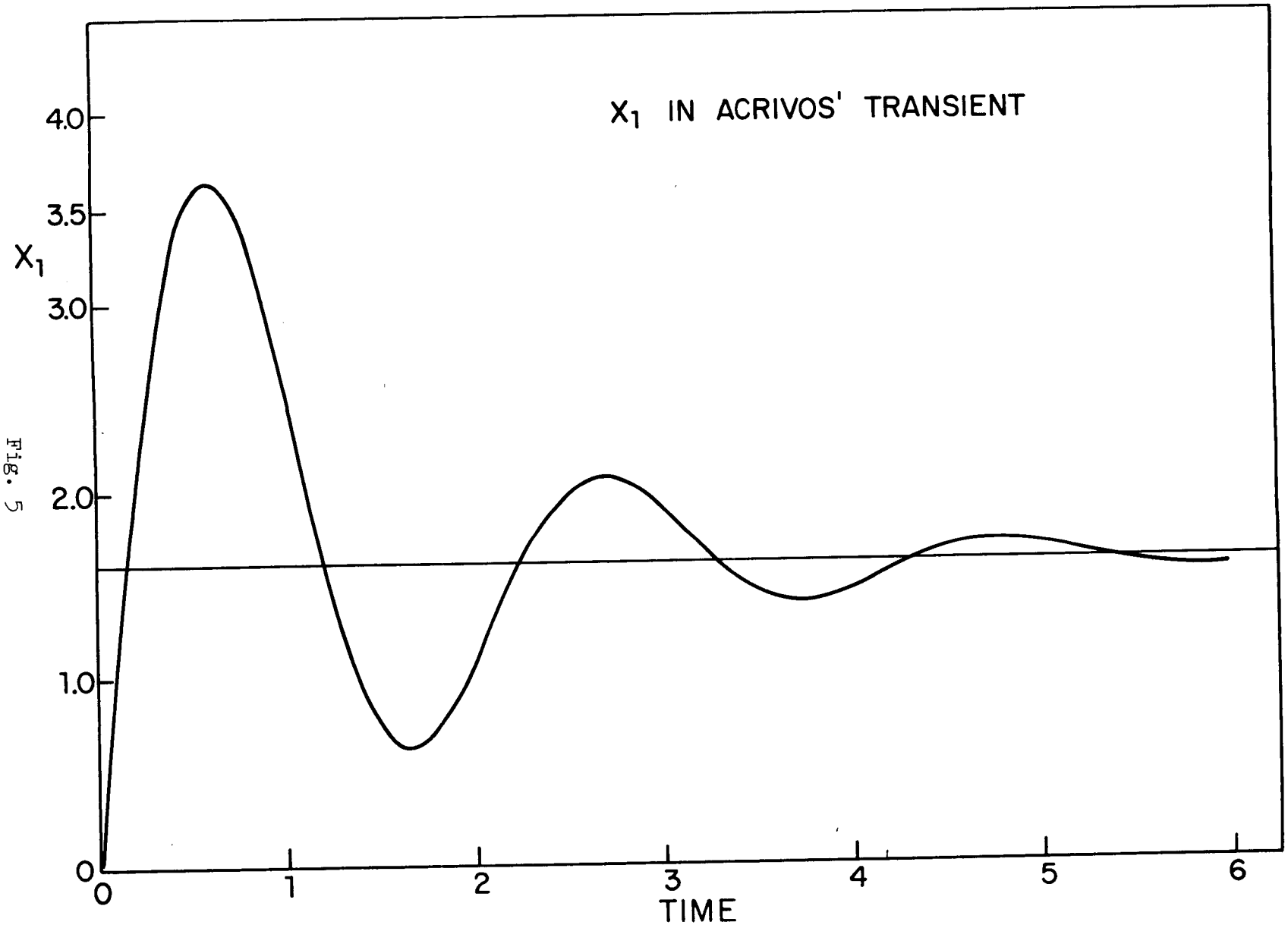

Fig. 5

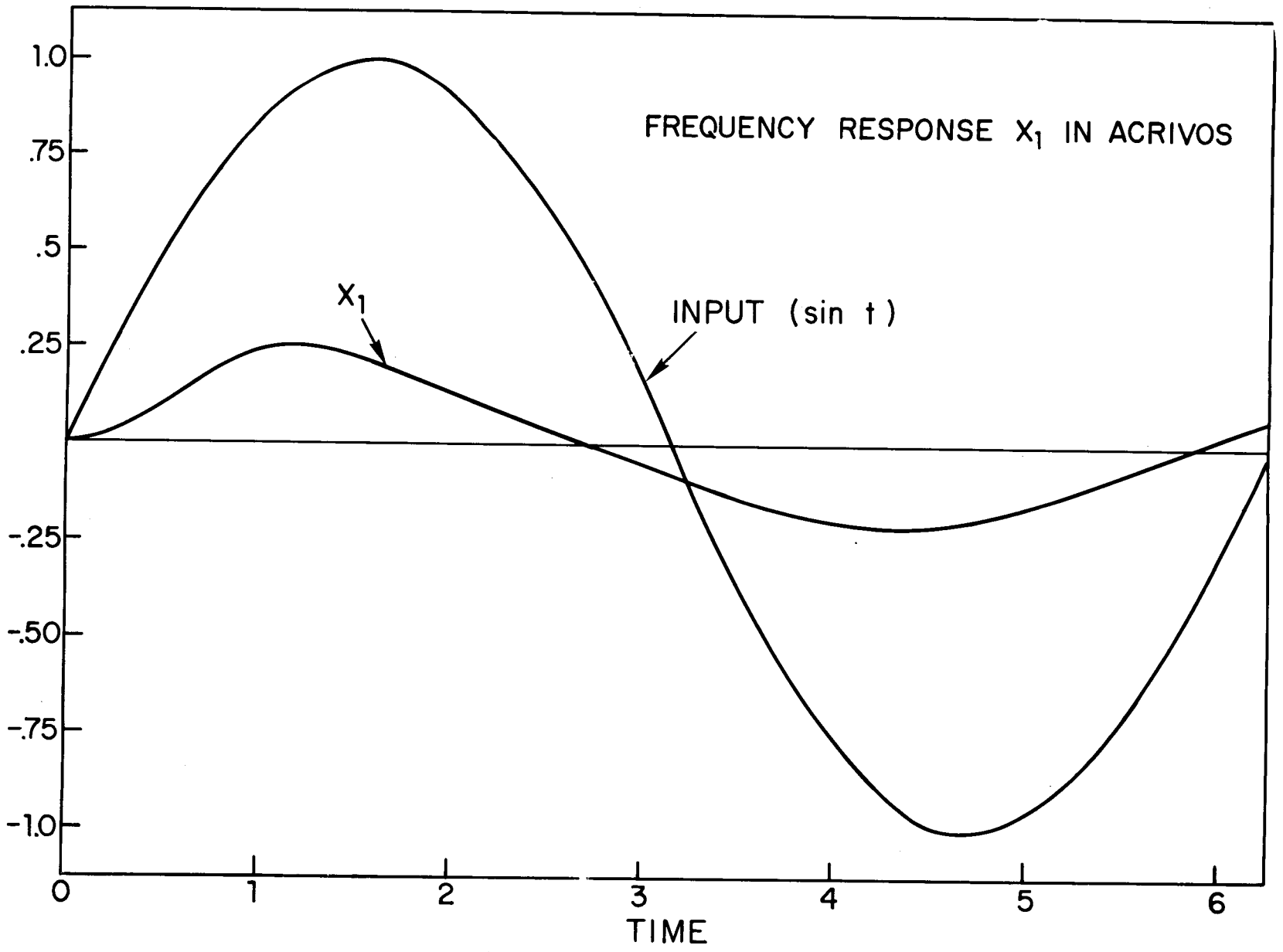
FREQUENCY RESPONSE X_1 IN ACRIVOS
X_1
INPUT (sin t)
1.0
.75
.5
.25
-.25
-.50
-.75
-1.0
0
1
2
3
4
5
6
TIME

Fig. 6

```
      NOV 17,1964
      BEGIN
      LOAD   .5,TA,TO,T1,S1,S2,BE,ND,X0,M1,M3,F1,35,FI,EC
      RINT   X0,X0
      EQUAT  TO,  T,T1,T2
      ADD    TO,TA,    PT,
HEAD4 IF     T2,FI,HEAD6
      MULT   S1,BE,    D1,
      ADD    D1,S2,    D1,
      SUBT   BE,S2,    D2,
      MULT   BE,D2,    D3,
      PSEUO  D1,+   D1,RK,
      MULT   D1,D3,    BE,
      EQUAT  T2,T1,F1,  F,
      IF     BE,M1,HEAD9
      EQUAT  EC,BE,FI,T2
      IF     .5,.5,HEAD10
HEAD6 EQUAT  F1,  F,35,T5
      IF     .5,.5,HEAD3
HEAD9 PSEUO  BE,+   T2,RK
HEAD10MULT   BE,M1,    M2
      ADD    M2,M3,    F1
      ADD    T1,T2,    T1
      MULT   .5,T1,    T5
HEAD3 IF     T5,PT,HEAD1
      SUBT   T5,  T,    T3,
      EQUAT  T5,  T,
      IF     .5,.5,HEAD2
HEAD1 SUBT   PT,  T,    T3
      EQUAT  PT,  T,
      ADD    PT,TA,    PT
HEAD2 ETPHI   F,T3,PH,
      MULT   PH,XO,    X1,
      EQUAT  X1,XO,
      IF      T,T5,HEAD4
      RINT    T,    XO,  X
      IF      T,ND,HEAD5 HEAD3
HEAD5 END
            1      1
             .5
            1      1
             .2
            1      1
             .1
            1      1
 .128205128
            1      1
            1.4
            1      1
             .4
            1      1
            7.8
            1      1
```

Fig. 7

```
          6.
          2         2
.99750623           .049750934      -.049750934      .99750623
          2         2
          0                   0              0            -1.
          2         2
          0                  1.             -1.             0
          2         2
          0                  1.             -1.          -7.8
          1         1
        1.E30
          1         1
          5.
          1         1
          .2
```

FIG. 7

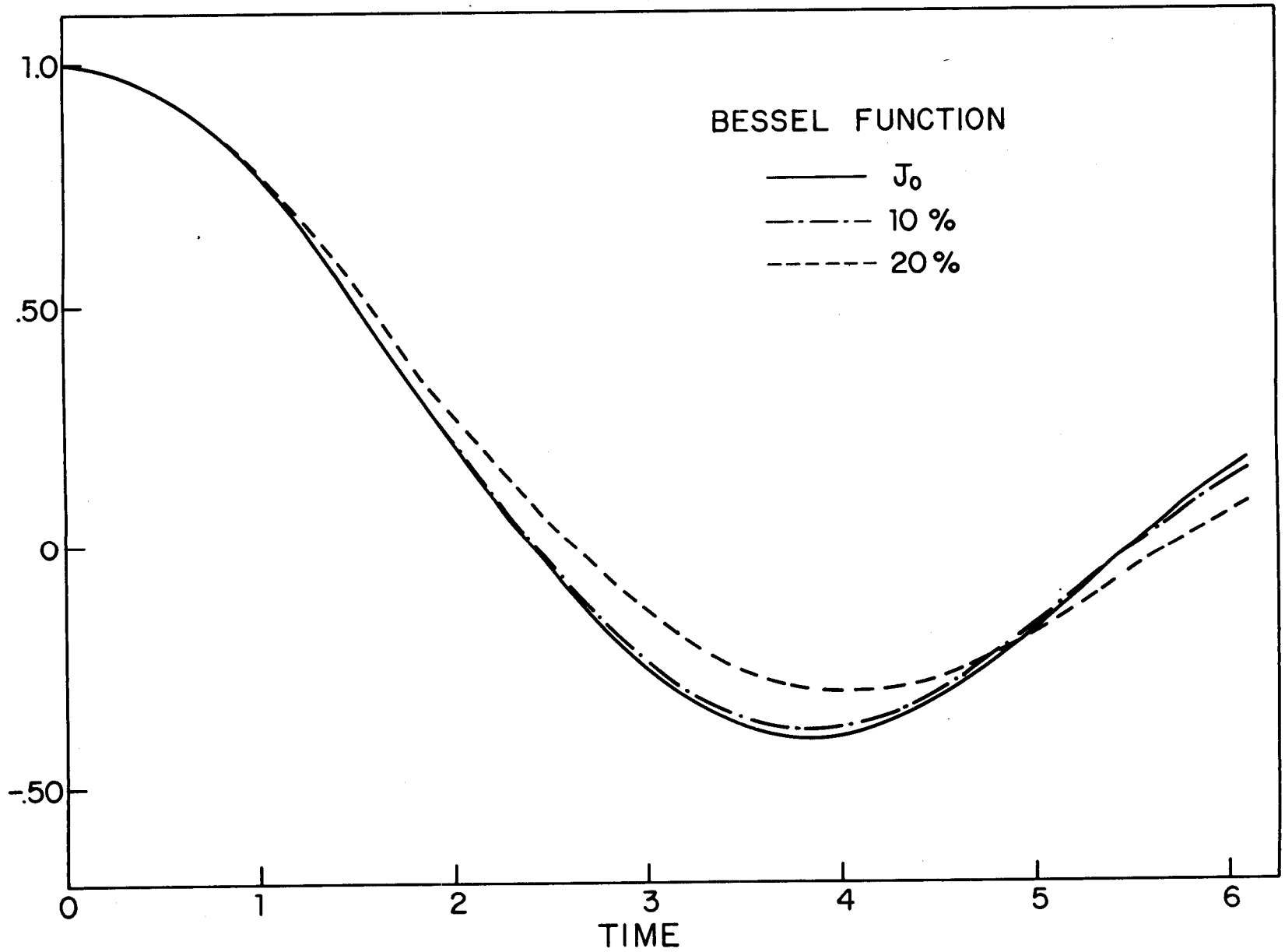

Fig. 8

CHAPTER X

STABILITY COMPUTATIONS

1. Description of the Problem: In the analysis of dynamical systems, one of the most fundamental questions is whether or not a given system is asymptotically stable. For a linear system in which the characteristic polynomial is readily available, the Routh-Hurwitz criterion is quite satisfactory for hand calculation. With the advent of Lyapunov techniques there is a desire to obtain not only information about stability, but also a Lyapunov function. In this chapter we will demonstrate some ASP methods for doing this.

2. Theory and References: For Lyapunov stability theory and definitions of stability see

[1] R. E. Kalman and J. E. Bertram", Control System Analysis and Design via the "Second Method" of Lyapunov; Journal of Basic Engineering, June, 1960.

[2] H. S. Wall, "Polynomials Whose Zeroes have Negative Real Parts", American Mathematical Monthly, 52, (1945) No. 6.

[3] Anthony Ralston, " A symmetric Matrix Formulation of the Hurwitz-routh Stability Criterion", IEEE Transactions on Automatic Control, July 1963.

[4] P. C. Parks, "Lyapunov and the Schur-Cohn Stability Criterion", IEE Transactions on Automatic Control, January 1964.

A different proof of the results of Ralston and Parks is given in

[5] R. E. Kalman, "On the Hermite-Fujiwara Theorem in Stability Theory", Quarterly of Applied Mathematics, 1965.

3. The Specific Problem: Given a continuous dynamical system

$$\dot{x} = Fx \tag{3.1}$$

or a discrete dynamical system

$$x_{k+1} = \Phi x_k. \tag{3.2}$$

We wish to determine 1) whether the system is asymptotically stable; 2) if so a Lyapunov function; 3) from this Lyapunov function we want to gain some idea of the relative stability of different state variables.

ASP has one very simple method of determining stability. Given a discrete system (3.2), perhaps obtained from a continuous system (3.1) by $\Phi = e^{\tau F}$, we need only raise Φ to very high powers, and see if its norm is, in the limit, increasing or decreasing.

This satisfies in a direct way requirement 1), but that is all.

4. Equations and Procedure: To obtain a Lyapunov function V for a linear stationary system (3.1) we must find a symmetric matrix P and a positive definite matrix Q such that

$$PF + F'P = - Q \tag{4.1}$$

or for the system (3.2) such that

$$\Phi'P\Phi - P = - Q. \tag{4.2}$$

The usual procedure is to select a positive definite matrix Q and find the corresponding P. Unless F has an eigenvalue with zero real part or, correspondingly, Φ has an eigenvalue of unit magnitude, there exists a unique solution to these problems for all Q. Then the system is stable if P is positive definite.

There exists a straightforward matrix inversion method of solving these equations (Bellman, INTRODUCTION TO MATRIX ANALYSIS, p. 231) but setting up the required matrix by means of ASP would be extremely tedious and

Moreover, the matrix would be of order $(\frac{n^2 + n}{2})$ where F is of order n. If n is 8, this would require the inversion of a rather well-populated matrix of order 36; a task which, besides being beyond the formal limits of ASP is likely to involve considerable numerical difficulties.

The method which we will use to obtain P relies upon the riccati and sampled riccati equations to find the equilibrium points of the equations

(4.3a) $$\dot{P} = F'P + PF + Q$$

and

(4.3b) $$P_{k+1} = \Phi' P_k \Phi + Q.$$

If the equilibrium points are reached, they represent the required solutions of the algebraic equations. The principal drawback of this scheme is that an equilibrium point will be reached only if the system is asymptotically stable. This precludes determining the number of unstable roots of F by determining the number of negative roots of P. However, it provides a very quick test to see if the system is unstable, for in such a case, the riccati equation does not converge. This will be illustrated in what follows.

We will apply our procedures to a continuous system with eigenvalues $(-1, -2, -3 \pm 2i, -3 \pm 4i, -4, -5)$. The system will appear in four forms: 1) continuous, with F in companion form; 2) discrete, with Φ in companion form; 3) continuous, with F in "scrambled" form obtained by a similarity transformation; 4) discrete, with Φ the exponential of F in the 3) form.

For 1) and 2) we shall obtain Lyapunov functions by the methods described in [3] and [4]. This will provide some check on the accuracy of the method. For 3) and 4) we shall use the identity matrix for Q and obtain a matrix P. This will enable us to give some statements about relative stability of the various components, as follows.

When obtaining P according to (4.1), we have that

$$x'(0)Px(0) = \int_0^\infty x'(t)Qx(t)dt \tag{4.4}$$

and using (4.2)

$$x_o'Px_o = \sum_{k=1}^{\infty} x'_k Qx_k. \tag{4.5}$$

Therefore the diagonal terms of P will tell us the value of the integral and sum (4.3) and (4.4) when x_o is in turn each vector of the usual orthonormal basis. If $Q = I$, then larger terms correspond to components with larger time constants.

5. Results: This material will provide one of the very best illustrations of the fact that diagonal and scalar transformations, easily derived and applied, are essential to the successful use of ASP (see section 6 of this chapter).

The characteristic polynomial for the eigenvalues we have chosen is

$$s^8 + 23s^7 + 267s^6 + 1782s^5 + 7663s^4 + 21324s^3 + 36669s^2 + 34470s + 13000. \tag{5.1}$$

The matrix Q corresponding to this polynomial for the special Lyapunov function given by Parks is twice the outer product of the vector

$$a' = (0,\ 34470,\ 0,\ 21324,\ 0,\ 1782,\ 0,\ 24)$$

with itself. That is, $Q = 2aa'$. Note that a' consists of every second coefficient in (5.1). The corresponding P appears in Fig. 1.

The entire system was transformed by T, $F^* = TFT^{-1}$, where $T = \text{diag}\ (10^3,\ 10^3,\ 10^3,\ 10^3,\ 10^3,\ 10^2,\ 10,\ 1)$ and the resulting F and Q used in the program appearing in Fig. 2. This gave the very reasonably

accurate P appearing in Fig. 3, with a maximum error of about 3 in the sixth significant figure. This matrix had rank eight with a very high confidence level. That is (see Chapter V) with $P_2 = 1$, we had the following table of ρ and rank

rank	ρ
6	$.29.10^{-2}$
7	$.85.10^{-3}$
8	$.11.10^{-5}$

One aspect of this computation is that the zeroes appearing in P do not have identically zero derivatives, hence they must decay to zero and their accuracy can never be better than that of the larger terms. This of course is not peculiar to this method of computation. The same inaccuracies would occur in the inversion technique.

To obtain Φ in companion form we need to select some time interval τ over which to compute $e^{\tau F}$. Having arbitrarily selected $\tau = .25$ we have the problem of computing the characteristic polynomial. It would not have been unreasonable to compute it by hand, but after all we should use ASP, so we decided to compute the characteristic coefficients as the components of the solution of

$$[\Gamma, \Phi\Gamma, \Phi^2\Gamma, \ldots, \Phi^7\Gamma]x = \Phi^8\Gamma$$

where Γ is some vector such that $[\Phi, \Gamma]$ is completely controllable.* Even then we had the choice of what form to use for F. Feeling that computing Φ from F in companion form would perhaps provide the more difficult problem, we decided to do that; so we let $\Gamma' = (0,0,0,0,0,0,0,1)$ and $\Phi = e^{.2\,F}$. Note that (Φ,Γ) is completely controllable because (F,Γ) is.

---

*This method of computing the coefficients of the characteristic equation of a matrix goes back to the celebrated Russian applied mathematician (Krylov).

The only check which we performed on these calculations was to compare the first and last coefficients obtained in this way with the determinant of Φ as computed analytically ($e^{-6.} = .0024787524$) and with the trace of Φ as computed by the machine (3.37923825).

The characteristic polynomial as computed was

$$s^8 + 3.37923745^7 + 5.0860058s^6 + 4.5157148s^5 + 2.6094069s^4 + 1.0075931s^3 + .25321779s^2 + 0.37539943s + .0024787312,$$

which agrees very well in the two given coefficients. Assuming that we were working with the true coefficients we computed Q for the Lyapunov-Parks (Schur-Cohn) method and used SAMPL in the program shown in Fig. 4 to compute P for $Q = aa'$, where

$$a' = (.99999386, -3.3791469, 5.0853854, -4.5132249, 2.6029441, -.99640206, .24061150, -.029163730).$$

This is also the first row of P and may be compared with the results printed in Fig. 5. This matrix has a rank of only six,

rank	ρ
4	$.3810^{-3}$
5	$.75.10^{-4}$
6	$.54.10^{-4}$
7	$.23.10^{-3}$

which is disappointing, since six-decimal accuracy was achieved in the first row of P. This is not, however, as important as it appears. We shall illustrate our reasoning in Section 7 of this chapter. When using the differential equation (4.3a) to obtain the P matrix as an equilibrium point, the question is not so much whether the equilibrium is indefinite but whether it exists. There exist unique solutions to (4.1) and (4.2) but they are (for positive definite Q) positive definite only if F and Φ are asymptotically stable. On the other hand the solutions to our differential equations are always nonnegative definite and if (F, Q) (or (Φ, Q)) is completely controllable they are positive definite (not necessarily computationally). This allows us to use the nonnegative definite option in PSEUO to compute rank, but it prevents us from ever reaching an equilibrium if the system is not asymptotically stable.

To illustrate the general use of ASP in creating Lyapunov functions, we take F in a "scrambled" form (Fig. 6) obtained by a similarity transformation and using Q = I run the program shown in Fig. 7. This eventually converged, to the P in Fig. 8, indicating stability, but had a rank of 7:

rank	ρ
5	$.20.10^{-3}$
6	$.90.10^{-4}$
7	$.80.10^{-4}$
8	$.10.10^{-3}$

Again using a sampling interval of .25, we obtained Φ for the scrambled system and obtained the sampled Lyapunov function generated by (4.3). Instead of using Q = 1, we used Q = .25·I, since this should be approximately the same as Fig. 8. The resulting P appears in Fig. 9, notice that it is very similar to Fig. 8, the differences being introduced by sampling.

Now looking at the matrix P in Fig. 8 we can say that transients following initial conditions in x_1 will involve much smaller excursions than those in (for instance) x_3.

The following table of $\|x\|$ following initial conditions in x_1 and x_3 illustrates this very well.

t	x_1	x_3
0	1.0	1.0
.25	2.1	21.0
.5	1.9	20.0
.75	1.3	15.0
1.0	.66	7.8
1.25	.25	2.7
1.5	.05	1.9

6. Numerical Considerations: The system provides an excellent illustration of how important diagonal and scalar transformations are in ASP. Despite the fact that the eigenvalues of F are close together, operating in a companion basis gives very large ratios of numbers in the F matrix. As we see in (5.1) the largest element in F is 36669. and the smallest is 1. However the 1 is not ignorable since it provides a vital link in the topology of the system. Adding to the numerical problem is the fact that the P and Q matrices produced by the Lyapunov-Parks method have extremely large elements and are very poorly conditioned.

For these reasons we find certain very simple transformations necessary. Looking at the input in Fig. 2 we see that the system has been transformed so that the largest element/smallest element ratio is now 36.669 instead of 366669., and the same ratio in Q is $\frac{1188}{318}$ instead of $\frac{1188180900}{576}$. In addition to this transformation we equalize $\|Q\|$ and $\|F\|$ before computing the solution to the riccati equation.

To illustrate the effect of the two operations, we made three runs over 1000 computing intervals.

1) With no transformation, no equalization.
2) With equalization, no transformation.
3) With transformation, no equalization.

In 1) the most significant fact is that the size of $\|Q\|$ makes the computing interval so small that convergence is not approached. This gives a double argument for these operations -- they improve accuracy and save machine time.

In 2), slightly better results were obtained, the diagonal elements (compare with Fig. 1) are

$$(9\cdot2\cdot10^{6},\ 5\cdot4\cdot10^{7},\ 6\cdot7\cdot10^{7},\ 2\cdot0\cdot10^{7},\ 2\cdot4\cdot10^{6},\ 1\cdot3\cdot10^{5},$$
$$1\cdot9\cdot10^{3},\ 2\cdot0\cdot10).$$

In 3) results were even better

$$(1.3.10^{8},\ 5.1.10^{8},\ 3.1.10^{8},\ 6.8.10^{7},\ 6.3.10^{6},\ 2.5.10^{5},$$
$$4.1.10^{3},\ 2.3.10).$$

Notice that better is a relative term here, since none of these answers is acceptable.

It is not impossible that 3) could ultimately have converged to the correct answer, but we have illustrated our main point, that on any basis of accuracy or cost, simple transformations are essential.

That they are not always vital is illustrated by our experience in computing the characteristic equation of $e^{\cdot25F}$. It is quite evident here that a better job of inversion could be done if the columns of

$$[\Gamma, \Phi\Gamma, \ldots, \Phi^7\Gamma]$$

are normalized.

However in the present case no benefit was gained by so doing. Apparently the spread of eigenvalues ($e^{-.25}$ to $e^{-1.25}$) was not large enough to demand such sophistication.

Another artifice which can be used to save time and improve accuracy is the use of a changing time interval. In these runs we have used the largest computing interval possible, as indicated by the coding in Fig. 2 and Fig. 7. We then tried taking the output of this program and using it as input to another in which the computing interval had been decreased by a factor of ten. Unfortunately for illustrative purposes, this gave no improvement in accuracy.

Another observation about the mode of convergence. Using the maximum computing interval, we got better accuracy with an error criterion of 10^{-6} than with 10^{-8}. This was because the error never went under 10^{-8}, hence we were given the 1000^{th} iterant. By this time the error was no longer decreasing, it was just randomly distributed in the 10^{-5} - 10^{-7} range. An error criterion of 10^{-6} therefore selected a slightly better iterant.

One option in RICAT which has helped to improve accuracy is to use the transpose instead of the inverse when the riccati equation is linear (see Chapter IV). This saves time, and according to past experience, it improves accuracy. Here it did not. The matrices quoted above were all computed using the transpose, in Fig. 10 the inverse was used (compare with Fig. 1 and Fig. 3) and the accuracy was only slightly improved.

7. Remarks: To illustrate how convergence implies stability, regardless of the rank of P, we took

$$F = \text{diag}\,(10^{-2},\ -2,\ -3 \pm 2i,\ -3 \pm 4i,\ -4,\ -5),$$

used the same "scrambling" transformation to produce the F^* in Fig. 8, and tried to obtain a Lyapunov function.

The diagonal terms of the matrix obtained after 1000 iterations are all larger than those obtained previously. All but the third and fifth are quite close to the previous ones. Furthermore the percentage difference of successive iterants is decreasing. The following table shows, however, that the matrix is in no sense converging and displays an unbounded growth indicating instability.

$a_{33}(k)$	$a_{55}(k)$	k	$\rho(k)$
$1.2.10^5$	$.26.10^5$	100	$.2.10^{-2}$
$1.5.10^5$	$.39.10^5$	200	$.1.10^{-2}$
$1.8.10^5$	$.52.10^5$	300	$.1.10^{-2}$
$2.1.10^5$	$.66.10^5$	400	$.1.10^{-2}$
$2.5.10^5$	.80	500	$.1.10^{-2}$
$2.8.10^5$	$.95.10^5$	600	$.1.10^{-2}$
$3.1.10^5$	$1.1.10^5$	700	$.9.10^{-3}$
$3.5.10^5$	$1.2.10^5$	800	$.9.10^{-3}$
$3.8.10^5$	$1.4.10^5$	900	$.8.10^{-3}$
$4.2.10^5$	$1.6.10^5$	1000	$.8.10^{-3}$

We append a few Parks-Lyapunov and Schur-Cohn functions. We will give F, P, and Q where $V = x'(t)Px(t)$ is a possible Lyapunov function for the system $\dot{x} = Fx (x_{k+1} = Fx_k)$ and $PF + F'P + Q = 0$ $(P = F'PF + Q)$. That is, if F is stable,

$$x'(0)px(0) = \int_0^\infty x'(t)Qx(t)dt$$

$$(x_o'Px_o = \sum_{i=1}^{\infty} x_i'Qx_i).$$

$$F = [-a_1] \qquad P = [a_1] \qquad Q = 2a_1^2$$

- - - - - - - - - - - - -

$$F = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix} \qquad P = \begin{bmatrix} a_1a_2 & 0 \\ 0 & a_1 \end{bmatrix} \qquad Q = 2\begin{bmatrix} 0 & 0 \\ 0 & a_1^2 \end{bmatrix}$$

- - - - - - - - - - - - -

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_4 & -a_3 & -a_2 & -a_1 \end{bmatrix} \qquad P = \begin{bmatrix} a_3a_4 & 0 & a_1a_4 & 0 \\ 0 & a_2a_3 - a_1a_4 & 0 & a_3 \\ a_1a_4 & 0 & a_1a_2 - a_3 & 0 \\ 0 & a_3 & 0 & a_1 \end{bmatrix}$$

$$Q = 2\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & a_3^2 & 0 & a_1a_3 \\ 0 & 0 & 0 & 0 \\ 0 & a_1a_3 & 0 & a_1^2 \end{bmatrix}$$

- - - - - - - - - - - - -

$$
F = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & -a_8 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & -a_7 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & -a_6 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & -a_5 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & -a_4 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & -a_3 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & -a_2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -a_1
\end{bmatrix}
$$

$$
Q = 2 \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_7^2 & a_7 a_5 & a_7 a_3 & a_7 a_1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_7 a_5 & a_5^2 & a_5 a_3 & a_5 a_1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_7 a_3 & a_5 a_3 & a_3^2 & a_3 a_1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_7 a_1 & a_5 a_1 & a_3 a_1 & a_1^2 & 0 & 0 & 0
\end{bmatrix}
$$

$$
P = \begin{bmatrix}
a_7a_8 & 0 & a_5a_8 & 0 & a_3a_8 & 0 & a_1a_8 & 0 \\
 & a_6a_7 - a_5a_8 & 0 & a_4a_7 - a_3a_8 & 0 & a_2a_7 - a_1a_8 & 0 & a_7 \\
 & & a_3a_8 - a_4a_7 + a_5a_6 & 0 & a_1a_8 - a_2a_7 + a_3a_6 & 0 & a_1a_6 - a_7 & 0 \\
 & & & a_4a_5 - a_3a_6 + a_2a_7 - a_1a_8 & 0 & a_2a_5 - a_1a_6 + a_7 & 0 & a_5 \\
 & & & & a_3a_4 - a_2a_5 + a_1a_6 - a_7 & 0 & a_1a_4 - a_5 & 0 \\
 & \text{Symmetric} & & & & a_5 - a_1a_4 + a_2a_3 & 0 & a_3 \\
 & & & & & & a_1a_2 - a_3 & 0 \\
 & & & & & & & a_1
\end{bmatrix}
$$

$$F = [-a_1] \qquad P = (1 - a_1^2) \qquad Q = (1 - a_1^2)^2$$

- - - - - - - - - - -

$$F = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix} \qquad P = \begin{bmatrix} 1 - a_2^2 & a_1(1 - a_2) \\ a_1(1 - a_2) & 1 - a_2^2 \end{bmatrix}$$

$$Q = \begin{bmatrix} (1 - a_2^2)^2 & a_1(1 - a_2^2)(1 - a_2) \\ a_1(1 - a_2^2)(1 - a_2) & a_1^2(1 - a_2)^2 \end{bmatrix}$$

- - - - - - - - -

$$F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix}$$

$$Q = \begin{bmatrix} (1 - a_3^2)^2 & (1 - a_3^2)(a_1 - a_2a_3) & (1 - a_3^2)(a_2 - a_1a_3) \\ & (a_1 - a_2a_3)^2 & (a_1 - a_2a_3)(a_2 - a_1a_3) \\ \text{symmetric} & & (a_2 - a_1a_3)^2 \end{bmatrix}$$

$$P = \begin{bmatrix} (1 - a_3^2) & a_1 - a_2a_3 & a_2 - a_1a_3 \\ & 1 + a_1^2 - a_2^2 - a_3^2 & a_1 - a_2a_3 \\ \text{symmetric} & & 1 - a_3^2 \end{bmatrix}$$

- - - - - - - - - - - - - - - - - -

F = companion $(a_8, a_7, a_6, a_5, a_4, a_3, a_2, a_1)$

$Q = aa'$

$a = [(1 - a_8^2), (a_1 - a_7a_8), (a_2 - a_6a_8), (a_3 - a_5a_8), (a_4 - a_4a_8),$

$(a_5 - a_3a_8), (a_6 - a_8), (a_7 - a_1a_8)]$.

- - - - - - - - - - - - - - - - - -

Some sidelights were discovered during the development of this chapter. Despite the fact that they are not new and not difficult to prove, they seem to be sufficiently interesting in themselves to warrant inclusion.

1. Let F be a matrix in companion form, and λ an eigenvalue of F. Then $(1, \lambda, \ldots, \lambda^{n-1})$ is an associated eigenvector.

2. Let $\Lambda = \text{diag}(\lambda_i)$ $\lambda_i \neq \lambda_j$ if $i \neq j$. Then

$$\begin{bmatrix} 1 & 1 & - & 1 \\ \lambda_1 & \lambda_2 & - - - & \lambda_n \\ - & - & - & - \\ \lambda_1^{n-1} & \lambda_2^{n-1} & - - - & \lambda_n^{n-1} \end{bmatrix}$$

is nonsingular and

$$V\Lambda V^{-1} = C$$

where C is in companion form.

Proof: It is well known that the Vandermonde determinant of distinct numbers is nonzero, hence V^{-1} exists. The first $n-1$ rows of $V\Lambda$ are the same as the last $n-1$ rows of V. Hence the first $n-1$ rows of $V\Lambda V^{-1}$ are in companion form. But Λ is similar to $V\Lambda V^{-1}$ so $V\Lambda V^{-1} = C$, the companion form of Λ.

Alternately from 1. above, V is the matrix of eigenvectors, so if C is the companion form of Λ, $V^{-1}CV = \Lambda$.

3. If V is an invertible matrix such that the first $n-1$ rows of VF are the last $n-1$ rows of V, then VFV^{-1} is the companion form of F.

4. The minimal polynomial of a companion matrix F is its characteristic polynomial.

Proof: Case 1. The last row of F is zero. Then F is a maximal Jordan block and its minimal polynomial is $S^n = 0$.

Case 2. If some coefficient of the characteristic polynomial is nonzero, so is some coefficient b_p of the minimal polynomial $\sum_{\ell=0}^{m} b_\ell F^\ell = [a_{ij}]$. Because F satisfies the minimal polynomial, $a_{1,p+1} = 0$. Consider the terms in $a_{1,p+1}$. They consist of $b_p \neq 0$ occurring in $b_p F^p$, and no other term unless $m = n$.

5. Let J be a single Jordan block with eigenvalue λ. Then

$$V = \begin{bmatrix} 1 & 1 & 1 & - & 1 \\ \lambda & \lambda + 1 & \lambda + 1 & - & \lambda + 1 \\ \lambda^2 & \lambda^2 + 2\lambda & (\lambda + 1)^2 & - & (\lambda + 1)^2 \\ \lambda^3 & \lambda^3 + 3\lambda^2 & \lambda^3 + 3\lambda^2 + 3\lambda & - & (\lambda + 1)^3 \\ \lambda^4 & \lambda^4 + 4\lambda^3 & \lambda^4 + 4\lambda^3 + 6\lambda^2 & - & (\lambda + 1)^4 \\ - & - & - & - & - \\ \lambda^{n-1} & \lambda^{n-1} + (n-1)\lambda^{n-2} & \lambda^{n-1} + (n-1)\lambda^{n-2} + \frac{(n-1)(n-2)}{2}\lambda^{n-3} & - & (\lambda + 1)^n \end{bmatrix}$$

is nonsingular and VJV^{-1} is the companion form for J.

Proof: Subtract the j^{th} column of V from the $(j+1)^{st}$, $j = n-1, n-2, \ldots, 1$. This leaves V in lower triangular form with 1 on the diagonal, hence V^{-1} exists.

The first $n-1$ rows of VJ are the last $n-1$ rows of V, the conclusion follows from 3.

6. Let J be an $n \times n$ matrix in Jordan form. It follows from 4. and the fact that the minimal polynomial is an invariant, that J has a companion form only if distinct Jordan blocks have distinct eigenvalues. We show now that J has a companion form if distinct Jordan blocks have distinct eigenvalues.

Proof: We will define a transforming matrix T satisfying 3.

Consider a $k \times k$ Jordan block M in J. From 5 we have a V which would put this block in companion form. Extend V to an $n \times k$ matrix in the obvious way; alternatively, take the first k columns of the V which puts a single $n \times n$ Jordan block in companion form.

Let these k columns be the k columns of T which are multiplied with M.

Do this for all Jordan blocks. This completely defines T. Now if T is nonsingular it satisfies 3. and we are through.

By the method used in 5. we can reduce this question to that of the singularity of column juxtaposes of matrices

$$\begin{bmatrix} 1 & 0 & 0 & 0 & - \\ \lambda & 1 & 0 & 0 & - \\ \lambda^2 & 2\lambda & 1 & 0 & - \\ \lambda^3 & 3\lambda^2 & 3\lambda & 1 & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ \lambda^n & (n-1)\lambda^{n-1} & \binom{n}{2}\lambda^{n-2} & \binom{n}{3}\lambda^{n-3} & - \end{bmatrix}$$

with distinct eigenvalues λ.

Now prove the assertion by induction.

In 5. we showed it true for one eigenvalue.

If one of the eigenvalues is zero it is easy to reduce T to a form which is singular only if the induction hypothesis is violated.

If none of the λ's are zero: Let λ be the eigenvalue with the largest Jordan block and μ some other eigenvalue. Then dividing the λ columns by μ and subtracting from the μ columns we can obtain an equivalent problem with a zero eigenvalue.

$$P = \begin{bmatrix} 448110000 & 0 & 27212000 & 0 & 23166000 & 0 & 312000 & 0 \\ & 986768430 & 0 & 24097610 & 0 & 8891490 & 0 & 34470 \\ & & 540952146 & 0 & 56452668 & 0 & 845586 & 0 \\ & & & 106953144 & 0 & 4847922 & 0 & 21324 \\ & & & & 8807544 & 0 & 162588 & 0 \\ & & & & & 313206 & 0 & 1782 \\ & \text{symmetric} & & & & & 4626 & 0 \\ & & & & & & & 24 \end{bmatrix}$$

Fig. 1

```
BEGIN
LOAD    F, Q,ZR,LT, C, D,PC,2.,PO,
MULT   2.,  Q,     P,
NORM    F,    NF,
NORM    P,    NP,
PSEUO NP,    NP,RK,
MULT   NF,NP,    LM,
MULT   LM,  P,     Q,
SUBT   ZR,  F,    MF,
TRANP   F,    FT,
JUXTC MF,ZR,    TP,
JUXTC   Q,FT,    BT,
JUXTR TP,BT,    TH,
NORM   TH,    NT
PSEUO NT,    NT,RK,
MULT   LT,NT,     T,
ETPHI  TH,  T,PH,    PRINT
MULT   LM,PO,    ZR,
PSEUO LM,    LM,RK,
RICAT  ZR,PH,  C,  D,PC,     P,  K,AL,
MULT   LM,  P,     R,
RINT    R,  P,
PSEUO   R,+ IPI,RK,PRINT
PUNCH   R,  Q
END
```

8	8			
0	1.	0	0	0
0	0	0	0	0
1.	0	0	0	0
0	0	0	0	1.
0	0	0	0	0
0	0	0	1.	0
0	0	0	0	0
0	0	10.	0	0
0	0	0	0	0
0	10.	0	0	0
0	0	0	0	0
10.	-13.	-34.47	-36.669	-21.324
-7.663	-17.82	-26.7	24.	
8	8			
0	0	0	0	0
0	0	0	0	1188.1809
0	735.03828	0	614.2554	0
827.28	0	0	0	0
0	0	0	0	0
735.03828	0	454.71296	0	379.99368
0	511.776	0	0	0
0	0	0	0	0
0	614.2554	0	379.99368	0
317.5524	0	427.68	0	0
0	0	0	0	0
0	0	827.28	0	511.776
0	427.68	0	576.	

Fig. 2

```
  8      8
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0
  1      1
10.001
  1      8
   0              0              0              0              0
   0              0              0
  1      3
 1.E-9          1.          1000.
  1      4
   0           100.           1.              0
  1      1
  2.
  8      8
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0              0
   0              0              0              0
```

Fig. 2

MATRIX P

NUMBER OF ROWS 8 NUMBER OF COLUMNS 8

0.44810752E 03	-0.16849236E-02	0.27721081E 03	-0.32718499E-03	0.23165917E 02	-0.11238925E-03
0.31199894E 02	-0.40461378E-04				
-0.16849236E-02	0.98676452E 03	-0.16424969E-02	0.24097686E 03	-0.13950908E-03	0.88914660E 02
-0.18950608E-03	0.34469916E 02				
0.27721081E 03	-0.16424969E-02	0.54095074E 03	-0.38067595E-03	0.56452557E 02	-0.15107245E-03
0.84558443E 02	-0.59397428E-04				
-0.32718499E-03	0.24097686E 03	-0.38067595E-03	0.10695294E 03	-0.33858197E-04	0.48479147E 02
-0.60728463E-04	0.21323974E 02				
0.23165917E 02	-0.13950908E-03	0.56452557E 02	-0.33858197E-04	0.88075341E 01	-0.12395261E-04
0.16258784E 02	-0.55841069E-05				
-0.11238925E-03	0.88914660E 02	-0.15107245E-03	0.48479147E 02	-0.12395261E-04	0.31320577E 02
-0.17469832E-04	0.17819990E 02				
0.31199894E 02	-0.18950608E-03	0.84558443E 02	-0.60728463E-04	0.16258784E 02	-0.17469832E-04
0.46259972E 02	-0.89220926E-05				
-0.40461378E-04	0.34469916E 02	-0.59397428E-04	0.21323974E 02	-0.55841069E-05	0.17819990E 02
-0.89220926E-05	0.23999992E 02				

FIG. 3

```
BEGIN
LOAD   F, Q, G, R, D,PC,
TRANP  F,   FT,
SAMPL FT, Q, G, R, Q, D,PC,     P, K,AL,
RINT   P, P
PSEUO  P,+ 1PI,RK,PRINT
END
      8      8
         0              1.              0              0              0
         0              0               0              0              0
        1.              0               0              0              0
         0              0               0              0             1.
         0              0               0              0              0
         0              0               0             1.              0
         0              0               0              0              0
         0              0              1.              0              0
         0              0               0              0              0
         0              1.              0              0              0
         0              0               0              0              0
        1.    -.002478734     .037539967    -.2532183768    1.007595337
-2.6094121455    4.515722478    -5.08601306        3.37924
      8      8
  .999987712   -3.379126186     .08535415   -4.513197187   2.6029281135
 -.99639594     .2406100256   -.029163549818  -3.379126186    11.4186341
-17.1842646     15.2508502    -8.79573063    3.366988986    -.8130616294
 .09854852922    5.08535415   -17.1842646    25.86114466    -22.9514881
 13.23697395   -5.067088497    1.223602229   -.1483088067   -4.513197187
 15.2508502    -22.9514881     20.36919915   -11.74767219    4.496986614
-1.085933835    .1316224729    2.6029281135   -8.79573063    13.23697395
-11.74767219    6.775318018   -2.593578874     .6262982962   -.07591155911
 -.99639594     3.366988986   -5.067088497    4.496986614    -2.593578874
 .9928170691   -.2397457987    .02905880071    .2406100256     .8130616294
 1.223602229   -1.085933835    .626298296     -.2397457987    .0578938958
-.007017128937  -.029163549818  .09854852922   -.1483088067    .1316224729
-.07591155911   .02905880071  -.007017128937   .0008505231475
      1      8
         0              0               0              0              0
         0              0               0
      1      1
        0
      1      2
     1.E-7           1.
      1      4
        0            100.             1.           1000.
```

Fig. 4

MATRIX P

NUMBER OF ROWS 8 NUMBER OF COLUMNS 8

0.99999384E 00	-0.33791468E 01	0.50853848E 01	-0.45132231E 01	0.26029401E 01	-0.99639608E 00
0.24060622E-00	-0.29161556E-01				
-0.33791468E 01	0.12417846E 02	-0.20556492E 02	0.20307245E 02	-0.13233040E 02	0.58382457E 01
-0.16610840E 01	0.24058066E-00				
0.50853848E 01	-0.20556492E 02	0.38221213E 02	-0.43268226E 02	0.32917666E 02	-0.17213748E 02
0.58378630E 01	-0.99620940E 00				
-0.45132231E 01	0.20307245E 02	-0.43268226E 02	0.57597196E 02	-0.52421236E 02	0.32916105E 02
-0.13231401E 02	0.26022442E 01				
0.26029401E 01	-0.13233040E 02	0.32917666E 02	-0.52421236E 02	0.57594770E 02	-0.43263954E 02
0.20303351E 02	-0.45116063E 01				
-0.99639608E 00	0.58382457E 01	-0.17213748E 02	0.32916105E 02	-0.43263954E 02	0.38214687E 02
-0.20550682E 02	0.50829767E 01				
0.24060622E-00	-0.16610840E 01	0.58378630E 01	-0.13231401E 02	0.20303351E 02	-0.20550682E 02
0.12412692E 02	-0.33770088E 01				
-0.29161556E-01	0.24058066E-00	-0.99620940E 00	0.26022442E 01	-0.45116063E 01	0.50829767E 01
-0.33770088E 01	0.99910301E 00				

FIG. 5

$$\begin{bmatrix} -7 & 2 & -35 & 18 & -21 & 6 & 2 & -8 \\ -6 & 5 & -54 & 44 & -18 & 18 & 24 & -12 \\ 8 & 0 & 41 & -16 & 30 & 0 & 4 & 16 \\ 0 & -16 & 48 & -84 & 0 & -42 & -48 & 8 \\ -12 & 0 & -63 & 24 & -46 & 0 & -6 & -24 \\ 0 & 24 & -72 & 123 & 0 & 61 & 72 & -12 \\ 10 & 0 & 54 & -20 & 36 & 0 & 1 & 20 \\ 0 & -20 & 60 & -101 & 0 & -54 & -60 & 5 \end{bmatrix}$$

Fig. 6

```
BEGIN
LOAD    F, Q,ZR,LT, C, D,PC,2.,PO
MULT  2.,  Q,     P,
NORM    F,    NF,
NORM    P,    NP,
PSEUO NP,    NP,RK,
MULT  NF,NP,    LM,
MULT  LM,  P,     Q,
SUBT  ZR,  F,    MF,
TRANP   F,    FT,
JUXTC MF,ZR,    TP,
JUXTC   Q,FT,    BT,
JUXTR TP,BT,    TH,
NORM  TH,    NT,
PSEUO NT,    NT,RK,
MULT  LT,NT,     T,
ETPHI TH, T,PH,    PRINT
MULT  LM,PO,    ZR,
PSEUO LM,    LM,RK,
RICAT ZR,PH, C, D,PC,     P, K,AL,
MULT  LM,  P,     R,
RINT    R,  P
PUNCH   R,  Q,
PSEUO   R,+ 1PI,RK,PRINT
END
```

8 8

-7.	2.	-35.	18.	-21.
6.	2.	-8.	-6.	5.
-54.	44.	-18.	18.	24.
-12.	8.	0	41.	-16.
30.	0	4.	16.	0
-16.	48.	-84.	0	-42.
-48.	8.	-12.	0	-63.
24.	-46.	0	-6.	-24.
0	24.	-72.	123.	0
61.	72.	-12.	10.	0
54.	-20.	36.	0	1.
20.	0	-20.	60.	-101.
0	-54.	-60.	5.	

8 8

1.	0	0	0	0
0	0	0	0	1.
0	0	0	0	0
0	0	0	1.	0
0	0	0	0	0
0	0	1.	0	0
0	0	0	0	0
0	1.	0	0	0
0	0	0	0	0
1.	0	0	0	0
0	0	0	0	1.
0	0	0	0	0
0	0	0	1.	

Fig. 7

```
    8      8
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0
    1      1
 10.001
    1      8
     0             0             0             0             0
     0             0             0
    1      3
  1.E-9           1.          1000.
    1      4
     0          100.            1.           0
    1      1
     1.
    8      8
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0             0
     0             0             0             0
```

Fig. 7

MATRIX P

NUMBER OF ROWS 8 NUMBER OF COLUMNS 8

0.22948560E 01	-0.21754838E 01	0.20610330E 02	-0.12414787E 02	0.85826727E 01	-0.57872545E 01
-0.59379476E 01	0.26902515E 01				
-0.21754838E 01	0.69380915E 01	-0.34442308E 02	0.31603233E 02	-0.88120139E 01	0.18066667E 02
0.19654478E 02	-0.20606137E 01				
0.20610330E 02	-0.34442308E 02	0.25098671E 03	-0.17699056E 03	0.91184579E 02	-0.93104426E 02
-0.97907014E 02	0.23235960E 02				
-0.12414787E 02	0.31603233E 02	-0.17699056E 03	0.15783449E 03	-0.51084081E 02	0.86971477E 02
0.90577329E 02	-0.15829172E 02				
0.85826727E 01	-0.88120139E 01	0.91184579E 02	-0.51084081E 02	0.39892167E 02	-0.24218623E 02
-0.24427154E 02	0.10482375E 02				
-0.57872545E 01	0.18066667E 02	-0.93104426E 02	0.86971477E 02	-0.24218623E 02	0.49614302E 02
0.52436596E 02	-0.62102454E 01				
-0.59379476E 01	0.19654478E 02	-0.97907014E 02	0.90577329E 02	-0.24427154E 02	0.52436596E 02
0.57129399E 02	-0.49980175E 01				
0.26902515E 01	-0.20606137E 01	0.23235960E 02	-0.15829172E 02	0.10482375E 02	-0.62102454E 01
-0.49980175E 01	0.56334405E 01				

FIG. 8

MATRIX P

NUMBER OF ROWS 8 NUMBER OF COLUMNS 8

0.24274220E 01	-0.21680675E 01	0.20459716E 02	-0.12433867E 02	0.85219829E 01	-0.58551928E 01
-0.60634254E 01	0.25981777E 01				
-0.21680675E 01	0.67887457E 01	-0.33562320E 02	0.30290446E 02	-0.87729258E 01	0.17268206E 02
0.18862665E 02	-0.17716052E 01				
0.20459716E 02	-0.33562320E 02	0.24739847E 03	-0.17325947E 03	0.90344101E 02	-0.91178476E 02
-0.96518813E 02	0.21776534E 02				
-0.12433867E 02	0.30290446E 02	-0.17325947E 03	0.15239606E 03	-0.50982086E 02	0.83510271E 02
0.87463649E 02	-0.14419687E 02				
0.85219829E 01	-0.87729258E 01	0.90344101E 02	-0.50982086E 02	0.39665682E 02	-0.24371780E 02
-0.24834466E 02	0.10131323E 02				
-0.58551928E 01	0.17268206E 02	-0.91178476E 02	0.83510271E 02	-0.24371780E 02	0.47593814E 02
0.50324453E 02	-0.55469705E 01				
-0.60634254E 01	0.18862665E 02	-0.96518813E 02	0.87463649E 02	-0.24834466E 02	0.50324453E 02
0.55227789E 02	-0.44559700E 01				
0.25981777E 01	-0.17716052E 01	0.21776534E 02	-0.14419687E 02	0.10131323E 02	-0.55469705E 01
-0.44559700E 01	0.53921065E 01				

FIG. 9

- 297 -

MATRIX P

NUMBER OF ROWS 8 NUMBER OF COLUMNS 8

0.44810817E 03	-0.12370304E-02	0.27721113E 03	-0.23160005E-03	0.23165947E 02	-0.82607347E-04
0.31199912E 02	-0.36114717E-04				
-0.12370304E-02	0.98676633E 03	-0.90876401E-03	0.24097717E 03	-0.84354330E-04	0.88914763E 02
-0.12361985E-03	0.34469945E 02				
0.27721113E 03	-0.90876401E-03	0.54095149E 03	-0.15373451E-03	0.56452638E 02	-0.54905185E-04
0.84558530E 02	-0.31612076E-04				
-0.23160005E-03	0.24097717E 03	-0.15373451E-03	0.10695303E 03	-0.89844849E-05	0.48479191E 02
-0.23625867E-04	0.21323988E 02				
0.23165947E 02	-0.84354330E-04	0.56452638E 02	-0.89844849E-05	0.88075444E 01	-0.12478451E-06
0.16258796E 02	-0.22669186E-05				
-0.82607347E-04	0.88914763E 02	-0.54905185E-04	0.48479191E 02	-0.12478451E-06	0.31320602E 02
0.29948283E-05	0.17820000E 02				
0.31199912E 02	-0.12361985E-03	0.84558530E 02	-0.23625867E-04	0.16258796E 02	0.29948283E-05
0.46259999E 02	0.22877160E-06				
-0.36114717E-04	0.34469945E 02	-0.31612076E-04	0.21323988E 02	-0.22669186E-05	0.17820000E 02
0.22877160E-06	0.23999999E 02				

FIG. 10

CHAPTER XI

ASYMPTOTIC BEHAVIOR OF ROOTS OF THE CLOSED-LOOP SYSTEM

1. Description of the Problem. We shall examine the behavior of the closed loop poles of optimal systems as the weighting factor for the control energy tends to zero.

$$\frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_o}{s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_o}, \tag{1.1}$$

with an associated system in control canonical form,

$$\begin{aligned} \dot{x} &= Fx + Gu \\ y &= Hx \end{aligned} \tag{1.2}$$

where $F = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & & 1 \\ -a_o & \cdots & & -a_{n-1} \end{pmatrix}$

$G' = (0, 0, \ldots, 1)$

$H = (b_o, b_1, \ldots, b_m, 0, 0, \ldots, 0)$.

We wish to optimize the functional

$$\int_0^\infty (\|y\|^2 + r\|u\|^2)dt \quad (r = \text{scalar}), \tag{1.3}$$

and then examine the eigenvalues of the optimal system matrix $F - GK$ as r tends to 0.

2. Theory and References: For the procedures involved in optimization see Chapter VII.

The asymptotic behavior of the closed loop poles is described in

[1] R. E. Kalman, "When is a Linear Control System Optimal?", Journal of Basic Engineering, March, 1964.

The relevant result is this: as $r \to 0$ in (1.3) m of the closed loop poles will go to the m zeroes of (1) in the left-half plane (or their reflections about the imaginary axis, if some of the zeros are in the right-half plane) and the other $n-m$ poles will converge to a butterworth pattern with radius

$$\left(\frac{a_o^2 b_m^2}{b_o^2} + \frac{b_o^2}{r} \right)^{\frac{1}{2(n-m)}}$$

The following proposition supplements the results of the reference, providing a simpler derivation of the product of the roots, though not specifying where the individual roots are located.

PROPOSITION. Given the transfer function

$$\frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_o}{x^n + a_{n+1} s^{n-1} + \ldots + a_1 s + a_o}.$$

If we optimize

$$\int_0^\infty (y'y + ru^2)dt$$

the product of the roots of the optimal closed loop system matrix will be

$$\sqrt{a_o^2 + \frac{b_o^2}{r}}. \tag{2.1}$$

Proof: The product of the closed-loop poles is invariant under a change of basis. Therefore we may choose the control-canonical representation with F in companion form,

$$G' = [0,\ 0,\ \ldots,\ 0,\ 1]$$

and

$$H = [b_o,\ b_1,\ \ldots,\ b_m 0,\ 0,\ \ldots,\ 0].$$

Then the closed loop matrix $F - r^{-1}GG'P$ is also in companion form and the product $(-1)^n \pi_i \lambda_i$ of its roots is $(a_o + rp_{1n})$, where P satisfies the relation

$$\dot{P} = PF + F'P + H'H - r^{-1}PGG'P = 0.$$

To determine p_{1n}, we note that

$$\dot{p}_{1n} = 0 = -2p_{1n}a_o + b_o^2 - 1p_{1n}^2,$$

so that

$$p_{1n} = r[-a_o \pm \sqrt{a_o^2 + r^{-1}b_o^2}\ ]$$

and $(-1)^n \pi_i \lambda_i = \sqrt{a_o^2 + r^{-1}b_o^2}$. Since we know that the constant term of the characteristic polynomial of $r^{-1}GG'P$ must be positive, it follows that

$$p_{1n} = r[-a_o + \sqrt{a_o^2 + r^{-1}b_o^2}].$$

and (2.1) is proved.

If $m = 0$, then as $r \to 0$ the closed-loop poles approach a butterworth pattern with radius

$$\left(a_o^2 + r^{-1}b_0^2\right)^{\frac{1}{2n}}.$$

If the open-loop transfer function has m zeroes, then the other $n-m$ poles will approach a butterworth pattern with radius

$$\left(\frac{a_o^2 b_m^2}{b_o^2} + \frac{b_m^2}{r}\right)^{\frac{1}{2(n-m)}}.$$

3. The Specific Problem: We took the following four transfer functions

$$\frac{1}{s^6 + s^5 + .25s^4 - .5s^3 + 1.25s^2 - 5.s - 2.5} \qquad (A)$$

$$\frac{1 - s^2}{\sim} \qquad (B)$$

$$\frac{1 - s + .5s^2}{\sim} \qquad (C)$$

$$\frac{1 + s + .5s^2}{\sim} \qquad (D)$$

with poles at ± 1, $-1 \pm i$, $1/2 \pm i$, and zeroes at (B) ± 1, (C) $1 \pm i$, and (D) $-1 \pm i$.

We selected control-canonical coordinates with

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 2.5 & .5 & -1.25 & .5 & -.25 & .5 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

and

$$H_A = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$H_B = [1 \quad 0 \quad -1 \quad 0 \quad 0 \quad 0]$$

$$H_C = [1 \quad -1 \quad .5 \quad 0 \quad 0 \quad 0]$$

$$H_D = [1 \quad 1 \quad .5 \quad 0 \quad 0 \quad 0].$$

Then systems (B) and (D) are not completely observable, because of cancellation of a factor between the numerator and denominator of the transfer functions.

4. Results: Using the program appearing in Fig. 1, we obtained the closed loop system matrices $F - GK$ which minimize the performance index

4.1) $$\int_0^\infty [(Hx)^2 + ru^2]dt$$

'or

$$r^{-1} = .1,\ 1.,\ 10,\ 30,\ 100,\ 300,\ 1000,\ 3000.$$

Using a separate (non-ASP) program we then obtained the eigenvalues of these matrices. The results are graphed in Fig. 2, 3, 4 and may be summarized as follows:

1) As $r^{-1} \to 0$, the closed loop eigenvalues are "the same" as the open loop eigenvalues with negative real parts. In this case they are $-1,\ -1,\ -1 \pm i,\ -1/2 \pm i$.

2) The eigenvalues of (C) and (D) were the same, as the lemma in Section 6 of this chapter assures us. The performance indices were different of course.

3) Observability does not seem to affect the conclusions, in both (B) and (D) the expected behavior occurred.

In the graphs, the straight lines are the asymptotes of the roots. The circles have radius

$$\left(|b_m| \sqrt{a_o^2 + r^{-1}} \right)^{\frac{1}{6-m}} .$$

5. Numerical Problems: There were no numerical problems in ASP, however there are some points of interest. In Fig. 1 the code from

```
NORM   Q,    NQ,
MULT   SJ,   C,   C,
```

to

is an equalization of $\|H'H\|$ and norm$\|Gr^{-1}G'\|$ to cut down on computer time and help accuracy. This will be done routinely in all such optimization problems.

Fig. 5 is a table for system (A) showing the max value of the performance index and the times required for convergence using different values of r^{-1}. τ is the time step used in the riccati equation, N the number of steps required for convergence, T the total time required for convergence.

p_{ii}^{max} is the largest diagonal element of the performance index matrix in the computer weighting, p_{ii}^{max} normalized returns it to the weighting (4.1).

Roughly we may say that if the equalization had not been done then for $r^{-1} = 3000$ we would have had $\tau = \frac{9}{3000 + 6} \approx .003$ instead of $\tau = \frac{9}{\sqrt{3000} + 6} \approx .15$. Disregarding the accuracy problem, this would have caused N to be over 3000.

The other coding aspect is the method of introducing a sequence of r^{-1}, s. This is done very nicely by using a LOAD in the loop. However it is most convenient not to have to use a counter in the loop. This may done merely by terminating with an error return. Following the last RI, FORTRAN finds an input error and ejects, which is satisfactory if this is in the last ASP program. A less time consuming method (and the only method possible if another ASP program follows) is to follow the last RI with a dummy RI having the wrong dimension. This causes an ASP (not FORTRAN) error which goes on to the next BEGIN card.

6. Appendix: We shall give here a necessary and sufficient condition for the irrelevance of off diagonal terms in the error weighting of a cost functio

Let us consider a single input, stationary, real linear system

$$\dot{x} = Fx + gu$$

with performance index

$$\int_0^\infty [\|x\|_Q^2 + u^2]dt. \tag{6.1}$$

We know [1] that the optimal feedback gain k which specifies the control law: $u = -k'x$) will be the same for the different performance indices (1) implied by the error weighting matrices Q_1 and Q_2 if

$$\|\Phi(s)g\|_{Q_1}^2 = \|\Phi(s)g\|_{Q_2}^2 \quad \text{for all pure imaginary } s, \tag{6.2}$$

where $\Phi(s) = sI - F)^{-1}$.

Let $\gamma_i(s) = (\ (s)g)_i$ and $r_i\ (s) = \dfrac{\gamma_i(s)}{\gamma_j(s)}$, where $r_{ij}(s)$ is a rational function of s. We will show that $q_{k\ell} = q_{\ell k}$ has no effect upon (2) if and only if $r_{k\ell}(i\omega)$ is pure imaginary, or, more simply, if and only if $r_{k\ell}(s)$ is an odd function of the real variable s.

What is needed is a necessary and sufficient condition for the coefficient of $q_{k\ell}$ in (2) to be zero. But this coefficient is

$$[\bar{\gamma}_k(i\omega)\gamma_\ell(i\omega) + \bar{\gamma}_\ell(i\omega)\gamma_k(i\omega)] = 2\ \mathrm{Re}\ \bar{\gamma}_k(i\omega)\gamma_\ell(i\omega).$$

The necessary and sufficient condition that the real part of the product $\bar{w}z$ be zero is that $w = icz$, where c is a real constant. A necessary and sufficient condition that $\mathrm{Re}\ \bar{\gamma}_k(i\omega)\gamma_\ell(i\omega) = 0$ then is that $\gamma_k(i\omega) = if(\omega)\gamma_\ell(i\omega)$ where $f(\omega)$ is real. But this says precisely that $r_{k\ell}(i\omega)$ is imaginary. But if $r_{k\ell}(i\omega)$ is imaginary, then $r_{k\ell}(i\omega) + \overline{r_{k\ell}(i\omega)} = r_{k\ell}(i\omega) + r_{k\ell}(-i\omega) = 0$. But this says that $r_{k\ell}(s)$ is an odd function.

Clearly this condition cannot be determined from the open loop alone, but if $\dot{x}_k = cx_\ell$ then $s\gamma_k(s) = c\gamma_\ell(s)$ and the condition is satisfied. On the other hand if $\dot{x}_k = cx_\ell + u$, then the condition is satisfied if and only if $\gamma_\ell(s)$ is an even function of the real variable s.

As an example, consider a system in characteristic form

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 1 \\ -a_o & -a_1 & -a_g & -a_{n-1} \end{bmatrix} \quad G = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix}$$

then $\gamma_i(s) = s^{i-1}$ and if Q is replaced by

$$\hat{Q} = \begin{bmatrix} q_{11} & 0 & q_{13} & \cdots \\ 0 & q_{22} & 0 & q_{24} \\ q_{13} & 0 & q_{33} & \cdots \\ \cdot & q_{24} & \cdot & \\ \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \end{bmatrix}$$

i.e., q_{ij} is set equal to 0 when $i + j = \text{odd}$, the control law remains the same.

This observation saves a little computing time and explains a phenomenon which might appear odd if encountered for the first time.

The situation will occur in Chapter XVII and appears to be true also after the introduction of sampling.

```
        BEGIN
        LOAD   TT   D D1,PC   F   G   Q RI ZR XX
        TRANP  F,    FT,
        TRANP  G,    GT,
        SUBT   ZR, F,     1,
HEAD 1MULT   RI,GT,     C,
        MULT    G, C,     5,
        NORM    Q,    NQ,
        NORM    5     N5
        PSEUO NQ,    PQ,RK,
        MULT   PQ,N5     M1
        DECOM M1,     S,SJ,ER,PE, E,RK,
        MULT    S, Q,     6,
        MULT   SJ, 5,     5,
        MULT   SJ, C,     C,
        JUXTC   1, 5,     2,
        JUXTC   6,FT,     3,
        JUXTR   2, 3,    PH,
        NORM   PH,    NP
        PSEUO NP,    NP,RK,
        MULT   NP,TT,     T,
        RINT   PH,PH
        ETPHI PH, T,PH,
        MULT    T,D1,     4,
        ADD     4, D,    D2,
        RICAT  Q,PH, C,D2,PC,XX, P, K,AL,
        MULT    G, K,    GK,
        SUBT    F,GK,    CF,
        RINT    P,PER K, K CF,CF
        LOAD   RI,
        IF     TT,TT,HEAD 1
        END
TT          1      1
          9.
 D          3      1
          .00001              0              0
D1          3      1
            0           -1.          -1000.
PC          4      1
          100.          100.           1.           1.
 F          6      6
            0           1.             0            0            0
            0             0            0           1.            0
            0             0            0            0            0
           1.             0            0            0            0
            0             0           1.            0            0
            0             0            0           1.
          2.5            .5         -1.25           .5         -.25
       -1.
 G          6      1
            0             0            0            0            0
    1.
 Q          6      6
```

Fig. 1

```
1.          -1.          .5          0          0
0           -1.          1.          -.5        0
0           0            .5          -.5        .25
0           0            0           0          0
0           0            0           0          0
0           0            0           0          0
0           0            0           0          0
0
RI          1     1
30.
ZR          6     6
                  0           0           0          0          0
                  0           0           0          0          0
                  0           0           0          0          0
                  0           0           0          0          0
                  0           0           0          0          0
                  0           0           0          0          0
                  0           0           0          0          0
                  0
XX          6     6
                  1.          0           0          0          0
                  0           0           1.         0          0
                  0           0           0          0          1.
                  0           0           0          0          0
                  0           1.          0          0          0
                  0           0           0          1.         0
                  0           0           0          0          0
                  1.
RI          1     1
100.
RI          1     1
300.
RI          1     1
1000.
RI          1     1
3000.
RI          2     2
0                 0           0          0
```

Fig. 1

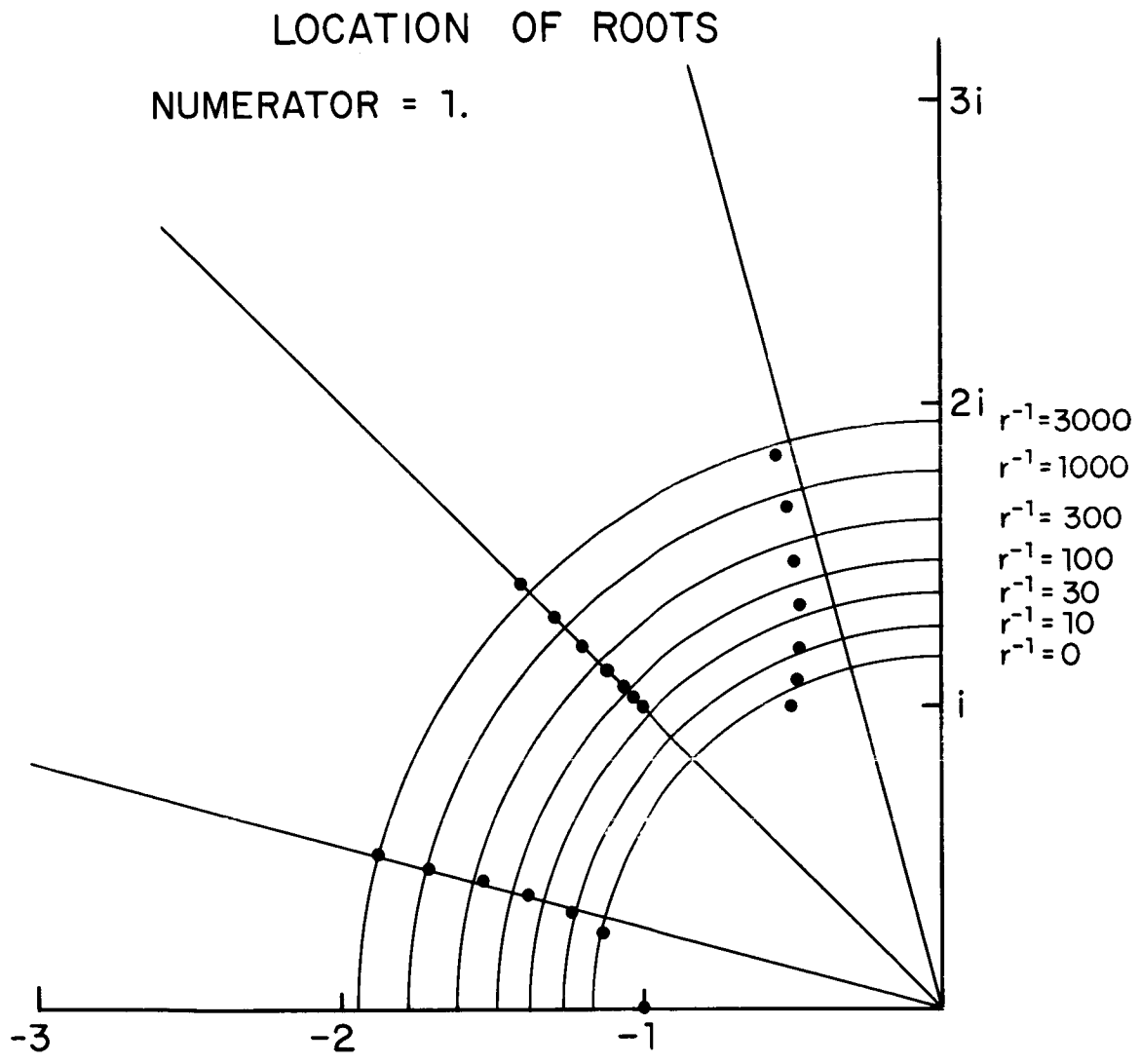
LOCATION OF ROOTS
NUMERATOR = 1.
3i
2i
i
-3
-2
-1
r⁻¹=3000
r⁻¹=1000
r⁻¹= 300
r⁻¹= 100
r⁻¹= 30
r⁻¹= 10
r⁻¹= 0

Fig. 2

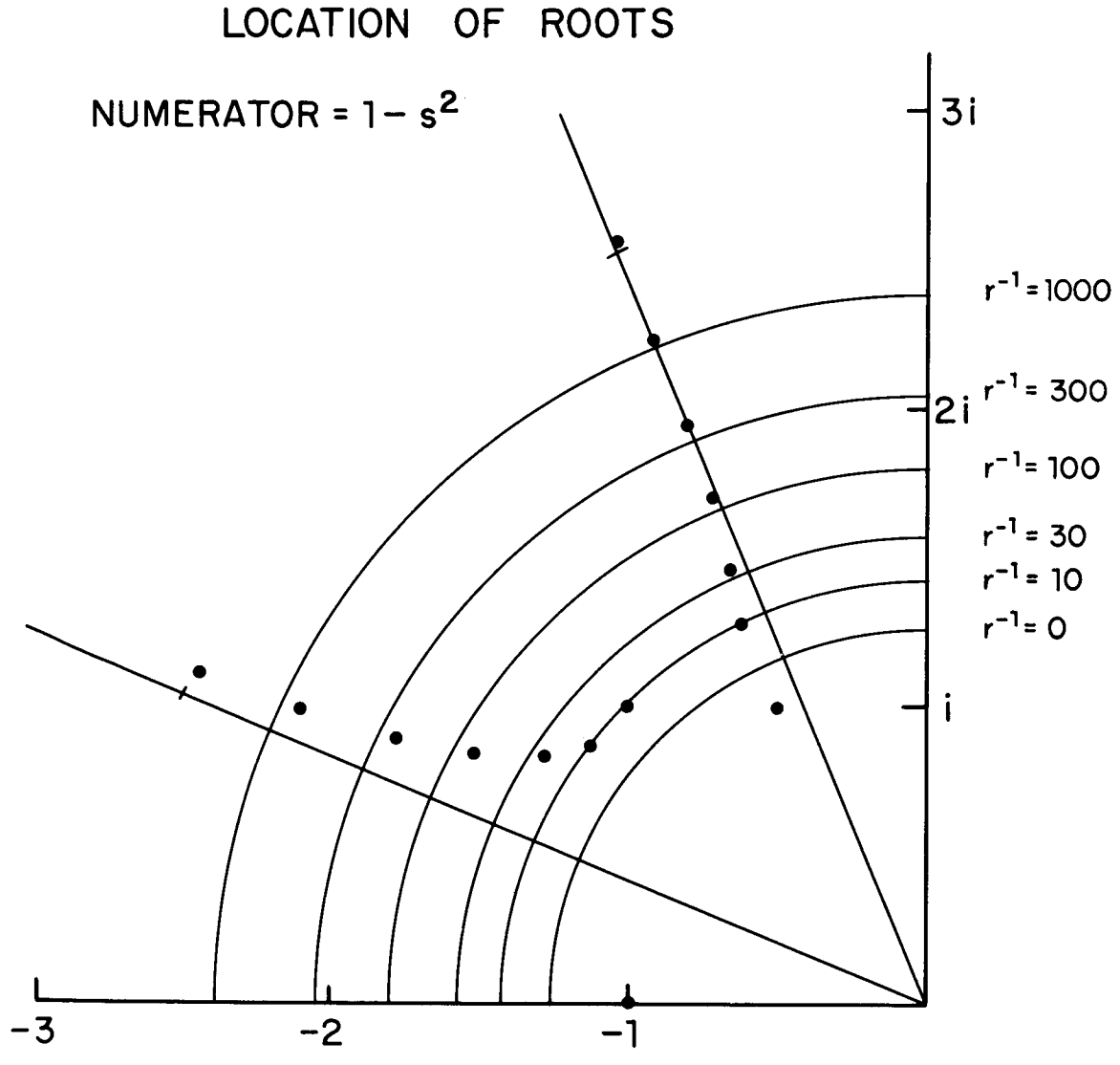
LOCATION OF ROOTS
NUMERATOR = 1 − s²
3i
2i
i
r⁻¹ = 1000
r⁻¹ = 300
r⁻¹ = 100
r⁻¹ = 30
r⁻¹ = 10
r⁻¹ = 0
-3
-2
-1

Fig. 3

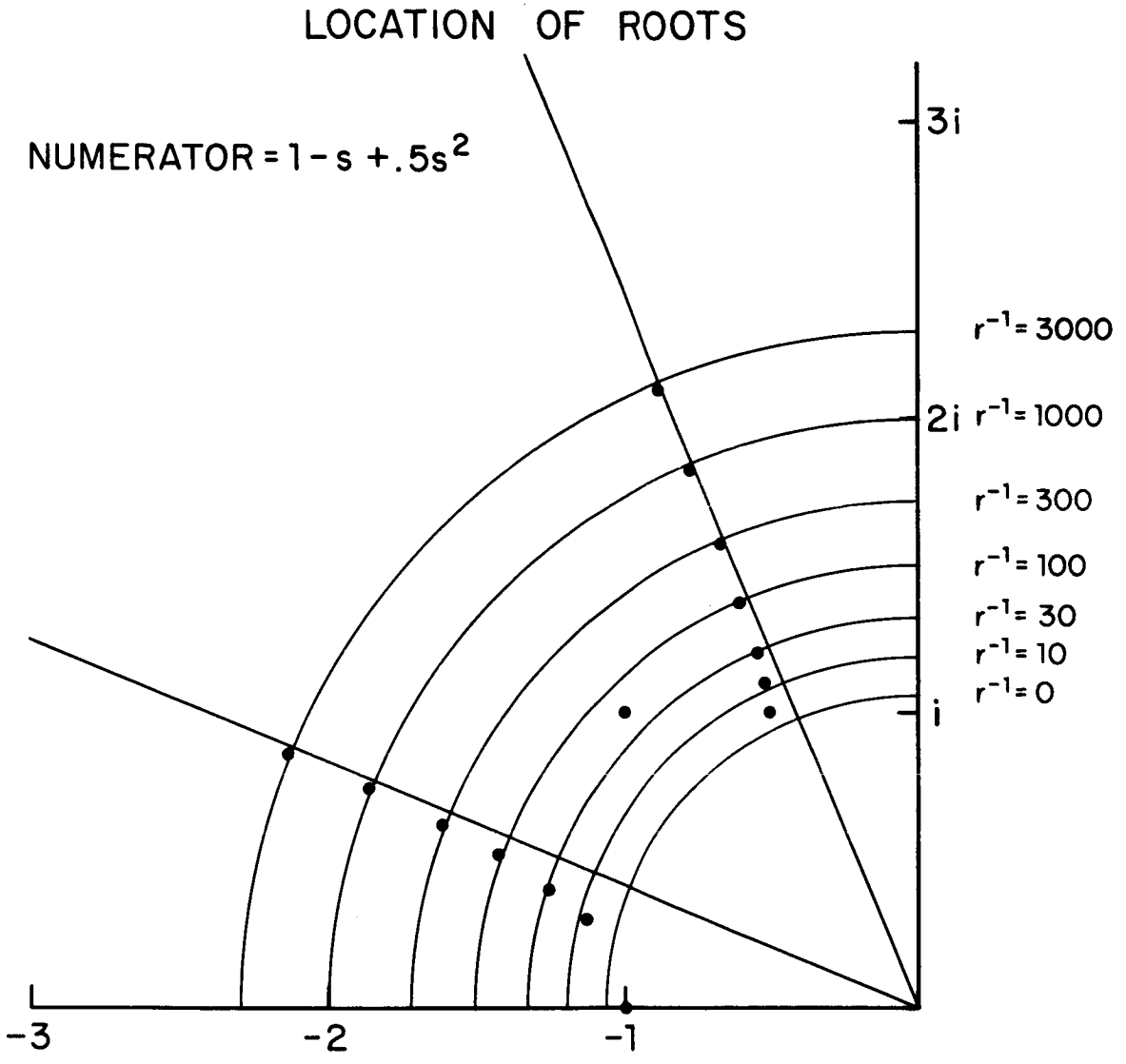
LOCATION OF ROOTS
NUMERATOR = 1 - s + .5s²
3i
2i
i
r⁻¹ = 3000
r⁻¹ = 1000
r⁻¹ = 300
r⁻¹ = 100
r⁻¹ = 30
r⁻¹ = 10
r⁻¹ = 0
-3
-2
-1

Fig. 4

Numerator = 1.

r^{-1}	τ	N	T	P_{ii}max	P_{ii} max normalized	i
.1	1.425	17	24.2	638	2020.	3
1.	1.286	16	20.6	213	213.	3
10.	.982	17	16.7	96	30.4	3
30.	.784	23	18.0	81	14.8	3
100	.563	28	15.8	77	7.7	3
300	.386	32	12.4	89	5.1	2
1000	.239	46	11.0	117	3.7	2
3000	.148	66	9.8	151	2.8	2

FIG. 5

CHAPTER XII

MATCHING DYNAMICS

1. Description of the Problem: We are interested here in the construction of a control system to force the output of a given system to have specified dynamics. Two interesting features of our specific example should be noted. (i) The process of matching dynamics will be simplified because in our given system

$$\begin{aligned} \dot{x} &= Fx + Gu, \\ y &= Hx, \end{aligned} \tag{1.1}$$

HG will be the zero matrix. (ii) Because the desired dynamics already match certain modes of (the open-looP) F, we will find that the performance index is not positive definite, but merely nonnegative definite.

2. Theory and References: Source material for the problem was obtained from Dynamic Optimization of Continuous Processes, J. G. Balchen, published at Trondheim, Norway by Institütt for Reguleringstenchnik (1961).

The method for obtaining specified dynamics will now be derived. We wish to change the lagrangian so as to minimize $\|\dot{y} - Ly\|_Q^2$ instead of $\|y\|_Q^2$. The idea is to make the output obey approximately the dynamics of some desired system

$$\dot{y} = Ly. \tag{2.1}$$

Notice that this differs from the so-called "model-follower" problem (see Chapter XVIII) in which we wish to minimize $\|y - y_d\|_Q^2$, where y_d is a solution of the system (2.1).

This problem then is to minimize the integral of

$$\|\dot{y} - Ly\|_Q^2 + \|u\|_R^2 = \|HFx + HGu - LHx\|_Q^2 + \|u\|_R^2.$$

We will for the moment make the simplifying assumption that HG is zero, as would be satisfied, for instance, by any impulse response matrix $T(t, \tau)$ such that $T(\tau, \tau)$ is the zero matrix, or, equivalently, by any transfer-function matrix $Z(s)$ in which every numerator has degree at least two less than its denominator. This gives us an equivalent system (with the usual lagrangian) which is unchanged except that a new output has been deinfed:

$$y_1 = (HF - LH)x. \tag{2.2}$$

Obviously the controllability has not been affected. So we need examine only the observability matrix,

$$[(F'H' - H'L'), F'(F'H' - H'L'), \ldots, F'^{n-1}(F'H' - H'L')]. \tag{2.3}$$

To decide whether or not (2.3) is completely observable requires rather advanced algebraic analysis. In the simplest case, we have the following result:

If $p = 1$, so that $L = \lambda$ (1×1 matrix), then (2.3) is completely observable iff $\lambda \neq$ eigenvalue of F.

This follows immediately if we observe that (2.3) is $(F - \lambda I)$ times the usual observability matrix

$$[H', F'H', \ldots, F'^{n-1}H']. \tag{2.4}$$

Now let us consider the changes in the riccati equation which take place because of the new lagrangian. We will no longer assume that HG is zero. We will however show that the performance index is still a quadratic form in $x(t_o)$ and formally derive the differential equation which the matrix of that form must satisfy.

We have the lagrangian

$$\mathcal{L} = \frac{1}{2} \|H(Fx + Gu) - LHx\|_Q^2 + \frac{1}{2} \|u\|_R^2$$

and the hamiltonian

$$\mathcal{H} = \min_u [\mathcal{L} + p'(Fx + Gu)].$$

This gives

$$u = - A^{-1}[G'p + Bx]$$

where

$$A = [N'QN + R], \quad N = HG$$

and

$$B = N'QM, \quad M = (HF - LH).$$

The hamilton-jacobi equation is

$$\frac{\partial V^o}{\partial t} + \mathcal{H} = 0,$$

where

$$2\mathcal{H} = \|Mx\|_Q^2 + 2p'Fx + \|G''p + N'QMx\|_{A^{-1}}^2.$$

Its solution can be given by assuming

$$V^o = \frac{1}{2} \|x\|_P^2,$$

where P is symmetric. This leads to the riccati equation

$$-\dot{P} = F'P + PF - PGA^{-1}G'P + M'QM - M'QNA^{-1}N'QM$$

$$- PGA^{-1}N'QM - M'QNA^{-1}G'P.$$

Collecting terms we see that this equation is identical with a standard riccati equation if the parameters of the latter are defined as

$$\hat{F} = F - GA^{-1}N'QM, \quad \hat{G} = G, \quad \hat{H} = M$$

$$\hat{R} = A$$

$$\hat{Q} = Q - QNA^{-1}N'Q.$$

Thus the new hamiltonian function fits into the standard computational scheme.

3. The Specific Problem. We are given the impulse response matrix

$$T(s) = \begin{bmatrix} \dfrac{1}{(s+1)(s+\frac{1}{2})(s+2)} & \dfrac{10}{(s+10)(s+\frac{1}{3})(s+\frac{1}{2})} & \dfrac{1}{(s+1)(s+\frac{1}{2})} \\ \dfrac{-s+\frac{3}{4}}{(s+\frac{1}{4})(s+1)^2(s+2)} & \dfrac{-4}{(s+5)(s+1)(s+2)} & \dfrac{4}{(s+1)(s+2)} \end{bmatrix}.$$

From this we obtain the time domain representation

$$T(t) = \begin{bmatrix} \frac{4}{3}e^{-\frac{1}{2}t} - 2e^{-t}, & +\frac{2}{3}e^{-2t} & \frac{180}{29}e^{-\frac{1}{3}t} - \frac{120}{19}e^{-\frac{1}{2}t} + \frac{60}{551}e^{-10t}, & -2e^{-\frac{1}{2}t} + 2e^{-t} \\ \frac{64}{63}e^{-\frac{1}{4}t} + \frac{5}{9}e^{-t} - \frac{7}{3}te^{-t} - \frac{11}{7}e^{-2t}, & -e^{-t} + \frac{4}{3}e^{-2t} - \frac{1}{3}e^{-5t}, & 4e^{-t} - 4e^{-2t} \end{bmatrix}$$

and then obtain a canonical realization by means of the system (1.1). (See pp. 340-356, final report NAS2-1107, July 1964).

We can tell a priori that the dimension of any realization is at least eight since there are seven eigenvalues one of which appears squared in $T(s)$ or, equivalently, multiplied by t in $T(t)$. A little further analysis shows

that the dimension is at least nine since the coefficient matrix associated with the eigenvalue -2 is of rank two. (ibid.)

Furthermore we can tell that the system is not completely controllable by any single input nor completely observable by any single output. This follows from the fact that not all eigenvalues appear in any column or row.

Using Method A, (Ibid.) , we can obtain the following canonical realization.

$$F = \text{diag}\left(-\frac{1}{4}, -\frac{1}{3}, -\frac{1}{2}, -2, -2, -5, -10, \begin{pmatrix} -1 & 1 \\ 0 & -1 \end{pmatrix}, -1\right)$$

$$H = \begin{bmatrix} 0 & \frac{180}{29} & 1 & 1 & 0 & 0 & \frac{60}{551} & 0 & \frac{6}{7} & 1 \\ \frac{64}{63} & 0 & 0 & 0 & 1 & 1 & 0 & 1 & -\frac{3}{7} & 0 \end{bmatrix}$$

$$G' = \begin{bmatrix} 1 & 0 & \frac{4}{3} & \frac{2}{3} & -\frac{11}{7} & 0 & 0 & -\frac{4}{9} & -\frac{7}{3} & 0 \\ 0 & 1 & -\frac{120}{19} & 0 & \frac{4}{3} & -\frac{1}{3} & 1 & -1 & 0 & 0 \\ 0 & 0 & -2 & 0 & -4 & 0 & 0 & 4 & 0 & 2 \end{bmatrix}$$

This realization was checked by comparing the step response computed by the ASP program appearing in Fig. 1 with that shown in Balchen p. 38 and also by checking the terminal values of the response with those computed directly from T(s).

We computed the controllability matrix using u_1 only, in the program appearing in Fig. 2. This should give us a matrix of rank five; it is evident from the placement of zeroes in the first column of G that we will lose $-\frac{1}{3}$, -5, -10, and -1 and it is clear from elementary controllability theory that we must lose also one of the eigenvalues -2. This demonstrates the advantages of having the system in canonical form, for the question is not so easily answered by numerical means.

Let us examine the situation in some detail for the insight it may give in the general problem of rank determination. Four rows and columns of W will be identically zero and thereby simplify the problem. We let p_1 and p_2 (see Chapter V) both be 10^{-4}, thus helping to ensure that most of the possible ranks will be considered and their errors printed. The times referred to are the times over which the controllability matrix integral is calculated.

At .2 W was so poorly conditioned that in spite of the small p_2, the rank was given as two. The first pivotal element was 1.34, the third was $1 \cdot 10^{-6}$.

At .4 the first pivotal element was 3.34, the fifth was $1 \cdot 10^{-8}$ and the sixth was negative. The errors however, after changing p_2 to 1, were

rank 2	$.6 \cdot 10^{-5}$
3	$1. \cdot 10^{-3}$
4	.6
5	.54

thus indicating that the pseudo-inversion error is not a satisfactory test for determining rank.

At .6 the first pivotal element was 6.3, the fifth was $.2 \cdot 10^{-7}$ and the sixth was negative. The errors were

rank 3	$.5 \cdot 10^{-3}$
4	.3
5	2.

At .8 the first pivotal element was 14.5, the fifth was $.5 \cdot 10^{-7}$, the sixth was $.14 \cdot 10^{-7}$ both well down in the noise level of the initial

pivot. The errors were

rank 3	$.1 \cdot 10^{-3}$
4	$.3 \cdot 10^{-1}$
5	.6
6	8.

This pattern then continues for as long as W was computed.

Very similar results were obtained when the controllability matrix was computed using all three inputs except that the difficulties increased because of the extra four dimensions and because the spread of eigenvalues which was $(-\frac{1}{4}, -2)$ when u_1 was used, becomes $(-\frac{1}{4}, -10)$ when all three are used. Since the elements of W are roughly proportional to $e^{-(\lambda_i + \lambda_j)t}$ this increase in spread can be expected to cause difficulties.

At .2 the first pivot was 10, the tenth $.2 \cdot 10^{-8}$. The minimum error of $.4 \cdot 10^{-4}$ occurred at a rank of 6.

At .4 the first pivot was 149, the tenth $.2 \cdot 10^{-6}$. The minimum error of $.6 \cdot 10^{-4}$ occurred at a rank of 6. These minima were not very definite

5	$.1 \cdot 10^{-3}$	$.9 \cdot 10^{-4}$
6	$.4 \cdot 10^{-4}$	$.3 \cdot 10^{-4}$
7	$.3 \cdot 10^{-3}$	$.7 \cdot 10^{-4}$.

This pattern continued with the smallest error slipping back to smaller ranks until abruptly at 1.4 p_2 could not force the error smaller and only two pivots were made on the -10 and -5 eigenvalues, the two pivotal elements being $1 \cdot 10^{10}$ and $.2 \cdot 10^{3}$. These numbers completely obscured the contributions of smaller eigenvalues.

4. Equations and Procedure. We now wish to design an optimal control so that y_1 behaves as $\dot{y}_1 = -y_1$ and y_2 behaves as $\dot{y}_2 = -2y_2$.

This gives us the performance index

$$\int_0^\infty [(\dot{y}_1 + y_1)^2_{q_{11}} + (\dot{y}_2 + 2y_2)^2_{q_{22}} + u'Ru]dt \tag{1.1}$$

where L can be rewritten as

$$L = \|(HF - LH)x\|^2_Q + \|u\|^2_R$$

where $Q = \text{diag}(q_{11}, q_{22})$ and $L = \text{diag}(-1, -2)$.

This simple form for L occurs because $HG = [0]_{2,3}$*. We know this is correct because each denominator in T(s) is of degree at least two greater than its numerator, equivalently $T(t)\big|_{t=0} = HG = [0]_{2,3}$.

Because of this, we need only define $\hat{H} = HF - LH$ and then compute the gains for the optimal regulators of the system $[\hat{H}, F, G]$.

However, because F and L have two common eigenvalues, the system $[\hat{H}, F]$ has an observable space of dimension eight. We expect therefore to obtain a performance index matrix P(0) of rank eight and to find that the vectors

$$[0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

and

$$[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]$$

are costless (lie in the null space of P(0)). This will be true only if the vectors lie in the null space of $P(\infty)$, but we are letting $P(\infty) = [0]_{10,10}$.

---

*$[0]_{m,n}$ means $m \times n$ zero matrix

This follows from the fact that these vectors will give the correct response without control effect a fact which can be seen formally from the two zero columns of

$$HF - LH = \begin{bmatrix} 0 & \frac{120}{29} & \frac{1}{2} & -1 & 0 & 0 & -\frac{540}{551} & 0 & 0 & 0 \\ \frac{16}{9} & 0 & 0 & 0 & 0 & -3 & 0 & 1 & \frac{4}{7} & 0 \end{bmatrix}$$

5. Results: Using the program appearing in Fig. 3, the performance index and optimal gain matrices were computed (Fig. 4 and 5). Fig. 6 illustrates $y_{1,10}(t)$ which is uncontrolled, being the desired response e^{-t} anyway, together with $y_{1,4}(t)$ which, uncontrolled would have the response e^{-2t}, also illustrated. These results were obtained using the program in Fig. 7.

```
        BEGIN
        LOAD    F, G, H, T,Z1,ND,
HEAD 1EAT       F,Z1,PH,IN
        MULT    H  IN      1
        MULT    1  G       SR
        RINT    Z1         SR STR
        ADD     Z1 T       Z1
        IF      ND Z1 HEAD 1
        END
F               10      10
        -.25                    0       0       0       0
                0               0       0       0       0
                0  -.333333333          0       0       0
                0               0       0       0       0
                0               0    -.5        0       0
                0               0       0       0       0
                0               0       0    -2.        0
                0               0       0       0       0
                0               0       0       0    -2.
                0               0       0       0       0
                0               0       0       0       0
             -5.                0       0       0       0
                0               0       0       0       0
                0          -10.         0       0       0
                0               0       0       0       0
                0               0    -1.     1.         0
                0               0       0       0       0
                0               0       0    -1.        0
                0               0       0       0       0
                0               0       0       0    -1.
G               10      3
     1.                         0       0       0      1.
                0  1.3333333       -6.3157895          -2.  .66666667
                0               0  -1.5714285  1.3333333          -4.
                0  -.33333333              0       0      1.
                0  -.44444444           -1.          4.  -2.3333333
                0               0       0       0      2.
H               2       10
                0  6.2068966            1.      1.         0
                0   .10889292           0   .85714286      1.
1.0158730                   0       0       0      1.
                1.              0       1.  -.42857143         0
T               1       1
        .2
Z1              1       1
                0
ND              1       1
        16.
```

Fig. 1

```
        BEGIN
        LOAD  P1,P2,Z1,PC,D2,D3,T1,ND,ZR,WO, I,RI, F, G,
        WRITE I  THIS PROGRAM COMPUTES THE CONTROLLABILITY MATRIX FOR THE
        WRITE  SYSTEM
        WRITE       ------- XDOT = F * X + G * U
        PIZER P1,P2,
        TRANP  G,    GT,
        MULT  RI,GT,     C,
        MULT   G, C,     Q,
        NORM   F,    NF,
        NORM   Q,    NQ,
        PSEUO NQ,    NQ,RK
        MULT  NF,NQ,    LM,
        MULT  LM, Q,     Q,
        MULT  LM,WO,    WO
        TRANP  F,    FT,
        SUBT  ZR, F,    MF,
        JUXTC FT,ZR,     1,
        JUXTC  Q,MF,     2,
        JUXTR  1, 2,     3,
        RINT   3, 3
        ETPHI  3,T1,PH,T9,
        MULT  T9,D2,    D4,
        MULT  T1,D3,    D5,
        ADD   D4,D5,     D,
        PSEUO LM,    LN,RK,
HEAD 1RICAT WO,PH, C, D,PC,   WO, K,A1,
        ADD   Z1,T1,    Z1,
        PSEUO WO,+ 1WI,AL,PRINT
        MULT  LN,WO,     W,
        RINT  Z1,     W, W
        IF    ND,Z1,HEAD 1
        END
      P1      1       1
          .0001
      P2      1       1
            1.E-4
      Z1      1       1
              0
      PC      1       4
              0                0               0               0
      D2      1       3
              0                1.              0
      D3      1       3
              0                0               1.
      T1      1       1
                .2
      ND      1       1
               3.
ZR          10      10
                0                0               0               0               0
                0                0               0               0               0
                0                0               0               0               0
```

Fig. 2

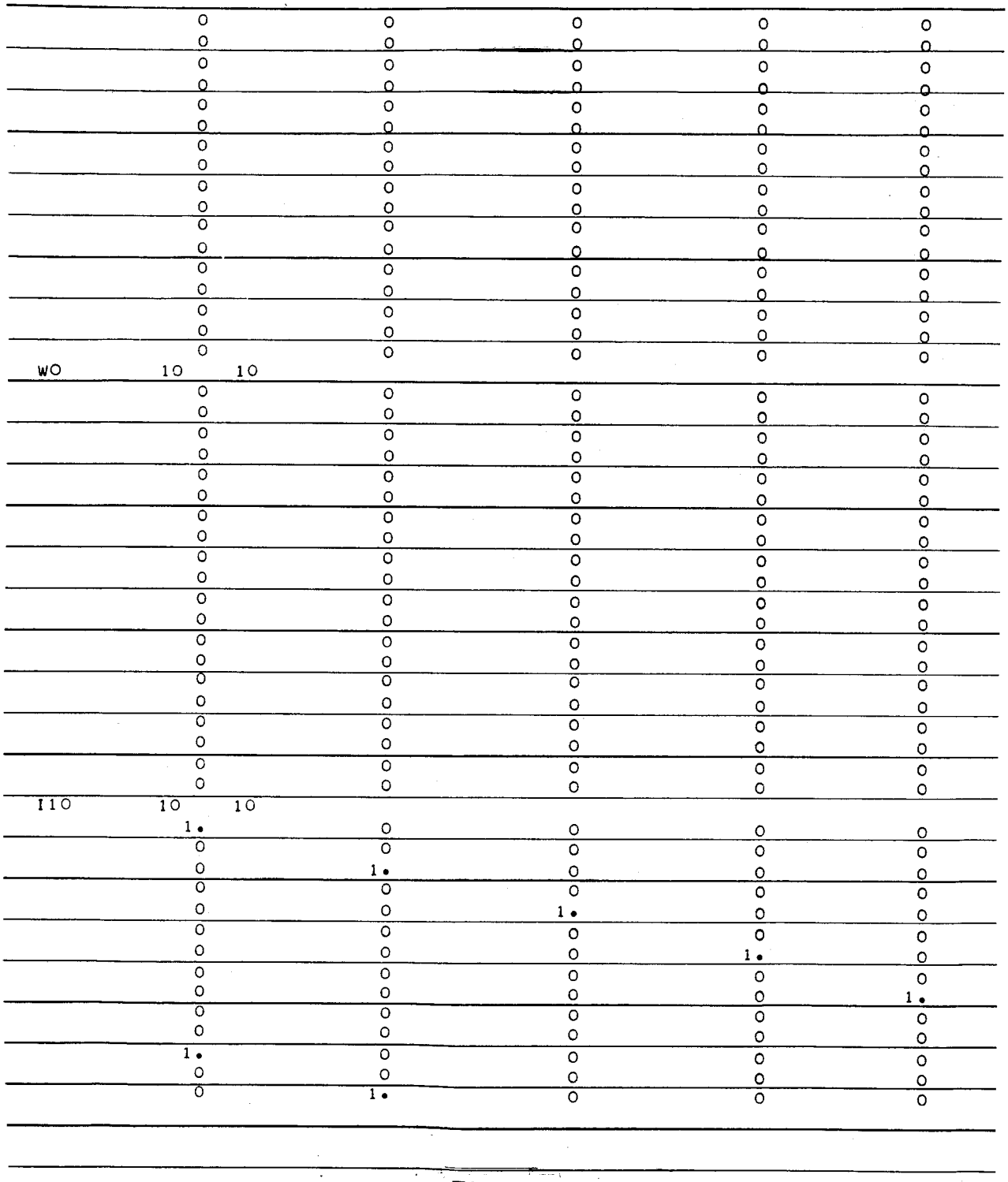

Fig. 2

```
                0          0          0          0          0
                0          0          1.         0          0
                0          0          0          0          0
                0          0          0          1.         0
                0          0          0          0          0
                0          0          0          0          1.
11        1         1
                1.
F         10        10
          -.25                        0          0          0          0
                0                     0          0          0          0
                0    -.333333333                 0          0          0
                0                     0          0          0          0
                0                     0          -.5        0          0
                0                     0          0          0          0
                0                     0          0          -2.        0
                0                     0          0          0          0
                0                     0          0          0          -2.
                0                     0          0          0          0
                0                     0          0          0          0
          -5.                         0          0          0          0
                0                     0          0          0          0
                0          -10.       0          0          0          0
                0                     0          0          0          0
                0                     0          -1.        1.         0
                0                     0          0          0          0
                0                     0          0          -1.        0
                0                     0          0          0          0
                0                     0          0          0          -1.
G         10        1
                1.                    0    1.3333333        .66666667  -1.5714285
                0                     0    -.44444444       -2.3333333            0
```

Fig. 2

```
        BEGIN
        LOAD   TT, D,D1,PC, F, G, Q,RI,ZR,T1,XO,ND,R1,RZ,XX, H,Q1,
        MULT   Q1, H,     8
        TRANP  H,    HT
        MULT   HT  8,     Q
        SUBT   ZR, F,     1,
        TRANP  F,    FT,
        TRANP  G,    GT,
        MULT   RI,GT,     C,
        MULT    G, C,     5,
        NORM    Q,    NQ,
        NORM    5,    N5,
        PSEUO NQ,    PQ,RK
        MULT   PQ,N5,    M1,
        DECOM M1,     S,SJ,ER,PE, E,RK,
        MULT    S, Q,     Q,
        MULT   SJ, 5,     5,
        MULT   SJ, C,     C,
        BLOT   NQ,
        JUXTC   1, 5,     2,
        JUXTC   Q,FT,     3,
        JUXTR   2, 3,    PH,
        NORM   PH,    NP,
        PSEUO NP,    NP,RK
        MULT   NP,TT,     T
        RINT   PH,PH
        ETPHI PH, T,PH
        MULT    T,D1,     4,
        ADD     4, D,     D,
        RICAT   Q,PH, C, D,PC,XX, P, K,AL,
        RINT    P,PER K, K
        IF     AL, D,ZILCH
        WRITE        THE PRECEDING MATRICES WERE THE MATRIX P OF THE
        WRITE     PERFORMANCE INDEX AND THE FEEDBACK GAIN MATRIX K.
        MULT    G, K,    GK,
        SUBT    F,GK,    CF,
        ETPHI CF,T1,P1,
HEAD 1MULT    K,XO,    KX,
        JUXTR XO,RZ,     6,
        JUXTR  6,KX,     7,
        RINT   R1,     7, X
        MULT   P1,XO,    XO,
        ADD    R1,T1,    R1,
        IF     ND,R1,HEAD 1
        IF     TT,TT,HEAD 2
ZILCH WRITE     THE RICCATI SOLUTION HAS FAILED TO CONVERGE IN 1000 STEPS
HEAD 2END
TT              1       1
                8.
 D              1       3
          .00001                  0               0
D1              1       3
                0               -1.          -1000.
```

- 3 Fig. 3

```
PC         1    4
         100.          100.          1.           1.
F         10   10
  -.25                 0             0            0            0
          0            0             0            0            0
          0  -.333333333             0            0            0
          0            0             0            0            0
          0            0           -.5            0            0
          0            0             0            0            0
          0            0             0          -2.            0
          0            0             0            0            0
          0            0             0            0          -2.
          0            0             0            0            0
          0            0             0            0            0
      -5.              0             0            0            0
          0            0             0            0            0
          0          -10.            0            0            0
          0            0             0            0            0
          0            0           -1.           1.            0
          0            0             0            0            0
          0            0             0          -1.            0
          0            0             0            0            0
          0            0             0            0          -1.
G         10   3
   1.                  0             0            0           1.
          0  1.3333333      -6.3157895          -2.   .66666667
          0            0    -1.5714285    1.3333333          -4.
          0  -.33333333              0            0           1.
          0  -.44444444            -1.           4.  -2.3333333
          0            0             0            0           2.
Z10       10   10
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
          0            0             0            0            0
R1         1    1
    1.
```

Fig. 3

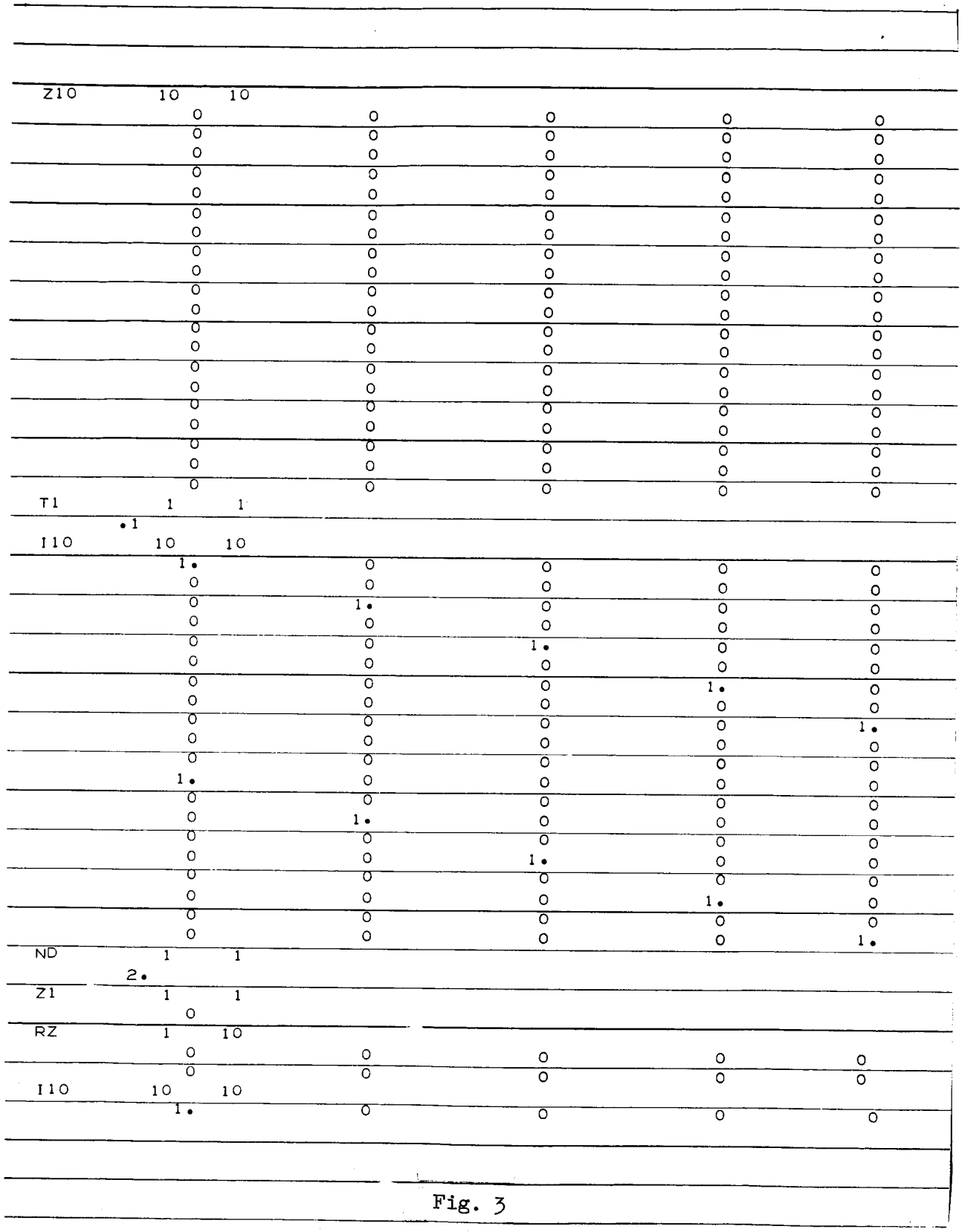

Fig. 3

```
                0                0                0                0                0
                0               1.                0                0                0
                0                0                0                0                0
                0                0               1.                0                0
                0                0                0                0                0
                0                0                0               1.                0
                0                0                0                0               1.
                0                0                0                0                0
                0                0                0                0                0
               1.                0                0                0                0
                0                0                0                0                0
                0               1.                0                0                0
                0                0                0                0                0
                0                0               1.                0                0
                0                0                0                0                0
                0                0                0               1.                0
                0                0                0                0                0
                0                0                0                0               1.
H               2   10
                0  4.1379310                    .5              -1.               0
                0  -.98003630                   0                0                0
 1.7777778                    0                0                0                0
               -3.               0               1.  .57142857                     0
Q1              2   2
              100.               0               0            100.
```

Fig. 3

MATRIX PER

NUMBER OF ROWS 10 NUMBER OF COLUMNS 10

0.18651055E 01 -0.38178767E-00	0.47299859E 01 0.98661084E 00	0.55695709E 00 0.63437553E 00	-0.89963835E 00 0.27012783E-09	0.64702535E-07	-0.24020306E 01
0.47299859E 01 -0.47560064E 01	0.47434569E 02 0.24163837E 01	0.56289651E 01 0.17040643E 01	-0.95265312E 01 -0.16740675E-09	0.44384938E-06	-0.43967746E 01
0.55695709E 00 -0.57076416E 00	0.56289651E 01 0.28632194E-00	0.66866959E 00 0.20029116E-00	-0.11370701E 01 -0.21195296E-10	0.52097808E-07	-0.52538913E 00
-0.89963835E 00 0.10664047E 01	-0.95265312E 01 -0.47458576E-00	-0.11370701E 01 -0.31721341E-00	0.20001247E 01 -0.16592643E-12	-0.80944704E-07	0.93295726E 00
0.64702535E-07 -0.31629325E-07	0.44384938E-06 0.31445590E-07	0.52097808E-07 0.23848388E-07	-0.80944704E-07 0.56151135E-19	-0.16048726E-14	-0.52262197E-07
-0.24020306E 01 0.47443745E-00	-0.43967746E 01 -0.13330194E 01	-0.52538913E 00 -0.78997989E 00	0.93295726E 00 -0.17066253E-13	-0.52262197E-07	0.36009959E 01
-0.38178767E-00 0.71583210E 00	-0.47560064E 01 -0.20606322E-00	-0.57076416E 00 -0.12981575E-00	0.10664047E 01 -0.23568652E-11	-0.31629325E-07	0.47443745E-00
0.98661084E 00 -0.20606322E-00	0.24163837E 01 0.53419448E 00	0.28632194E-00 0.33308635E-00	-0.47458576E-00 -0.10393154E-10	0.31445590E-07	-0.13330194E 01
0.63437553E 00 -0.12981575E-00	0.17040643E 01 0.33308635E-00	0.20029116E-00 0.21782497E-00	-0.31721341E-00 0.34073678E-10	0.23848388E-07	-0.78997989E 00
0.27012783E-09 -0.23568652E-11	-0.16740675E-09 -0.10393154E-10	-0.21195296E-10 0.34073678E-10	-0.16592643E-12 -0.45105954E-15	0.56151135E-19	-0.17066253E-13

Fig. 4

MATRIX K

NUMBER OF ROWS 3 NUMBER OF COLUMNS 10

0.54101467E 00	0.50563037E 01	0.58112279E 00	-0.79542577E 00	0.64138372E-07	-0.27179241E-00
-0.22660026E-00	0.22647130E-00	0.20402217E-00	0.10115046E-08		
0.39075574E 01	0.37438980E 02	0.43875806E 01	-0.67588778E 01	0.41919892E-06	-0.28574571E 01
-0.23484520E 01	0.18919291E 01	0.14517066E 01	-0.15456941E-09		
0.17169672E 02	-0.96524820E 01	-0.11641411E 01	0.22779363E 01	0.13085039E-06	-0.25951548E 02
0.19231991E 01	0.94811624E 01	0.56479795E 01	0.49527773E-11		

THE PRECEDING MATRICES WERE THE MATRIX P OF THE PERFORMANCE INDEX AND THE FEEDBACK GAIN MATRIX K.

Fig. 5

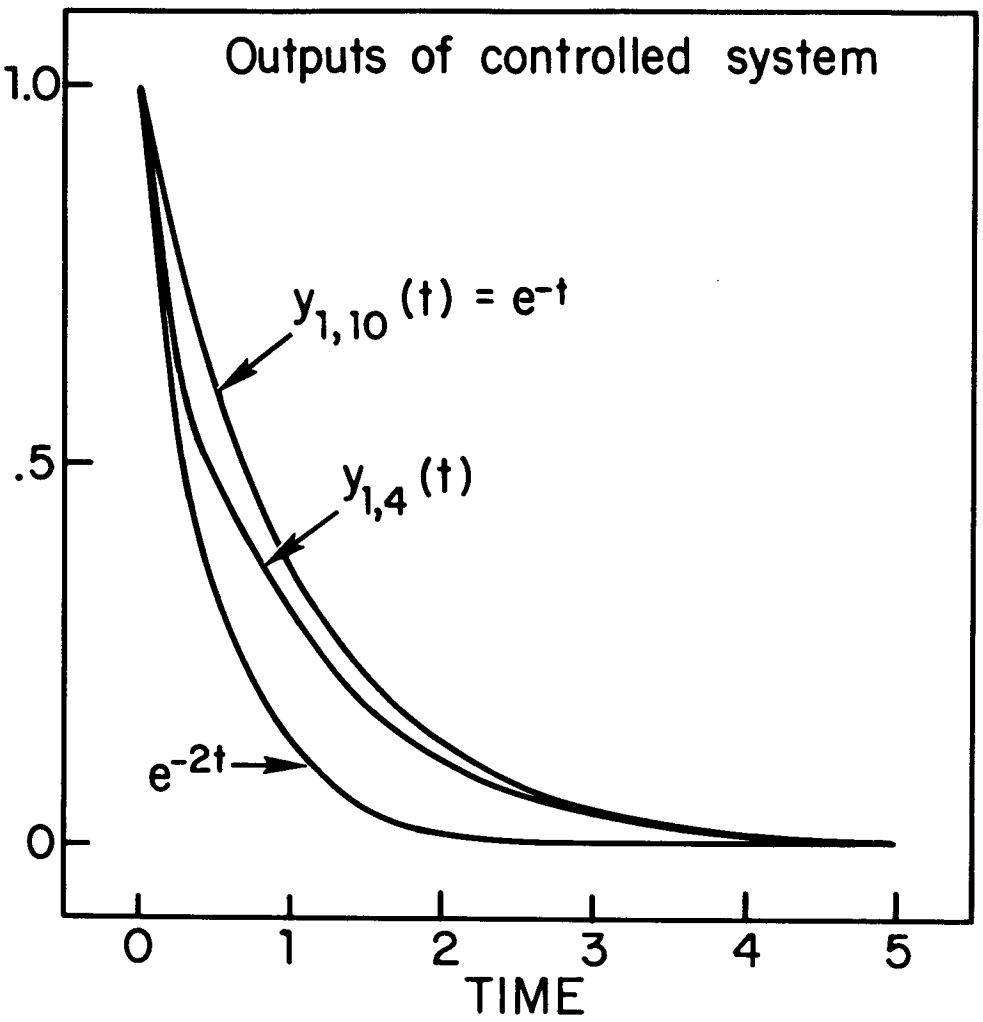
Outputs of controlled system
1.0
.5
0
$y_{1,10}(t) = e^{-t}$
$y_{1,4}(t)$
e^{-2t}
0
1
2
3
4
5
TIME

Fig. 6

```
        BEGIN
        LOAD    F, G, H, T,Z1,ND, K  I
        MULT    G, K,    GK
        SUBT    F,GK,    CF
        ETPHI CF, T,PH
HEAD 1MULT    H, I,     Y
        RINT  Z1,     Y, Y
        MULT  PH, I,    I
        ADD   Z1, T,    Z1,
        IF    ND,Z1,HEAD 1
        END
F           10     10
      -.25                    0          0          0          0
                0             0          0          0          0
                0  -.333333333             0          0          0
                0             0          0          0          0
                0             0       -.5           0          0
                0             0          0          0          0
                0             0          0       -2.           0
                0             0          0          0          0
                0             0          0          0       -2.
                0             0          0          0          0
                0             0          0          0          0
             -5.              0          0          0          0
                0             0          0          0          0
                0        -10.            0          0          0
                0             0          0          0          0
                0             0       -1.       1.            0
                0             0          0          0          0
                0             0          0       -1.           0
                0             0          0          0          0
                0             0          0          0       -1.
G           10      3
     1.                       0          0          0         1.
                0  1.3333333     -6.3157895          -2.  .66666667
                0             0  -1.5714285  1.3333333         -4.
                0  -.33333333               0          0         1.
                0  -.44444444            -1.        4.  -2.3333333
                0             0          0          0         2.
H           2      10
                0  6.2068966               1.         1.         0
                0   .10889292               0   .85714286         1.
  1.0158730                   0          0          0         1.
                1.            0          1.  -.42857143           0
T           1      1
        .2
Z1          1      1
                0
ND          1      1
        16.
K           3      10
   .54101467        5.0563037     .58112279     -.79542577     .64138372E-7
  -.27179241       -.22660026      .2264713      .20402217     .10115046E-8
```

Fig. 7

3.9075574	37.43898	4.3875866	-6.7588778	.41919892E-6
-2.8574571	-2.348452	1.8919291	1.4517066	-.15456941E-9
17.169672	-9.652482	-1.1641411	2.2779363	.13085039E-6
-25.951548	1.9231991	9.4811624	5.6479795	.49527773E-11

I10 10 10

1.	0	0	0	0
0	0	0	0	0
0	1.	0	0	0
0	0	0	0	0
0	0	1.	0	0
0	0	0	0	0
0	0	0	1.	0
0	0	0	0	0
0	0	0	0	1.
0	0	0	0	0
1.	0	0	0	0
0	0	0	0	0
0	1.	0	0	0
0	0	0	0	0
0	0	1.	0	0
0	0	0	0	0
0	0	0	1.	0
0	0	0	0	0
0	0	0	0	1.

Fig. 7

CHAPTER XIII

AN OPTIMAL CONTINUOUS TIME FILTER

1. Description of the Problem: We shall consider a special data-smoothing problem encountered in determining the position and velocity of a space vehicle. This will provide a convenient illustration of the hamiltonian technique for obtaining solutions of the variance equation. The example shows also how to obtain the model of the random process directly from physical considerations.

The physical picture is as follows. The position of a satellite is measured by means of a radio signal. It is assumed that the measurement contains additive noise which may be taken to be approximately gaussian and white relative to the bandwidth of the satellite motion. A second measurement of the satellite motion is available from an accelerometer. This reading is also subject to noise; but here the noise is due to drift and other very slowly varying effects, and may be considered to be a constant random variable during the interval of interest. The motion of the satellite is linearized and assumed to be one-dimensional, and subject to a constant, gaussian random acceleration.

The problem is to design an optimal filter which provides the best running estimates of the position and velocity of the satellite based on the two types of measurement noise and the variance of the acceleration.

2. Theory and References: For a derivation of the variance equation, see Chapter V.

The problem was suggested by the follpwing paper:

[1] E. L. Peterson, "Optimization of multi-input time-varying systems subject to multiple or redundant nonstationary inputs". Proc. First International Congress on Automatic Control, (Moscow 1960) Butterworths 1961. See also:

[2] R. E. Kalman, Discussion of Peterson's paper, in the same Proceedings.

3. The Specific Problem: The assumptions are formalized by setting up a model for the message process. Let z_1 denote the radio signal and a_1 the reading of the accelerometer. Both signals are supposed to be known exactly. The equation of motion (linearized, one-dimensional, with unit mass) is

$$\ddot{x} = a(t) = \text{acceleration} = \text{constant} = a$$

where a is a gaussian random variable with zero mean. The accelerometer measures a plus a constant gaussian random variable b with zero mean (the bias error of the accelerometer):

$$a_1 = a + b.$$

Let $Ea^2 = r_a$ and $Eb^2 = r_b$. We assume that a and b are uncorrelated (hence, by gaussianness, independent) and set

$$\rho = \frac{r_a r_b}{r_a + r_b}.$$

We replace a_1, b by two new random variables u_1 and x_3 which are orthogonal to each other (and thus, by gaussianness, independent); u_1 is exactly known and x_3 is to be estimated:

$$u_1 = \frac{\rho}{r_b} a_1, \quad x_3 = \frac{\rho}{r} a_1 - b.$$

Then the equations of motion are:

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = a = a_1 - b = u_1 + x_3.$$

The model is now fully described, and we have:

$$(3.1)\qquad F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \text{and} \quad H = [1 \quad 0 \quad 0].$$

The variance equations are

$$(3.2)\qquad \begin{aligned} d\sigma_{11}/dt &= 2\sigma_{12} - \sigma_{11}^2/r_{11}, \\ d\sigma_{12}/dt &= \sigma_{13} + \sigma_{22} - \sigma_{11}\sigma_{12}/r_{11}, \\ d\sigma_{13}/dt &= \sigma_{23} - \sigma_{11}\sigma_{13}/r_{11}, \\ d\sigma_{22}/dt &= 2\sigma_{23} - \sigma_{12}^2/r_{11}, \\ d\sigma_{23}/dt &= \sigma_{33} - \sigma_{12}\sigma_{13}/r_{11}, \\ d\sigma_{33}/dt &= -\sigma_{13}^2/r_{11}. \end{aligned}$$

The Hamiltonian equations are

$$(3.3)\qquad \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{p}_1 \\ \dot{p}_2 \\ \dot{p}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1/r_{11} & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1. & 0 \\ 0 & 0 & 0 & 0 & 0 & 1. \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}.$$

The transition matrix corresponding to these equations is easily found. (The sixth power of the matrix on the right-hand side of (3.3) is zero.) The result is:

$$(3.4)\quad \Theta(t,0) = \begin{bmatrix} 1 & 0 & 0 & t/r_{11} & t^2/2r_{11} & t^3/6r_{11} \\ -t & 1 & 0 & -t^2/2r_{11} & -t^3/6r_{11} & -t^4/24r_{11} \\ t^2/2 & -t & 1 & t^3/6r_{11} & t^4/24r_{11} & t^5/120r_{11} \\ 0 & 0 & 0 & 1 & t & t^2/2 \\ 0 & 0 & 0 & 0 & 1 & t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

We assume that the initial value of $\Sigma(0)$ is: $\sigma_{11}(0) = \sigma_{22}(0) = 0$, while $\sigma_{33}(0) = \rho$ is the effect due to the bias in the reading of the accelerometer. (Of course, all off-diagonal terms of $\Sigma(0)$ are zero.) Using (3.4) we find that the solution of the variance equation corresponding to these initial conditions is:

$$\Sigma(t) = \frac{r_{11}}{t^5/20 + r_{11}/\rho} \begin{bmatrix} t^4/4 & t^3/2 & t^2/2 \\ t^3/2 & t^2 & t \\ t^2/2 & t & 1 \end{bmatrix}.$$

It is easily verified by direct substitution that this is indeed a solution of the variance equation which satisfies the initial conditions stated above.

The optimal time-varying gains can now be obtained at once from the relation $K(t) = \Sigma(t)H'R^{-1}$; they are:

$$k_{11}(t) = t^4/4\alpha(t),\ k_{21}(t) = t^3/2\alpha(t),\ k_{31}(t) = t^2/2\alpha(t);$$

where

$$\alpha(t) = t^5/20 + r_{11}/\rho.$$

$$(3.5)\qquad \begin{bmatrix} d\hat{x}_1/dt \\ d\hat{x}_2/dt \\ d\hat{x}_3/dt \end{bmatrix} = \begin{bmatrix} -t^4/4\alpha & 1 & 1 \\ -t^3/2\alpha & 0 & 1 \\ -t^2/2\alpha & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \end{bmatrix} + \begin{bmatrix} t^4/4\alpha \\ t^3/2\alpha \\ t^2/2\alpha \end{bmatrix} z_1 + \begin{bmatrix} 0 \\ u_1 \\ 0 \end{bmatrix}.$$

This differential equation is difficult to solve. Considerable simplification is obtained by introducing a new set of state variables:

$$w_1 = \hat{x}_3 \qquad 2\hat{x}_1 = t^2 w_1 + w_2 + t w_3$$
$$w_2 = 2\hat{x}_1 - t\hat{x}_2 \qquad \hat{x}_2 = t w_1 + w_3,$$
$$w_3 = \hat{x}_2 - t\hat{x}_3, \qquad \hat{x}_3 = w_1.$$

Then by (3.5)

$$\begin{bmatrix} dw_1/dt \\ dw_2/dt \\ dw_3/dt \end{bmatrix} = \begin{bmatrix} -t^4/4\alpha & -t^2/4\alpha & -t^3/4\alpha \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \begin{bmatrix} t^2/2\alpha \\ 0 \\ 0 \end{bmatrix} z_1 + \begin{bmatrix} 0 \\ -t \\ 1 \end{bmatrix} u_1.$$

The transition matrix corresponding to this equation is

$$\Psi^{(w)}(t, \tau) = \begin{bmatrix} \beta/\alpha & -\delta/2\alpha & -(\gamma - \delta\tau/2)/\alpha \\ 0 & 1 & t - \tau \\ 0 & 0 & 1 \end{bmatrix}$$

where

$$\beta(\tau) = \tau^5/20 + r_{11}/\rho,\ \gamma(t, \tau) = (t^4 - \tau^4)/8,\ \delta(t, \tau) = (t^3 - \tau^3)/6.$$

Thus the transition matrix corresponding to (3.5) is found to be

$$\Psi^{(\hat{x})}(t\ ,\ \tau) = \begin{bmatrix} \alpha-\delta t^2/2 & (t-\tau)\alpha-(\gamma-\delta\tau)\tau^2/2 & (\tau^2/2-t\tau)\alpha+(\beta+\gamma\tau-\delta\tau^2/2)t^2/2 \\ -\delta t & \alpha-(\gamma-\delta\tau)t & -\alpha\tau + (\beta + \gamma\tau - \delta\tau^2/2)t \\ -\delta & -(\gamma-\delta\tau) & \beta + \gamma\tau - \delta\tau^2/2 \end{bmatrix}$$

4. Results: Using the program appearing in Fig. 1 we obtained the covariance matrix for the system (3.1) with $Q = 0$ and $R = .5$.

At $t = .95$, the variance matrix should have been:

$$\Sigma = \begin{bmatrix} .352674 & .74247275 & .78155026 \\ & 1.5631005 & 1.6453690 \\ \text{sym.} & & 1.7319673 \end{bmatrix}$$

and actually was computed to be

$$\Sigma = \begin{bmatrix} .35267451 & .7424271 & .78155027 \\ & 1.5631005 & 1.6453691 \\ \text{sym.} & & 1.7319676 \end{bmatrix}.$$

The transition matrix of the optimal filter as computed by sampling the optimal filter gain at intervals of .05 was, at $t = 1$

$$\Phi(1,\ 0) = \begin{bmatrix} .72218 & .79251 & .41728 \\ -.55489 & .58507 & .83441 \\ -.55461 & -.41522 & .83412 \end{bmatrix}$$

and should have been

$$\Phi(1,\ 0) = \begin{bmatrix} .722222 & .791666 & .416666 \\ -.555555 & .583333 & .833333 \\ -.555555 & .416666 & .833333 \end{bmatrix}.$$

This shows that the sampling is being done correctly. Accuracy of the transient can be improved by increasing the sampling rate.

```
          JAN. 26. 1965
          BEGIN
          LOAD  .5.Z1.D2.PC. T.D3. I.V0.QN.GS.ZR.FS.HS.RN.ND.
          MULT  RN.HS.    21.
          TRANP HS.    HT.
          MULT  HT.21.    22.
          TRANP FS.     1.
          SUBT  ZR. 1.    23.
          TRANP GS.    GT.
          MULT  GS.QN.    27
          MULT  27.GT     29
          JUXTC 23 22     24
          JUXTC 29 FS     25
          JUXTR 24 25     26
          MULT  .5  T     T1
          ETPHI 26 T1.28.T2
          MULT  T2.D2.    D4.
          ADD   D4.D3     D.
          RICAT V0.28.21. D.PC.    V0.KN.A1.
          ETPHI 26. T.28.T2
          MULT  T2 D2     D4
          ADD   D4.D3.    D
HEAD 1TRANP KN.    KF.
          MULT  KF.HS.    2
          SUBT  FS. 2.    CF.
          ETPHI CF. T.PH.
          MULT  PH. I.    I.
          RINT  T1.    V0.VAR
          ADD   T1. T.    T1.
          ADD   Z1. T.    Z1
          RINT  Z1.     I. X
          RICAT V0.28.21. D.PC.    V0.KN.A1.
          IF    ND.Z1.HEAD 1
          END
```

.5	1	1				
	.5					
Z1	1	1				
	0					
D2	1	3				
	0		1.	0		
PC	1	4				
	0		0	0	1.	
T	1	1				
	.1					
D3	1	3				
	0		0	.1		
I	3	3				
	1.		0	0	0	1.
	0		0	0	1.	
V0	3	3				
	0		0	0	0	0
	0		0	0	2.	
QN	1	1				
	0					
GS	3	1				
	0		1.	0		
ZR	3	3				
	0		0	0	0	0
	0		0	0	0	
FS	3	3				
	0		1.	0	0	0
	1.		0	0	0	

Fig. 1

HS	1	3		
	1.		0	0
RN	1	1		
	2.			
ND	1	1		
	2.			

Fig. 1

CHAPTER XIV

LOSS OF CONTROLLABILITY BY SAMPLING

1. Description of the Problem: We take a transfer function; after obtaining an irreducible representation and discretizing, we find the "deadbeat" control law. This control law is a function of the sampling interval and we will examine the behavior of the control and the closed loop transients, particularly in the vicinity of points where controllability is lost.

2. Theory and References: In addition to Chapter VIII, see: [1] R. E. Kalman, "On the General Theory of Control Systems", Proc. 1st International Congress in Automatic Control, Moscow (1960); published by Butterworths, London (1961), [2] R. E. Kalman, Y. C. Ho and R. Narendra, "Controllability of Linear Dynamical Systems", Contributions to Differential Equations, vol. 1, no. 2.

We will preface our results with some remarks about deadbeat control in general. Consider a completely controllable system

$$x_{k+1} = \Phi x_k + \Gamma u_k .$$

We assume that $\det \Phi \neq 0$.

Deadbeat control of a sampled system is any control law which takes the state of the system to zero in finitely many (=N) steps. Since

$$x_n - \Phi^n x_o = \Phi^{n-1}\Gamma u_o + \Phi^{n-2}\Gamma u_1 + \dots + \Phi\Gamma_{n-2} + \Gamma u_{n-1} \tag{2.1}$$

it is clear from complete controllability that deadbeat control exists and that we never have to use more than $N = n$ steps. However, the control law is generally not unique. This is because (2.1) admits many control sequences

$$u_o, u_1, \dots, u_{n-1}$$

as a solution. Even if $N = N_o$ = minimum number of steps necessary to take every state to 0, the solution may be nonunique. There is one case when the solution is unique (and $N_o = n$): if $m = 1$, the vectors $\Phi^{-1}\Gamma, \ldots, \Phi^{-N}\Gamma$ are linearly independent and span the state space if and only if $N = n$. In this case we may conclude that <u>any</u> method which takes every state x_o to 0 in n steps must yield the same control law.

Thus (recall Chapter VIII) we let $Q_n = I$, $R_n = 0$, and $Q_k = R_k = 0$ for $1 \leqq k < n$, and compute a time-varying closed-loop control law using the sampled data riccati equation. According to the preceding considerations, we must thereby obtain a control law which is in the case $m = 1$ identical with the unique deadbeat control law (for if not, $x_n' I x_n \neq 0$ for some vector x_n, which would contradict optimality.) In the general case, $m \neq 1$, the situation is very much more complicated.

The riccati equation usually yields a time-varying control law. In the present problem a special situation arises, as follows.

Let the control law given by the riccati equation be

$$u_k = - K_k x_k, \quad k = 0, \ldots, n-1.$$

It is a most interesting fact that K_o also defines a deadbeat control law. In other words (recall uniqueness), we have <u>along every optimal trajectory</u>

$$u_k = - K_o x_k = - K_k x_k, \quad k = 0, \ldots, n-1. \tag{2.2}$$

According to this equation, we can always use a <u>constant</u> control law defined via K_o. This fact is well known (see [1]); but the proof usually requires <u>ad hoc</u> argument. Here we shall give a direct proof based on the riccati equation.

First we note that the optimal transition matrix Φ_i corresponding to the state transition x_{n-i} to x_{n-i+1} is given by

$$\Phi_{i+1} = [I - \Gamma(\Phi_1 \ldots \Phi_i\Gamma)^{\#}\Phi_1 \ldots \Phi_i] \quad i = 0, \ldots, n-1. \tag{2.3}$$

We prove this by induction. Φ_1 is determined by the requirement

$$0 = x_n = \Phi x_{n-1} + \Gamma u_{n-1};$$

hence

$$k_{n+1} = - \Gamma^{\#}\Phi$$

and

$$\Phi_1 = \Phi - \Gamma\Gamma^{\#}\Phi = (1 - \Gamma\Gamma^{\#})\Phi.$$

In the second step,

$$0 = x_n = \Phi_1 x_{n-1} = \Phi_1(\Phi x_{n-2} + \Gamma u_{n-2});$$

hence

$$k_{n-2} = - (\Phi_1\Gamma)^{\#}\Phi_1\Phi$$

and

$$\Phi_2 = \Phi - \Gamma(\Phi_1\Gamma)^{\#}\Phi_1\Phi.$$

The general case may be proved similarly.

So far all calculations are valid for any Γ.

Now we shall assume that $m = 1$ and write Γ as γ. Then

$$\Phi_n^k \Phi^{-k}\gamma = 0 \quad \text{for all } k = 1, \ldots, n. \tag{2.4}$$

(2.4) shows, by complete controllability, that the n-fold application of $\Phi_n = (\Phi - \gamma K_o)$ will take every state into 0 in at most n steps. Since there is only <u>one</u> way of accomplishing this, K_o defines the optimal control law. Thus the <u>time-optimal control law may be computed by the riccati equation when</u> $m = 1$.

To prove (2.4), we note that

$$(A\gamma)^{\#} = \gamma^{1}A'/\|A\gamma\|^{2} \tag{2.5}$$

for any matrix A.

If $k = 1$, (2.4) follows at once from (2.3) and (2.5).

If $k = 2$, we calculate

$$\Phi_n\Phi^{-2}\gamma = [I - \gamma\gamma'\Phi_1' \cdots \Phi_{n-1}'\|\Phi_1 \cdots \Phi_{n-1}\gamma\|^{2}\Phi_1 \cdots \Phi_{n-1}]\Phi^{-1}\gamma.$$

But

$$\Phi_{n-1}\Phi^{-1}\gamma = 0$$

by direct computation. Hence, using (2.5) for $k = 1$,

$$\Phi_n^{2}\Phi^{-2}\gamma = \Phi_n\Phi^{-1}\gamma = 0,$$

the general case is established similarly.

This is the deadbeat transition matrix for the last step. Recursively define

$$\varphi_{i+1} = \varphi - \Gamma\left(\prod_{j=1}^{i}\varphi_j \cdot \Gamma\right)^{\#}\prod_{j=1}^{i}\varphi_j \cdot \varphi$$

e.g.

$$\varphi_3 = \varphi - \Gamma(\varphi_1\varphi_2\Gamma)^{\#}\varphi_1\varphi_2\varphi.$$

These define the deadbeat transition matrices φ_i from the $(n-i)^{ty}$ step to the $(n-i+1)^{th}$ step. We claim that

$$\varphi_i^{j}\varphi^{-k}\Gamma = 0 \quad \text{for} \quad \begin{matrix} j = 1, \ldots, i, \\ k = 1, \ldots, j. \end{matrix}$$

Proof: By induction on i.

$$\varphi_1 \varphi^{-1} \Gamma = \Gamma - \Gamma T^{\#} \Gamma = 0.$$

Crucial to all this is the property of controllability. In fact, controllability for a single-input nonsingular discrete system can be characterized as the existence of a K such that $\Phi - \Gamma K$ is similar to a single Jordan block with zero eigenvalue. Notice that no Φ whose rank is less than $n-1$ can be single input controllable. We see also that controllability of a discrete system does not guarantee that we can transfer from any state to any other unless Φ is nonsingular.

Even if $[\varphi, \Gamma]$ is derived from a completely controllable continuous time system it is possible for complete controllability to be lost by the introduction of sampling. This may occur (see the reference [2]) when the system is discretized with an interval σ such that

$$(2.6) \qquad \begin{cases} \mathrm{Im}[\lambda_i - \lambda_j] = 2\pi \dfrac{q}{\sigma} \\ q = \pm 1, \pm 2, \ldots, \end{cases}$$

for any two eigenvalues λ_i, λ_j such that $\mathrm{Re}[\lambda_i - \lambda_j] = 0$.

This condition is necessary and sufficient for a single input system and is independent of the choice of Γ. That is the system is no longer single-input controllable. Recall that this occurs if the discretized system matrix has more than one Jordan block associated with a given eigenvalue.

3. The Specific Problem. We take the transfer function

$$(3.1) \qquad T(s) = \frac{s + 3/2}{(s+2)(s^2 + \frac{\pi^2}{9})(s^2 + \frac{4\pi^2}{9})}$$

and an irreducible realization

$$H = [1,\ 3/2,\ 0,\ 0,\ 0]$$

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -9.620509 & -4.8103254 & -10.966226 & -5.4831135 & -2. \end{bmatrix}$$

$$G' = [\ 0 \quad 0 \quad 0 \quad 0 \quad 1\ ].$$

We easily see from (2.6) that controllability will be lost when σ is a multiple of 1.5, 2, 3, or 6. Let

$$F_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\frac{\pi}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{2\pi}{3} & 0 \end{pmatrix}$$

correspond to a realization of the factors $(s^2 + \pi^2/9)$ and $(s^2 + 4\pi^2/9)$ in (3.1). Let $e^{Ft} = A(t)$. By examining $[G,\ AG,\ A^2G,\ A^3G]$ for

$$A = \begin{bmatrix} \cos\frac{\pi}{3}t & \sin\frac{\pi}{3}t & 0 & 0 \\ -\sin\frac{\pi}{3}t & \cos\frac{\pi}{3}t & 0 & 0 \\ 0 & 0 & \cos\frac{2\pi}{3}t & \sin\frac{2\pi}{3}t \\ 0 & 0 & -\sin\frac{2\pi}{3}t & \cos\frac{2\pi}{3}t \end{bmatrix},$$

for $t = \sigma$, and appropriate G vectors, we conclude that the maximum controllability rank r is

σ	r
3/2	4
2	3
3	3
4	3
9/2	4
6	2

The two periods, 6 and 3 are commensurable so we might ask how the gains differ for σ and $6 + \sigma$. Since, except for the effect of the root -2, Φ will be the same for σ and $6 + \sigma$ we might hypothesize that the gains will be the same after σ is large enough for $e^{-12\sigma}$ to be negligible.

We would also like to see the closed loop transients for σ and $6 + \sigma$, and the transients and required control for σ in the vicinity of the poles.

4. Results: In Fig. 1 appears the program used to calculate the deadbeat control law

$$[k_1, k_2, k_3, k_4, k_5].$$

k_1, k_2, k_3 and k_5 are graphed in Fig. 2, 3, 4, and 5. Notice how sensitive they are to the sampling period. E.g., k_2 in neighborhoods of $\sigma = 2q$, $q = 1,2, \ldots,$ goes from $-\infty$ to $+\infty$.

In Fig. 6 and 7 appear the graphs of k_1 and k_2 for σ in [6, 9.4]. These show very well how the gains differ between say .3 and 6.4 but by the time we get to 1.4 and 7.4 the differences are less than 10%, by 2.5 and 8.5 the differences are less than 1%. When the differences are large, the small σ gains are larger of course, since the eigenvector associated with -2 must be taken to zero. After σ becomes large $e^{-12\sigma}$ is essentially zero without control.

Next we examine the rank of the matrices

$$[G, \varphi G, \ldots, \varphi^4 G]$$

and

$$[\Gamma, \varphi\Gamma, \ldots, \varphi^4\Gamma].$$

this is done with the program appearing in Fig. 8. Notice that we compute the pseudo-inverse iteratively, thus giving us some idea of the numerical sensitivity of rank. The following table gives our results.

t	r_Γ	ρ_Γ	r_G	ρ_G
1.45	4	$.3 \cdot 10^{-3}$	4	$.6 \cdot 10^{-4}$
	5	$.1 \cdot 10^{-4}$	5	$.2 \cdot 10^{-4}$
1.5	4	$.1 \cdot 10^{-5}$	4	$.2 \cdot 10^{-6}$
	5	$.2 \cdot 10^{-4}$	5	$.7 \cdot 10^{1}$
1.55	4	$.1 \cdot 10^{-3}$	4	$.6 \cdot 10^{-4}$
	5	$.5 \cdot 10^{-5}$	5	$.4 \cdot 10^{-5}$
1.95	3	$.6 \cdot 10^{-3}$	3	$.5 \cdot 10^{-3}$
	4	$.2 \cdot 10^{-4}$	4	$.1 \cdot 10^{-4}$
	5	$.2 \cdot 10^{-3}$	5	$.4 \cdot 10^{-4}$
2.00	3	$.1 \cdot 10^{-5}$	3	$.2 \cdot 10^{-6}$
	4	$.7 \cdot 10^{1}$	4	.1
2.05			3	$.4 \cdot 10^{-3}$
	4	$.4 \cdot 10^{-4}$	4	$.9 \cdot 10^{-5}$
	5	$.1 \cdot 10^{-3}$	5	$.1 \cdot 10^{-3}$

2.95	3	$.3 \cdot 10^{-3}$		
	4	$.9 \cdot 10^{-5}$	4	$.18 \cdot 10^{-4}$
	5	$.2 \cdot 10^{-5}$	5	$.22 \cdot 10^{-4}$
3.0	2	$.3 \cdot 10^{-6}$		
	3	$.2 \cdot 10^{1}$	3	$.1 \cdot 10^{-6}$
			4	.2
3.05	3	$.2 \cdot 10^{-3}$		
	4	$.28 \cdot 10^{-4}$	4	$.26 \cdot 10^{-4}$
	5	$.2 \cdot 10^{-2}$	5	$.5 \cdot 10^{-4}$
3.95			3	$.7 \cdot 10^{-3}$
	4	$.2 \cdot 10^{-4}$	4	$.9 \cdot 10^{-5}$
	5	$.1 \cdot 10^{-3}$	5	$.1 \cdot 10^{-3}$
4.	3	$.1 \cdot 10^{-5}$	3	$.2 \cdot 10^{-6}$
	4	$.2 \cdot 10^{1}$	4	$.1 \cdot 10^{2}$
4.05			3	$.4 \cdot 10^{-3}$
	4	$.3 \cdot 10^{-4}$	4	$.4 \cdot 10^{-5}$
	5	$.8 \cdot 10^{-4}$	5	$.2 \cdot 10^{-3}$
4.45	4	$.2 \cdot 10^{-3}$	4	$.6 \cdot 10^{-4}$
	5	$.1 \cdot 10^{-4}$	5	$.1 \cdot 10^{-4}$
4.5	4	$.2 \cdot 10^{-5}$	4	$.1 \cdot 10^{-6}$
	5	$.2 \cdot 10^{1}$	5	$.2 \cdot 10^{3}$
4.55	4	$.1 \cdot 10^{-3}$	4	$.4 \cdot 10^{-4}$
	5	$.9 \cdot 10^{-5}$	5	$.2 \cdot 10^{-4}$
5.95	2	$.2 \cdot 10^{-3}$		
	3	$.1 \cdot 10^{-4}$	3	$.1 \cdot 10^{-5}$
	4	$.1 \cdot 10^{-1}$	4	$.2 \cdot 10^{-2}$
6.0	1	$.5 \cdot 10^{-7}$		
	2	.7	2	$.7 \cdot 10^{-7}$
			3	$.3 \cdot 10^{2}$

. In general this table speaks for itself, both Γ and G giving the correct answers except at 3 and 6 where the Γ rank is low. We must remember however that G was chosen to maximize the controllability; it is not necessarily true that Γ will give us the largest controllability rank. For instance if

$$F = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad G = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \sigma = 2\pi,$$

then

$$\varphi = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \Gamma = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and

$$[G,\ \varphi G] \quad \text{has rank one,}$$

but

$$[\Gamma,\ \varphi\Gamma] \quad \text{has rank zero.}$$

Certain gain and sampling interval interchanges were made. For instance the gain $K_{2.7}$ which is deadbeat for $\sigma = 2.7$ was run with $\sigma = 2.8$; the results were not as unstable as we might have expected considering the differences between the K values.

The following results were obtained:

σ_K	σ_φ	stability
1.9	2.1	apparently stable
2.1	1.9	apparently stable
2.7	2.8	questionable
2.8	2.7	apparently unstable
2.9	3.1	apparently unstable
3.1	2.9	apparently unstable
3.3	3.4	questionable
3.4	3.3	questionable.

We were also interested in illustrating the lack of controllability by showing a vector which was ignored by the control. We noticed that K_6 was orthogonal to [1, -2, 4, -8, 16] to about six significant figures, so using this vector for initial condtion an uncontrolled run and a run using K_6 were both made and were indeed the same to about four decimal places.

Using the program appearing in Fig. 9 we obtained transients for various sampling periods. In Fig. 10 and 11 appear the responses for $\sigma = 2.5$ and $\sigma = 8.5$. As we have seen, the gain matrices are nearly the same, but we may wonder what is the effect of holding the input over three times as long a period. The graph shows that the value of the output and the wave shape are the same at corresponding control points. The printed output shows that the entire state is the same.

The following table shows the number actually obtained

Control point	$y_{2.5}$	$y_{8.5}$	$u_{2.5}$	$u_{8.5}$
0	1.	1.	7.0253	7.0428
1	.5773	.5800	5.1429	5.1557
2	.4971	.4975	4.4603	4.4649
3	.6674	.6662	2.5650	2.5778
4	.0260	.0302	-0.0175	0.0000

When we tried to obtain the transient near a pole we encountered one of TRNSI's drawbacks. If X is large enough for significance, the control exceeds the print format. So we used the program in Fig. 12 and computed transients for $\sigma = 2.6$, 2.8, and 2.9. The outputs are graphed in Fig. 13, 14, and 15 and illustrate beautifully how control energy and state excursions build up as an uncontrollable point is approached.

```
        BEGIN
        LOAD    T   F   G T1, I ZR   D PC Z1,RZ ON ND N3
HEAD 2EAT       F, T,PH,IN,
        MULT    IN, G,    GM,
        TRANP PH,    PT
        TRANP GM,    GT
        EQUAT   I   P   I X0 Z1   1 Z1, 6
HEAD 1SAMPL PT   P GT Z1 ZR   D PC       P KT AL
        TRANP KT       K
        RINT    K, K
        LOAD    T
        IF      T, T,HEAD 2
        END
 T              1       1
,1
 F              5       5
0                       1.
0                       0                       1.
0                       0                       0                       1.
0                       0                       0                       0                       1.
-9.620651               -4.8103255              -10.966227              -5.4831136              -2.
 G              5       1                                                                       1.
T1              1       1
.16
 I              5       5
1.
                        1.
                                                1.
                                                                        1.
                                                                                                1.
ZR              5       5
0
0
0
0
0
 D              1       2
0                       1.
PC              1       4
0                                                                       5.
Z1              1       1
0
RZ              1       5

ON              1       1
1.
ND              1       1
4.995
N3              1       1
2.4
 T              1       1
6.1
```

Fig. 1

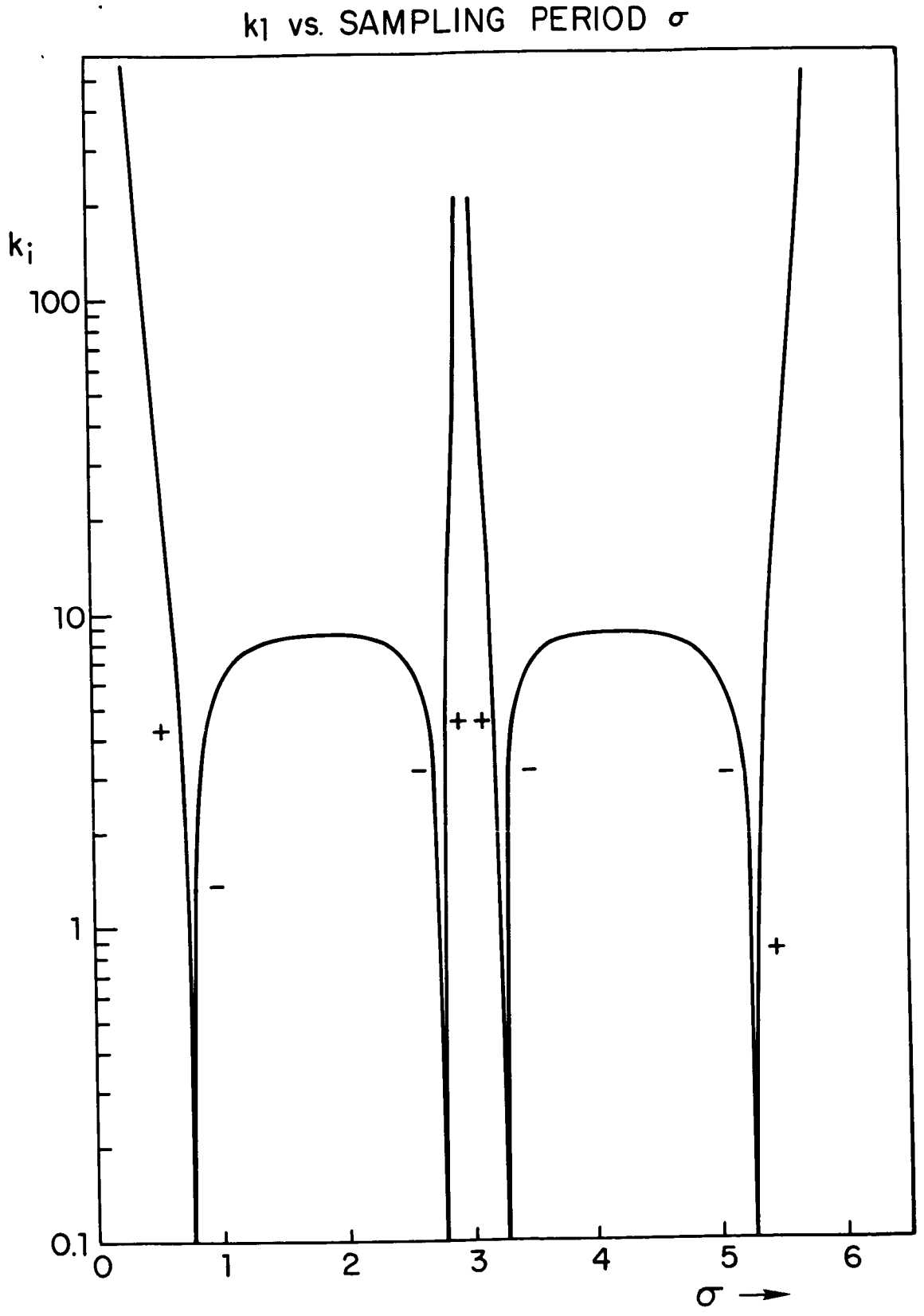

Fig. 2

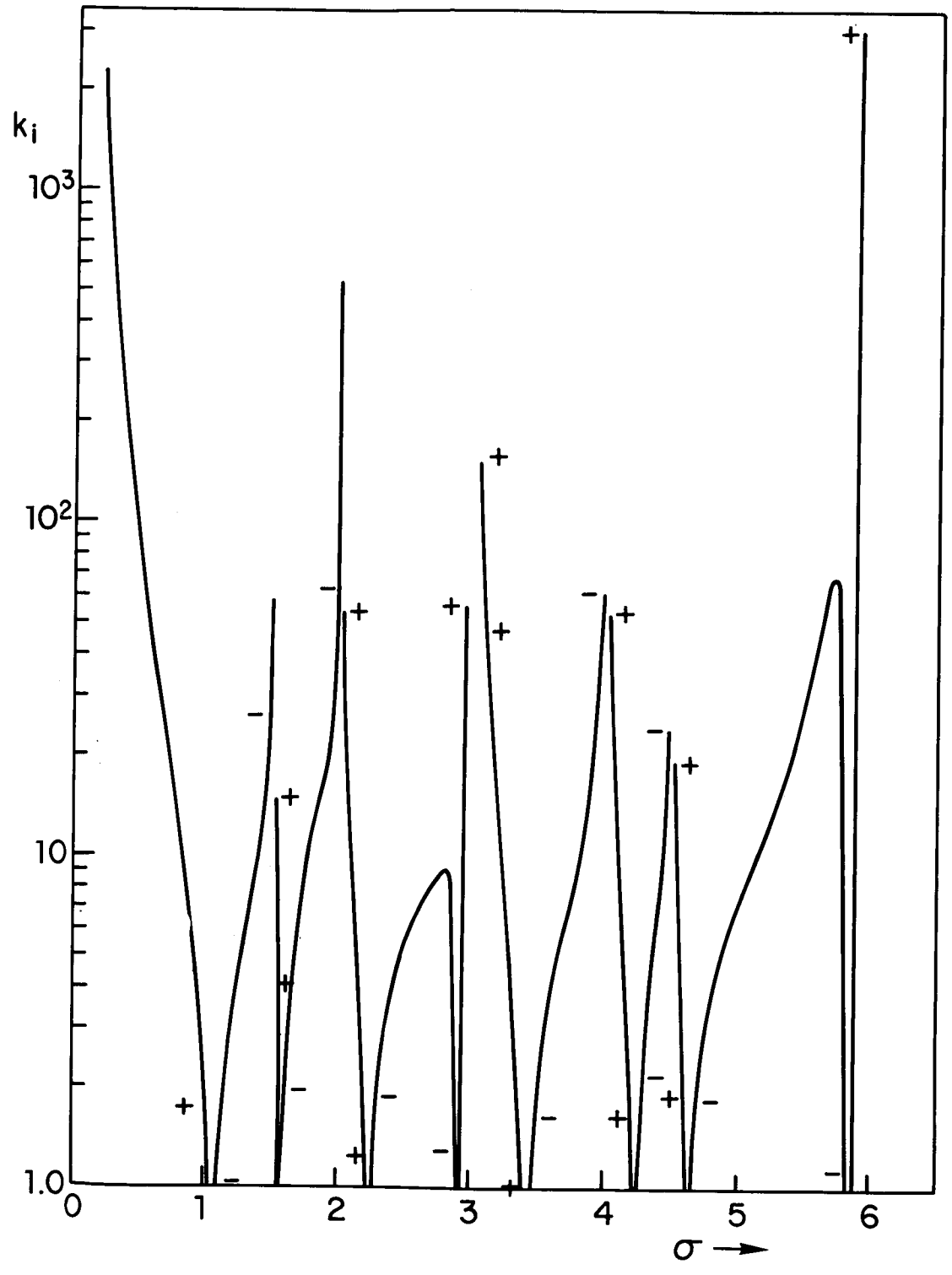
k_2 vs. SAMPLING PERIOD σ
k_i
10^3
10^2
10
1.0
0
1
2
3
4
5
6
$\sigma \rightarrow$

Fig. 3

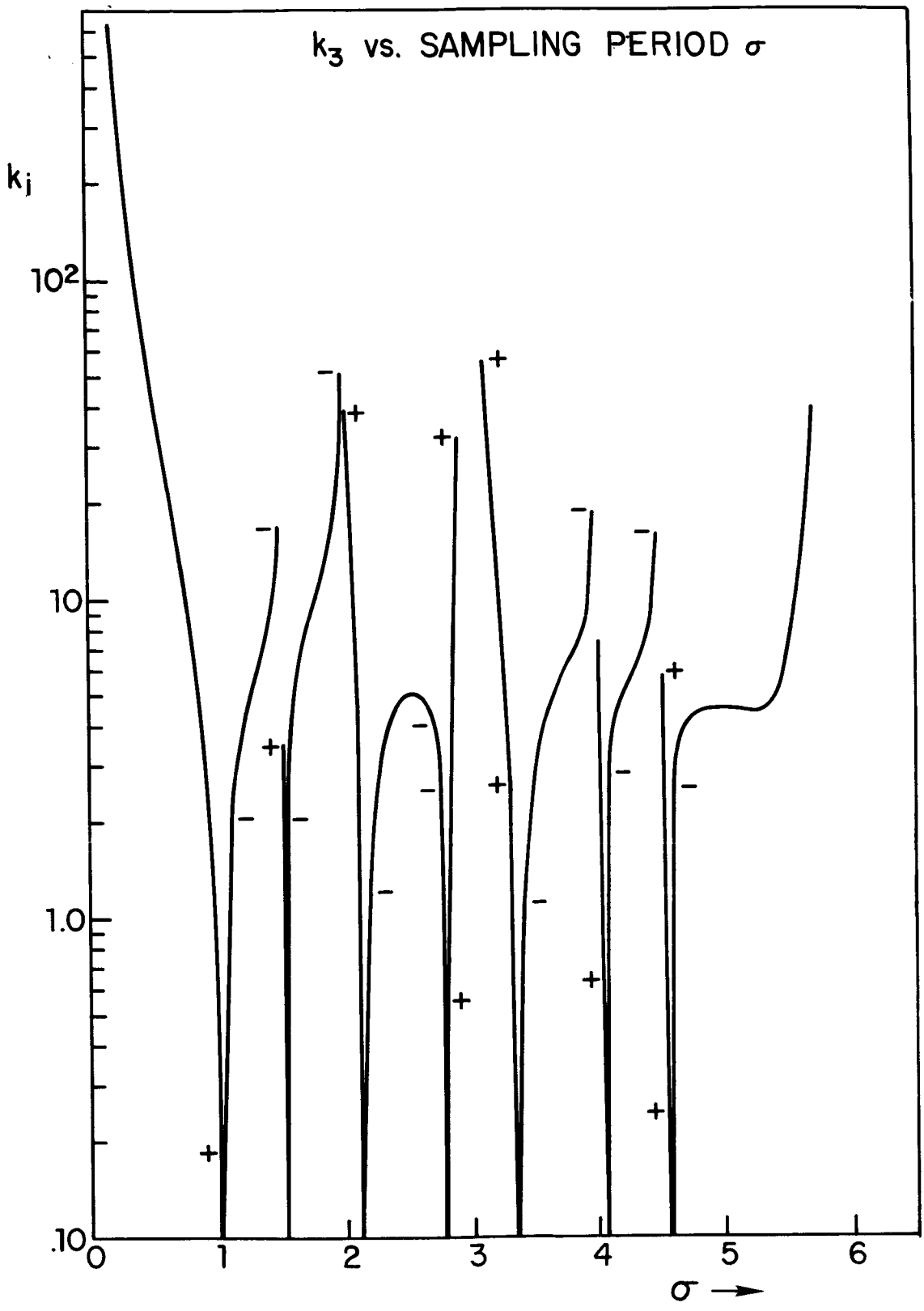

Fig. 4

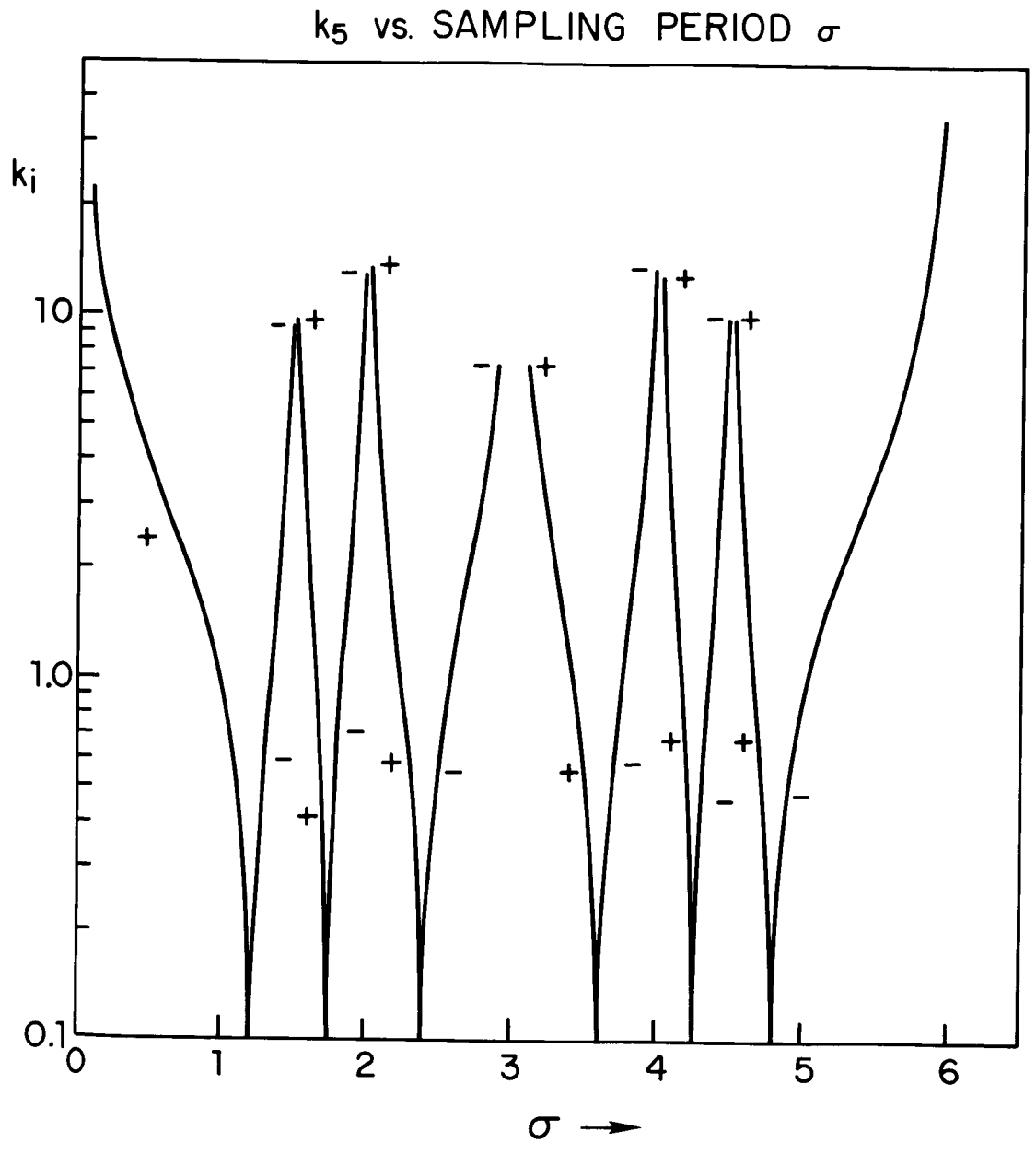

Fig. 5

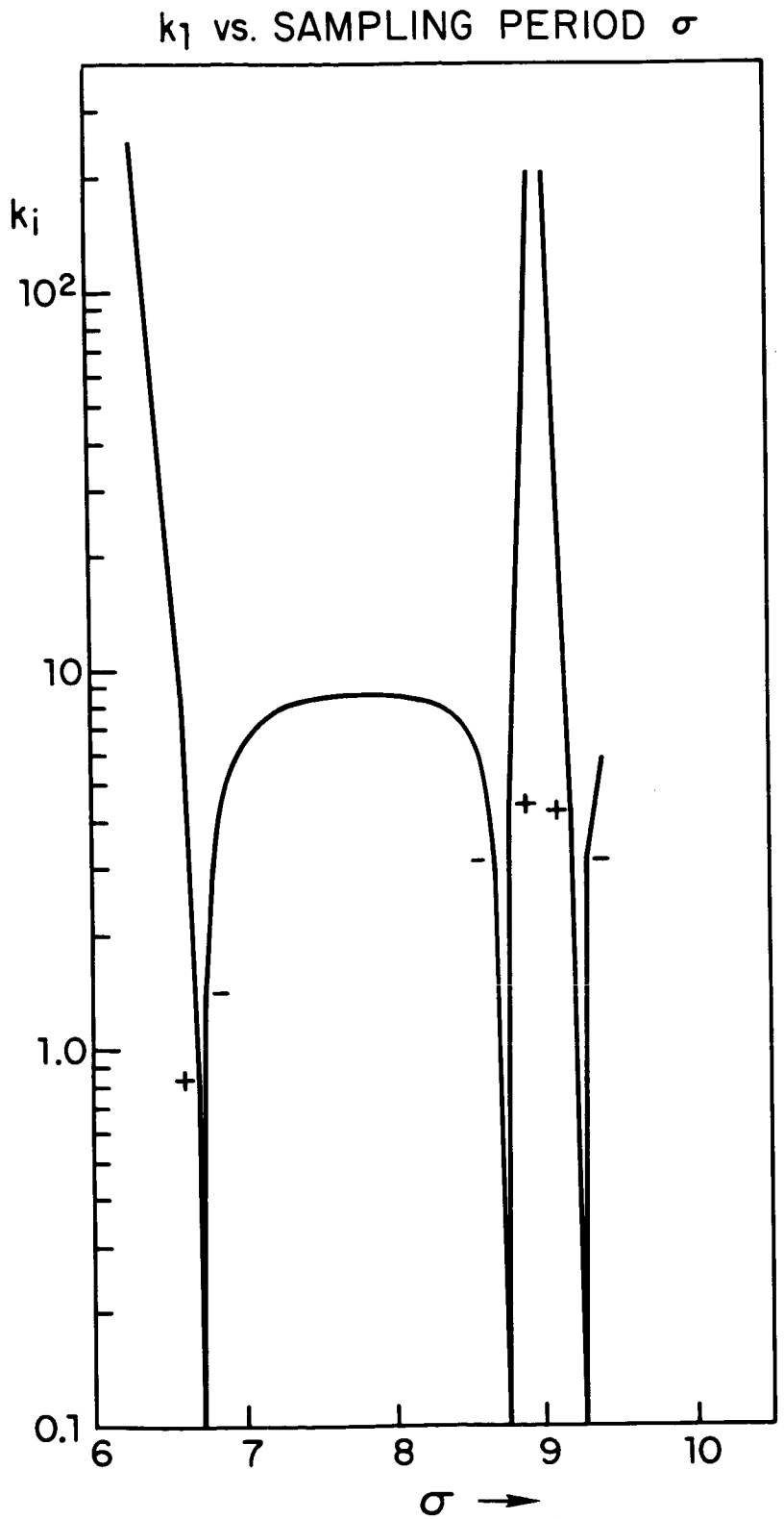

Fig. 6

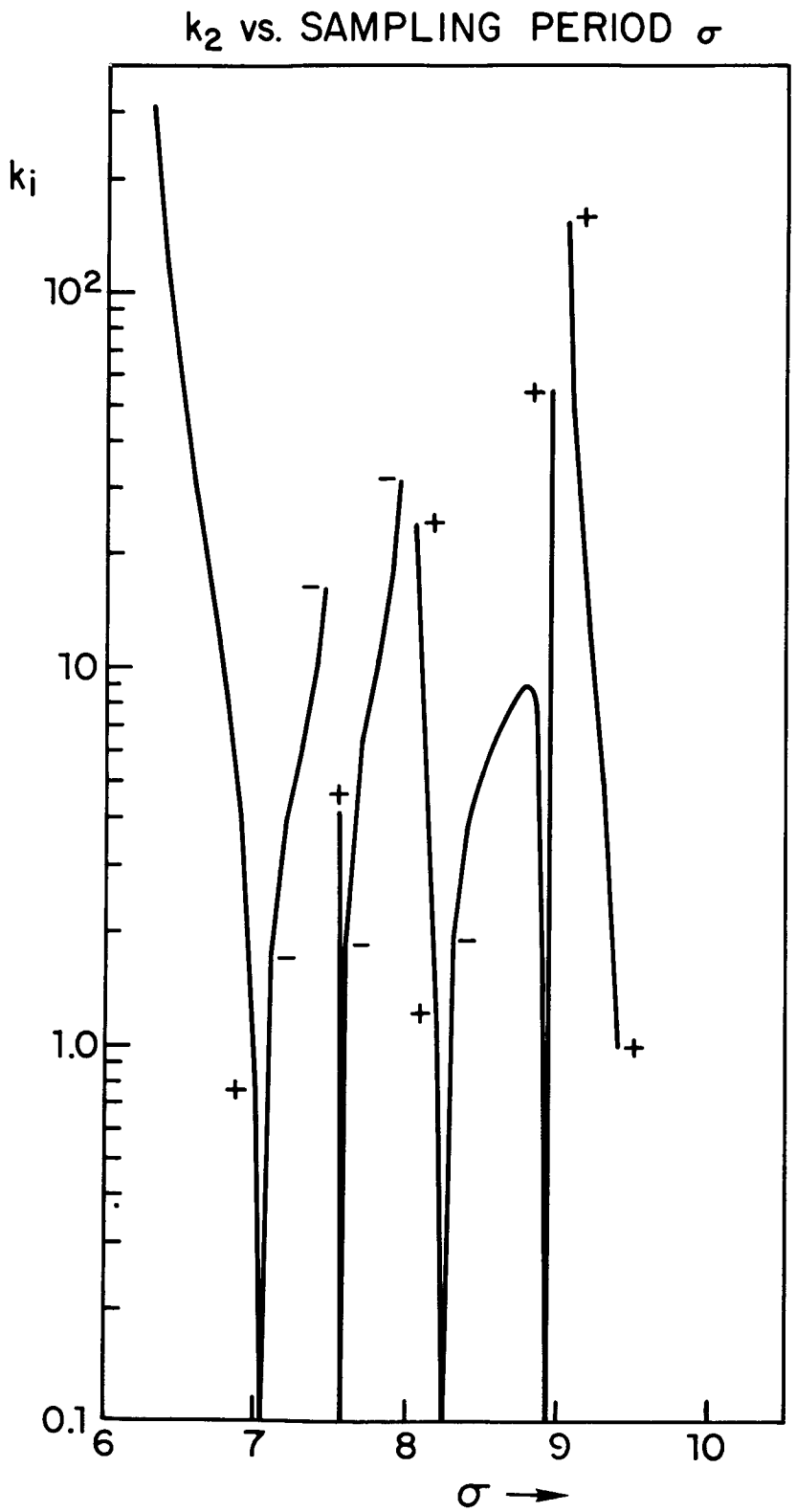

Fig. 7

```
        BEGIN
        LOAD    F   T   G P1,P2
        PIZER P1,P2
HEAD 1EAT     F, T,PH  IN
        MULT   IN   G     GM
        MULT   PH GM      22
        MULT   PH 22      23
        MULT   PH 23      24
        MULT   PH 24      25
        MULT   PH  G      32
        MULT   PH 32      33
        MULT   PH 33      34
        MULT   PH 34      35
        JUXTC  GM 22      41
        JUXTC  41 23      42
        JUXTC  42 24      43
        JUXTC  43,25,     44
        PSEUO  44,   1WI  RK PRINT
        JUXTC   G 32      41
        JUXTC  41 33      42
        JUXTC  42 34      43
        JUXTC  43,35      44
        PSEUO  44,   1WI  RK PRINT
        LOAD    T
        IF      T, T,HEAD 1
        END
 F              5      5
0                    1.
0                                  1.
0                                                  1.
0                                                                  1.
-9.6206509      -4.8103254      -10.966226      -5.4831135       -2.
 T              1      1
1.4
 G              5      1
0                                                                  1.
P1              1      1
.001
P2              1      1
1.
 T              1      1
1.45
 T              1      1
1.5
 T              1      1
1.55
 T              1      1
1.6
 T              1      1
1.9
 T              1      1
1.95
```

Fig. 8

```
          JULY 30, 1965
          BEGIN
          LOAD   Z1,T1, G I5  F ZR  D PC  X  T T2  H
          EAT     F, T1 PH IN
          MULT   IN  G     GM
HEAD 2EAT     F, T,P1,IN
          MULT   IN, G,    G1
          TRANP P1,    PT
          TRANP G1,    GT
          EQUAT I5  P
          SAMPL PT  P GT Z1 ZR  D PC      P KT AL
          TRANP KT      K
          RINT   K  K
          EQUAT  X X1
          TRNSI Z1  K Z1 X1 PH GM  H T2
          LOAD   T T2
          IF     Z1 Z1 HEAD 2
          END
Z1        1      1
0
T1        1      1
.1
 G        5      1
0                                                          1.
 I        5      5
1.
0              1.
0              0              1.
0              0                     1.
0              0                                          1.
 F        5      5
0              1.
0                             1.
0                                    1.
0                                                         1.
-9.6206509     -4.8103254    -10.966226    -5.4831135    -2.
ZR        5      5
0
0
0
0
0
 D        1      2
0              1.
PC        1      4
0              0                            5.
 X        5      1
1.
 T        1      1
2.5
T2        1      4
2.5            .1                           15.1
 H        2      5
```

Fig. 9

1.
0 1.
T 1 1
8.5
T2 1 4
8.5 .1 51.1

Fig. 9

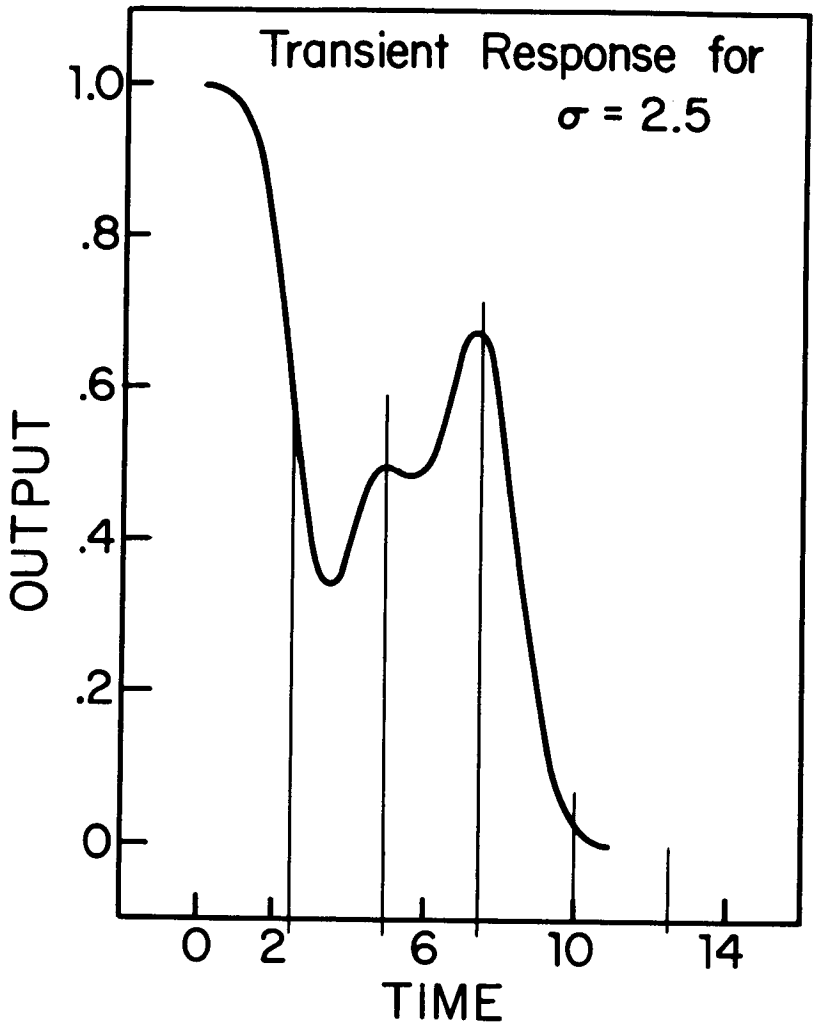

Fig. 10

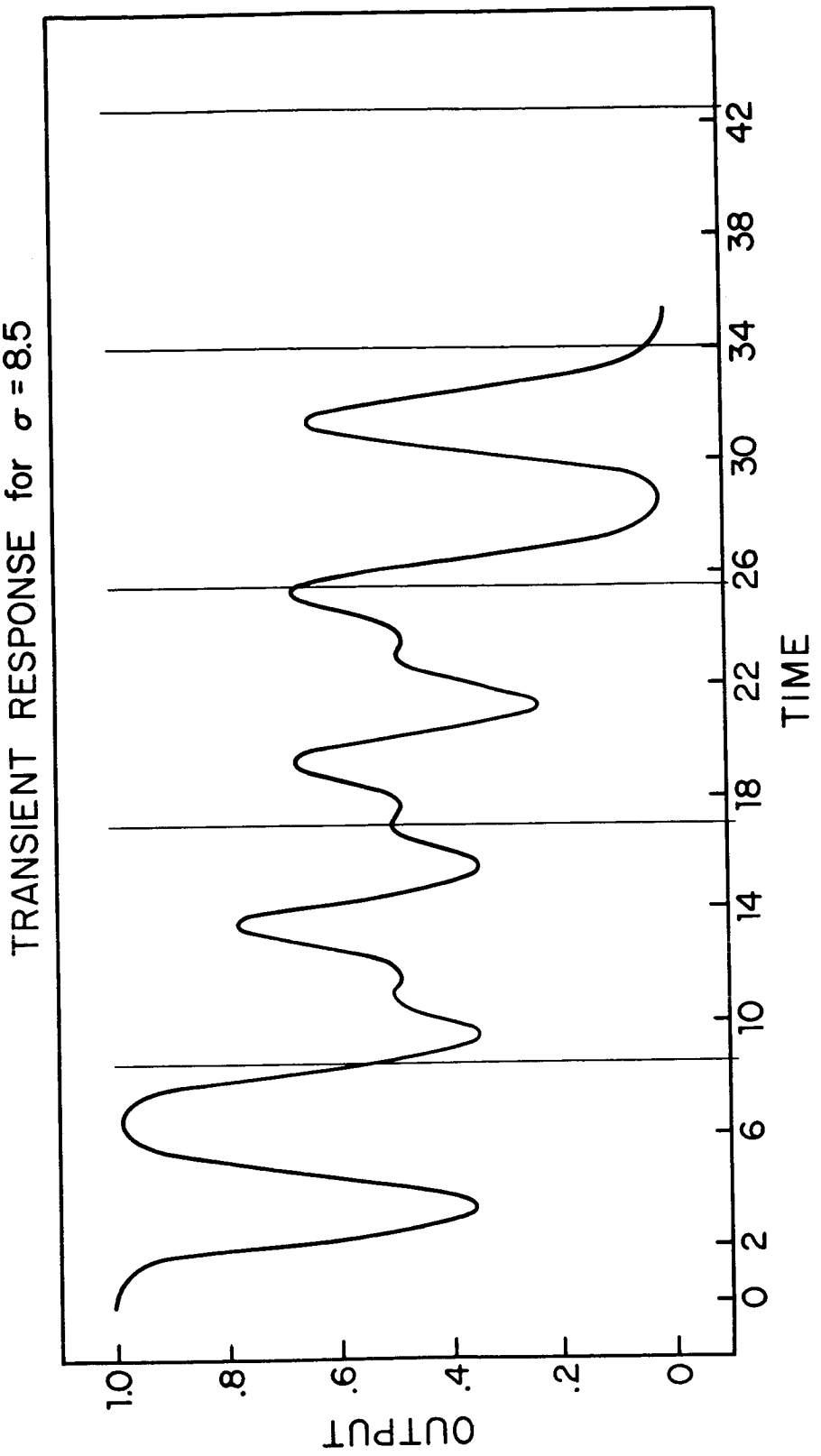
TRANSIENT RESPONSE for σ = 8.5
OUTPUT
1.0
.8
.6
.4
.2
0
0 2 6 10 14 18 22 26 30 34 38 42
TIME

Fig. 11

```
        BEGIN
        LOAD   Z1,T1, G I5   F ZR   D PC RZ   T ND N1
        EAT     F T1 PH IN
        MULT   IN *G     GM
HEAD 3EAT      F, T,P1,IN
        MULT   IN, G,    G1
        TRANP P1,    PT
        TRANP G1,    GT
        EQUAT I5   P
        SAMPL PT   P GT Z1 ZR   D PC       P KT AL
        TRANP KT       K
        RINT   K   K
        EQUAT I5, X Z1 T3
HEAD 2MULT     K   X     -U
        JUXTR  X RZ,     1
        JUXTR  1,-U      2
HEAD 1RINT   T3,     2,XU
        MULT   GM -U     3
        MULT   PH  X     X
        SUBT    X  3     X
        ADD    T1,T3,    T3
        JUXTR  X RZ      1
        JUXTR  1 -U      2
        IF     ND T3 HEAD 1
        ADD     T ND     ND
        IF     N1 ND HEAD 2
        LOAD    T ND N1
        IF     Z1,Z1,HEAD 3
        END
Z1          1       1
0
T1          1       1
.1
 G          5       1
0                                                                    1.
 I          5       5
1.
0               1.
0               0               1.
0               0                       1.
0               0                                                    1.
 F          5       5
0               1.
0                               1.
0                                       1.
0                                                                    1.
-9.6206509      -4.8103254      -10.966226      -5.4831135      -2.
ZR          5       5
0
0
0
0
0
```

Fig. 12

```
 D          1      2
0                 1.
PC         1      4
0                 0                                    5.
RZ         1      5
0                 0          0          0          0
 T         1      1
2.6
ND         1      1
2.55
NI         1      1
15.7
 T         1      1
2.7
ND         1      1
2.65
NI         1      1
16.3
```

Fig. 12

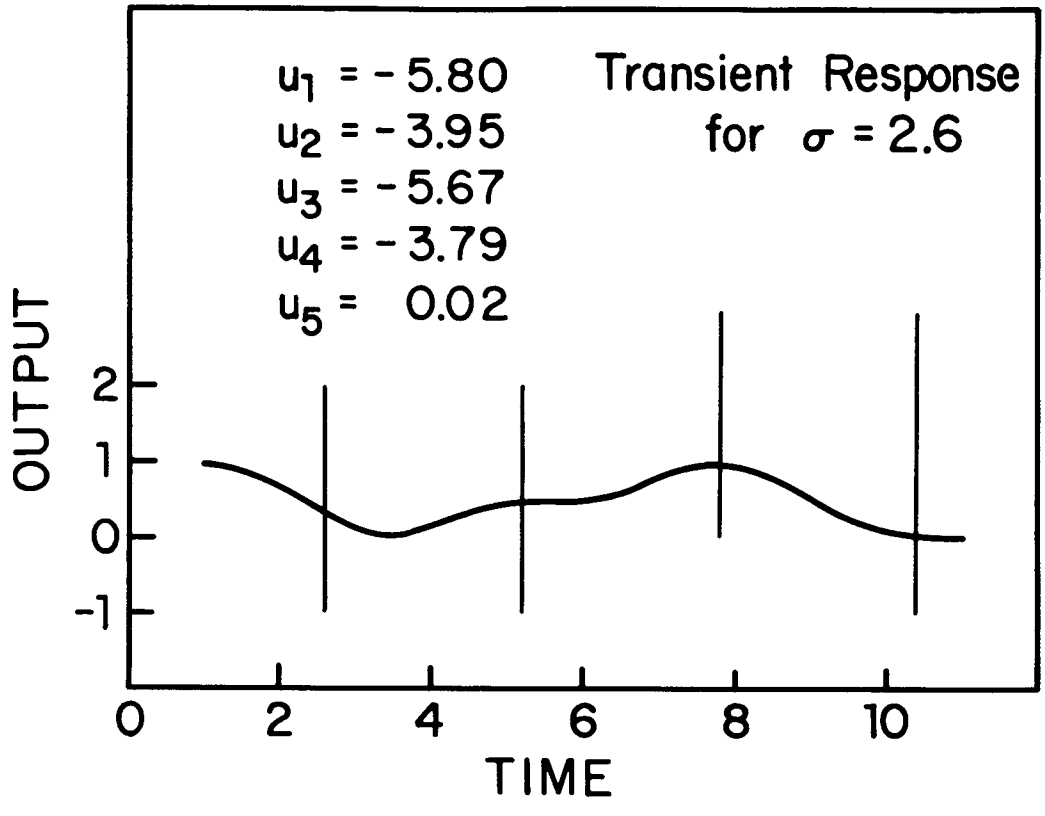
$u_1 = -5.80$
$u_2 = -3.95$
$u_3 = -5.67$
$u_4 = -3.79$
$u_5 = 0.02$
Transient Response
for $\sigma = 2.6$
OUTPUT
2
1
0
-1
0
2
4
6
8
10
TIME

Fig. 13

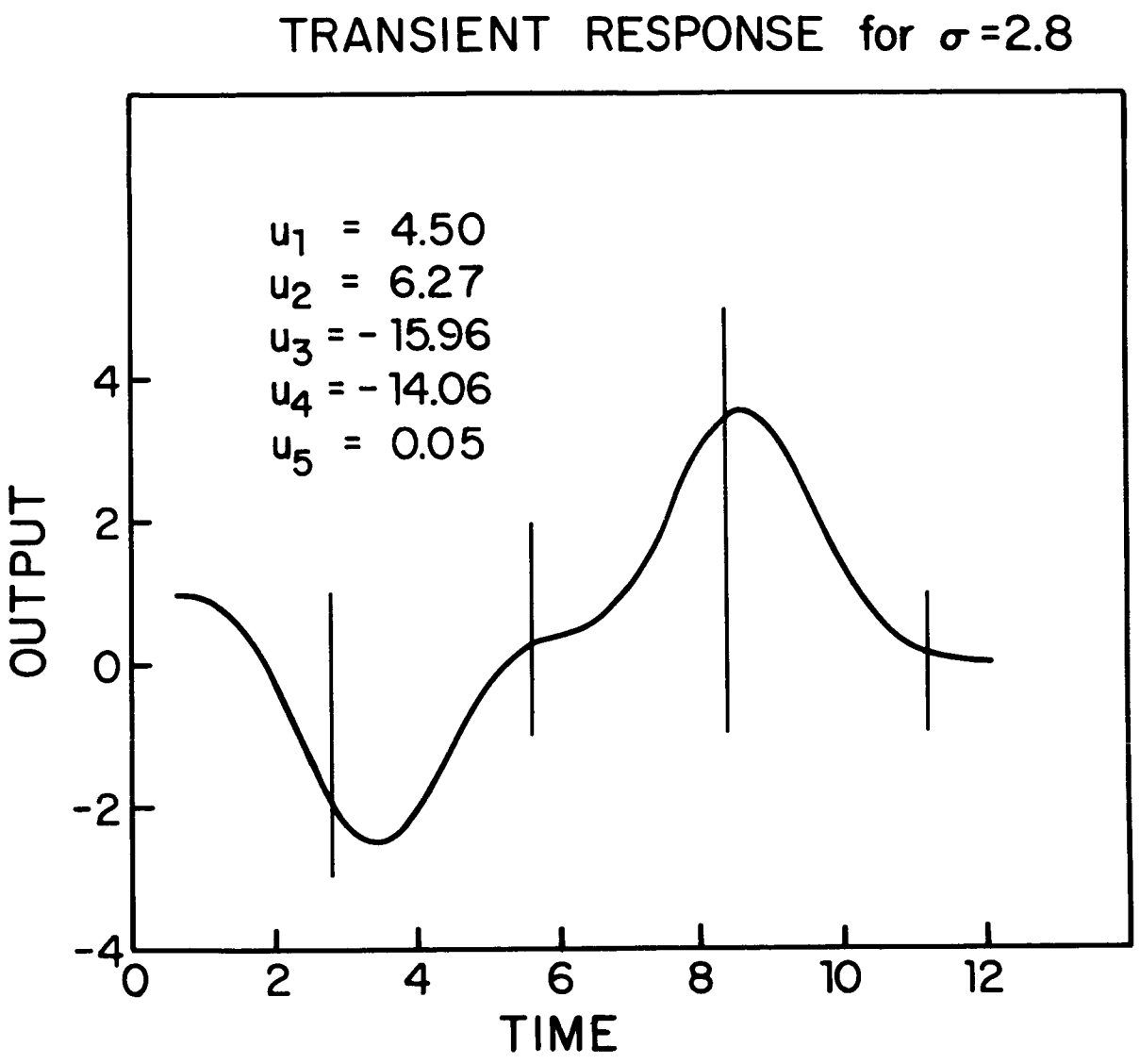
TRANSIENT RESPONSE for σ=2.8
u1 = 4.50
u2 = 6.27
u3 = - 15.96
u4 = - 14.06
u5 = 0.05
OUTPUT
4
2
0
-2
-4
0
2
4
6
8
10
12
TIME

Fig. 14

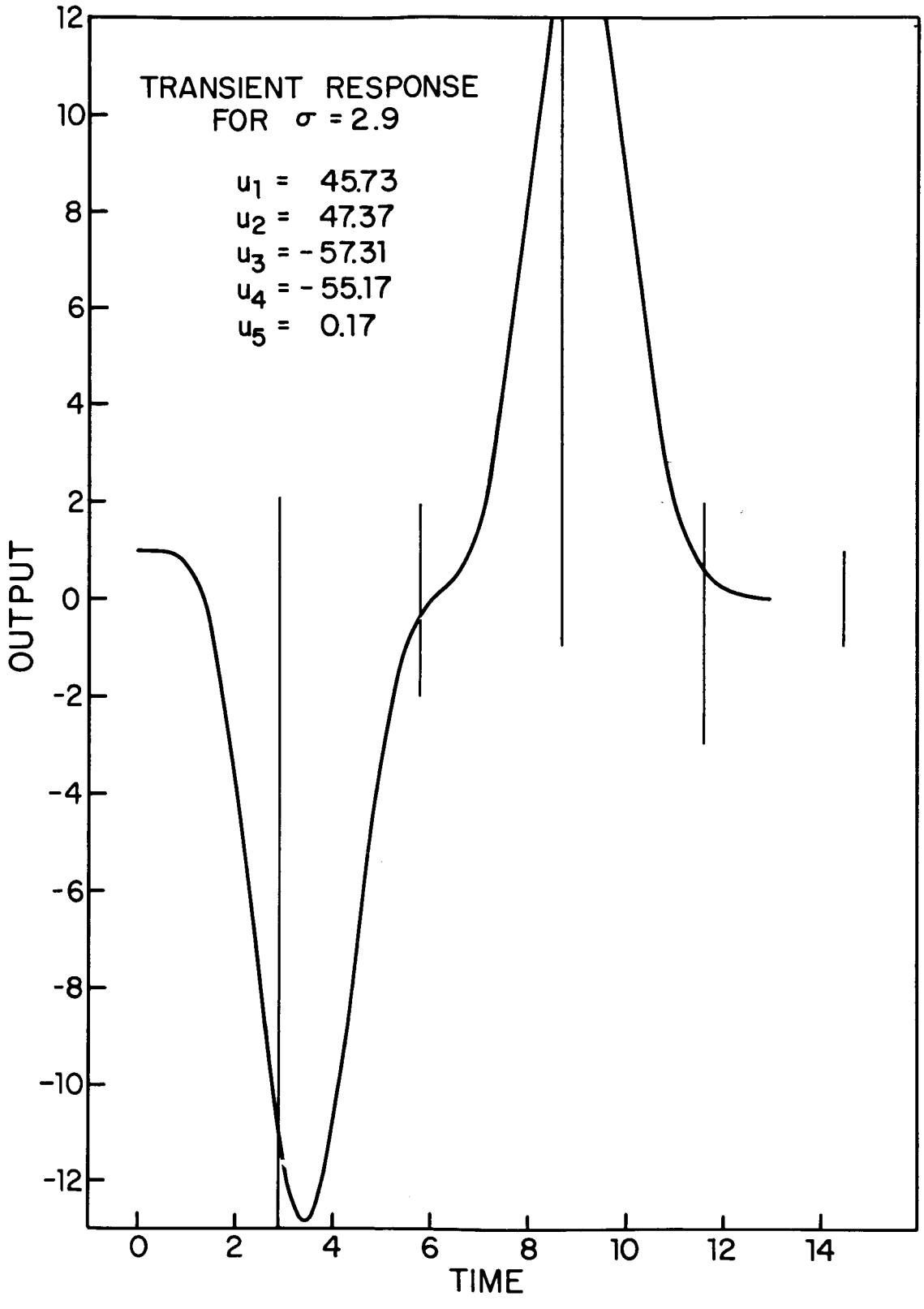
TRANSIENT RESPONSE
FOR σ = 2.9
u1 = 45.73
u2 = 47.37
u3 = -57.31
u4 = -55.17
u5 = 0.17
OUTPUT
TIME
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
0
2
4
6
8
10
12
14

Fig. 15

CHAPTER XV

COMPUTATION OF A MINIMAL REALIZATION

1. Description of the Problem: Given a $p \times m$ matrix $Z(s)$ of transfer functions we wish to obtain a triple $[H, F, G]$ of matrices such that

$$Z(s) = H(sI - F)^{-1}G.$$

Note that Z is then the formal Laplace transform of the impulse response matrix $He^{Ft}G$.

The method used in this chapter was developed by B. L. Ho and has the advantage of directly using experimental input-output data, when Z is interpreted as the z-transform of the sampled impulse-response function $H\Phi^{k}G$, k = integer.

Furthermore, in the two examples which we have used, the method appears to be well suited to computation, although as usual, a large spread in the eigenvalues will occasion numerical scaling problems. Because of these two favorable aspects of the method, it appears to be much better adapted for practical application than the methods appearing in the references 2 and 3.

2. Theory and References:

[1] B. L. Ho, SIAM J. Control, 1966.

[2] R. E. Kalman, "Mathematical description of linear dynamical systems", SIAM J. Control, 1 (1963) 152-192.

[3] R. E. Kalman, "Irreducible realizations and the degree of a rational matrix", SIAM J. 1965.

[4] J.R. Ragazzini and G.F. Franklin, "Sampled-data control systems", (1958) McGraw Hill.

Briefly we can describe our particular variant of Ho's method as follows:

Let $z_{ij}(s)$ denote the elements of $Z(s)$. Let s_{ijk} denote the coefficient of s^{-k} in the expansion of z_{ij} in powers of s^{-1}. Let δ_i denote the degree of the least common denominator of the i^{th} row of $Z(s)$, μ_j the degree of the l.c.d. of the j^{th} column, and $a = \max_i (\max(\delta_i, \mu_i))$. Then we form the matrices

$$S_{ij} = \begin{bmatrix} s_{ij1} & s_{ij2} & \cdots & s_{ija} \\ s_{ij2} & s_{ij3} & \cdots & s_{ija+1} \\ \cdots & \cdots & \cdots & \cdots \\ s_{ija} & s_{ija+1} & & s_{ij2a-1} \end{bmatrix}$$

$$\Sigma_{ji} = \begin{bmatrix} s_{ij2} & s_{ij3} & \cdots & s_{ija+1} \\ s_{ij3} & s_{ij4} & \cdots & s_{ija+1} \\ \cdots & \cdots & \cdots & \cdots \\ s_{ija+1} & s_{ija+2} & \cdots & s_{ij2a} \end{bmatrix}$$

$$S = [S_{ij}], \quad \text{and} \quad \Sigma = [\Sigma_{ij}],$$

$$i = 1, \ldots, p$$
$$j = 1, \ldots, m$$

$$\tilde{G} = [c_1, c_{a+1}, c_{2a+1}, \ldots, c_{(m-1)a+1}]$$

$$\tilde{H} = \begin{bmatrix} r_1 \\ r_{a+1} \\ r_{2a+1} \\ \vdots \\ r_{(p-1)a+1} \end{bmatrix}$$

where c_i is the i^{th} column of S and r_i is the i^{th} row of S.

At this point Ho proceeds by computing nonsingular matrices P and Q such that

$$PSQ = \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix}.$$

Then

$$P\Sigma Q = \begin{bmatrix} F_n & \vdots & x \\ - - & - & - - \\ x & \vdots & x \end{bmatrix}$$

$$\tilde{P}Q = \begin{bmatrix} G \\ --- \\ x \end{bmatrix}$$

$$\tilde{H}Q = [H \vdots x].$$

Then H, F, and G (which are respectively $p \times n$, $n \times n$, and $n \times q$ matrices), constitute a minimal realization of $Z(s)$. The x's represent terms for which no practical or theoretical use is known at present.

We proceed as follows.

Assuming that $m \geq p$, form the $p \times p$ matrix $SS' = B$ and apply D E C O M. This will give us nonsingular matrices T and R such that

$$RTSS'T'R' = \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix}.$$

However, this matrix is $p \times p$, whereas PSQ is $p \times q$. Hence we let $P = RT$ and $Q^* = S'T'R'$. Then we claim that

$$PSQ^* = \begin{bmatrix} F & x \\ x & x \end{bmatrix} \quad P\underline{G} = \begin{bmatrix} G \\ x \end{bmatrix}$$

and

$$HQ' = [H \quad x].$$

The reason for this is that Q^* may be made equal to Q by appending to Q^* $m - p$ columns independent of the previous columns of Q^*. This makes a nonsingular Q, and therefore Ho's formulas for a realization may be applied.

For computational purposes, we may just as well use Q^* rather than Q because we are uninterested in the last $m-p$ columns of PSQ anyway.

3A. The Specific Problem: We take

$$Z(s) = \frac{1}{s^4} \begin{bmatrix} s^3 - s^2 + 1 & 1 & -s^3 + s^2 - 2 \\ 1.5s + 1 & s + 1 & -1.5s - 2 \\ s^3 - 9s^2 - s + 1 & -s^2 + 1 & s^3 - s - 2 \end{bmatrix}$$

See Reference [3]. Then

$$S = \begin{bmatrix}
1 & -1 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & 1 & 0 & -2 \\
-1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -2 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\
0 & 0 & 1.5 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & -1.5 & -2 \\
0 & 1.5 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & -1.5 & -2 & 0 \\
1.5 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & -1.5 & -2 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\
1 & -9 & -1 & 1 & 0 & -1 & 0 & 1 & 1 & 0 & -1 & -2 \\
-9 & -1 & 1 & 0 & -1 & 0 & 1 & 0 & 0 & -1 & -2 & 0 \\
-1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & -2 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0
\end{bmatrix}$$

$$\Sigma = \begin{bmatrix}
-1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -2 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.5 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1.5 & -2 & 0 \\
1.5 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & -1.5 & -2 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-9 & -1 & 1 & 0 & -1 & 0 & 1 & 0 & 0 & -1 & -2 & 0 \\
-1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & -2 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$\tilde{G}' = \begin{bmatrix} 1 & -1 & 0 & 1 & 0 & 0 & 1.5 & 1 & 1 & -9 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & -1 & 0 & 1 \\ -1 & 1 & 0 & -2 & 0 & 0 & -1.5 & -2 & 1 & 0 & -1 & -2 \end{bmatrix}$$

$$\tilde{H} = \begin{bmatrix} 1 & -1 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & 1 & 0 & -2 \\ 0 & 0 & 1.5 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1.5 & -2 \\ 1 & -9 & -1 & 1 & 0 & -1 & 0 & 1 & 1 & 0 & -1 & -2 \end{bmatrix}$$

4A. Results: Applying the program appearing in Fig. 1, we obtained an F, G, and H and checked these by computing their impulse response. The results were excellent. At $t = 3$, the errors were less than four in the seventh significant digit.

The accuracy of the results was caused by the favorable circumstance of the zero eigenvalues. In the next problem we shall see how numerical problems can arise when nonzero eigenvalues are introduced.

3B. The Specific Problem: We consider

$$Z(s) = \begin{bmatrix} \frac{3(s+3)(s+5)}{(s+1(s+2)(s+4)} & \frac{6(s+1)}{(s+2)(s+4)} & \frac{2(s+7)}{(s+3)(s+4)} & \frac{2s+5}{(s+2)(s+3)} \\ \frac{2}{(s+3)(s+5)} & \frac{1}{s+2} & \frac{2(s-5)}{(s+1(s+2)(s+3)} & \frac{8(s+2)}{(s+1)(s+3)(s+5)} \\ \frac{2(s^2+75+18)}{(s+1)(s+3)(s+5)} & \frac{-2s}{(s+1)(s+3)} & \frac{1}{s+3} & \frac{2(5s^2+275+34)}{(s+1)(s+3)(s+5)} \end{bmatrix}$$

We will delineate S_{ij} with solid brackets and Σ_{ij} with dashed brackets

$$[S_{11}] = \begin{bmatrix} 3 & 3 & -18 & 60 & -192 & 648 \\ 3 & -18 & 60 & -192 & 658 & -2328 \\ -18 & 60 & -192 & 648 & -2328 & 8760 \\ 60 & -192 & 648 & -2328 & 8760 & -33912 \\ -192 & 648 & -2328 & 8760 & -33912 & 133368 \end{bmatrix}$$

$$\Sigma_{11} =$$

$$S_{12} = \begin{bmatrix} 6 & -30. & 132 & -552 & 2256 & -9120 \\ -30 & 132 & -552 & 2256 & -9120. & 36672 \\ 132 & -552 & 2256 & -9120 & 36672 & -147072 \\ -552 & 2256 & -9120 & 36672 & -147072 & 589056 \\ 2256 & -9120 & 36672 & -147072 & 589056 & -2357760 \end{bmatrix}$$

$$\Sigma_{12} =$$

$$S_{13} = \begin{bmatrix} 2 & -7 & 25 & -91 & 337 & -1267 \\ -7 & 25 & -91 & 337 & -1267 & 4825 \\ 25 & -91 & 337 & -1267 & 4825 & -18571 \\ -91 & 337 & -1267 & 4825 & -18571 & 72097 \\ 337 & -1267 & 4825 & -18571 & 72097 & -281827 \end{bmatrix}$$

$$\Sigma_{13} =$$

Clearly these matrices are each determined by a ten-vector. We shall write these ten-vectors for the remaining S_{ij}.

1, 1 ≃ [3, 3, -18, 60, -192, 648, -2328,
8760, -33912, 133368]

1, 2 ≃ [6, -30, 132, -55, 2256, -9120,
36673, -147072, 589056, -2357760]

1, 3 ≃ [2, -7, 25, -91, 337, -1267, 4835, -18571,
72097, -281827]

1,4 ≃ [2, -5, 13, -35, 97, -275, 793, -2315
6917, -20195]

2, 1 ≃ [0, 2, -16, 98, -544, 2882, -14896
75938, -384064, 1933442]

2, 2 ≃ [1, -3, 9, -27, 81, -243, 729, -2187, 6561,
-19683]

2, 3 ≃ [0, 2, -22, 110, 430, 1502, -4942, 15710,
-48910, 150302]

2, 4 ≃ [0, 8, -56, 320, -1712, 8888, -45416,
230000, -1158752, 5820008]

3, 1 ≃ [2, -4, 26, -172, 1010, -5524, 29066,
-149692, 761570, -3847204]

3, 2 ≃ [-2, 8, -26, 80, -242, 728, -2186, 6560,
-19682, 59048]

3, 3 ≃ [1, -3, 9, -27, 81, -243, 729, -2187, 6561,
-19683]

3, 4 ≃ [10, -36, 163, -780, 3834, -18996, 94482,
-470940, 2350314, -11738436]

$$G = \begin{bmatrix} 3 & 6 & 2 & 2 \\ 3 & -30 & -7 & -5 \\ -18 & 132 & 25 & 13 \\ 60 & -552 & -91 & -35 \\ -192 & 2256 & 337 & 97 \\ 0 & 1 & 0 & 0 \\ 2 & -3 & 2 & 8 \\ -16 & 9 & -22 & -56 \\ 98 & -27 & 110 & 320 \\ -544 & 81 & -430 & -1712 \\ 2 & -2 & 1 & 10 \\ -4 & 8 & -3 & -36 \\ 26 & -26 & 9 & 162 \\ -172 & 80 & -27 & -780 \\ 1010 & -242 & 81 & 3834 \end{bmatrix}$$

$$H' = \begin{bmatrix} 3 & 0 & 2 \\ 3 & 2 & -4 \\ -18 & -16 & 26 \\ 60 & 98 & -172 \\ -192 & -544 & 1010 \\ 6 & 1 & -2 \\ -30 & -3 & 8 \\ 132 & 9 & -26 \\ -552 & -27 & 80 \\ 2256 & 81 & -242 \\ 2 & 0 & 1 \\ -7 & 2 & -3 \\ 25 & -22 & 9 \\ -91 & 110 & -27 \\ 337 & -430 & 81 \\ 2 & 0 & 10 \\ -5 & 8 & -36 \\ 13 & -56 & 162 \\ -35 & 320 & -780 \\ 97 & -1712 & 3834 \end{bmatrix}$$

Rather than use the impulse response to check this decomposition, we shall use the following relation:

$$HF^kG = [s_{ijk}].$$

4B. Results: Using the program appearing in Fig. 2 we found S to have rank four and computed HG and HFG to be the matrices in Fig. 3. These atrocious results were not unexpected considering that S has an element spread from two to eleven million and the spread in SS' would be even worse.

To obtain better conditioning some simple transformations were made. The rows of S were multiplied by

[1, 1, .1, 1., .01, 1, 1, .1, .1, .01, 1, 1, .1, .1, .01]

and the columns of S by

[1, 1, .1, .1, .01, 1, 1, .1, .1, .01, 1, 1, .1, .1, .01, 1, 1, .1, .1, .01].

Despite the lack of intensive analysis which went into this choice, the results as obtained by the program in Fig. 4 were tremendously improved. Fig. 5 displays HG and HFG which are accurate to about for significant digits, Fig 6 shows HF^6G and HF^7G in which the largest error is about 2%. These errors will increase as the power of F increases. Presumably more careful preconditioning could further improve the accuracy.

```
        BEGIN
        LOAD    S,SB, G, H,P1,P2,II
        PIZER P1,P2,
        TRANP   S,    ST,
        MULT   ST, S,     P,
        DECOM   P,    TI, T,ER,PE, E,RK,
        MULT   PE, T,     T,
        MULT    T,ST,     P,
        TRANP   T,     Q,
        MULT    P, S,     2
        MULT    2, Q,     M
        MULT    P,SB,     1,
        MULT    1, Q,    FB,
        MULT    P, G,    GB,
        MULT    H  Q     HB
        RINT    E  E  M, E FB,FB GB,GB HB,HB
        MULT   II,GB,    GR,
        MULT   II,FB,     3
        TRANP  II,    II
        MULT   HB,II     HR,
        MULT    3,II     FR,
        LOAD    A,ND Z1
        MULT   HR GR      IR
        RINT   Z1,    IR,IMR
        ETPHI  FR  A PH
HEAD 1ADD     Z1, A     Z1
        MULT   PH GR      GR
        MULT   HR GR      IR
        RINT   Z1,    IR,IMR
        IF     ND,Z1,HEAD 1
        PUNCH FR, F,HR, H,GR, G,
        END
```

S	12	12			
	1.	-1.	0	1.	0
	0	0	1.	-1.	1.
	0	-2.	-1.	0	1.
	0	0	0	1.	0
	1.	0	-2.	0	0
	1.	0	0	0	1.
	0	0	0	-2.	0
	0	1.	0	0	0
	1.	0	0	0	-2.
	0	0	0	0	0
	1.5	1.	0	0	1.
	1.	0	0	-1.5	-2.
	0	1.5	1.	0	0
	1.	1.	0	0	-1.5
	-2.	0	1.5	1.	0
	0	1.	1.	0	0
	-1.5	-2.	0	0	1.
	0	0	0	1.	0
	0	0	-2.	0	0
	0	1.	-9.	-1.	1.

Fig. 1

	0	-1.	0	1.	1.
	0	-1.	-2.	-9.	-1.
	1.	0	-1.	0	1.
	0	0	-1.	-2.	0
	-1.	1.	0	0	0
	1.	0	0	-1.	-2.
	0	0	1.	0	0
	0	1.	0	0	0
	-2.	0	0	0	
SB	12 12				
	-1.	0	1.	0	0
	0	1.	0	1.	0
	-2.	0	0	1.	0
	0	0	1.	0	0
	0	-2.	0	0	1.
	0	0	0	1.	0
	0	0	-2.	0	0
	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	1.5
	1.	0	0	1.	1.
	0	0	-1.5	-2.	0
	1.5	1.	0	0	1.
	1.	0	0	-1.5	-2.
	0	0	1.	0	0
	0	1.	0	0	0
	-2.	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0
	0	-9.	-1.	1.	0
	-1.	0	1.	0	0
	-1.	-2.	0	-1.	1.
	0	0	0	1.	0
	0	-1.	-2.	0	0
	1.	0	0	0	1.
	0	0	0	-2.	0
	0	0	0	0	0
	0	0	0	0	0
	0	0	0		
G	12 3				
	1.	0	-1.	-1.	0
	1.	0	0	0	1.
	1.	-2.	0	0	0
	0	0	0	1.5	1.
	-1.5	1.	1.	-2.	1.
	0	1.	-9.	-1.	0
	-1.	0	-1.	1.	1.
	-2.				
H	3 12				
	1.	-1.	0	1.	0
	0	0	1.	-1.	1.
	0	-2.	0	0	1.5
	1.	0	0	1.	1.

Fig. 1

```
                0                0              -1.5            -2.             1.
              -9.               -1.              1.              0              -1.
                0                1.              1.              0              -1.
              -2.
P1              1                1
            1.E-6
P2              1                1
              1.
II              8               12
1.                      0               0               0               0
0                       0               0               0               0
0                       0               0               1.              0
0                       0               0               0               0
0                       0               0               0               0
0                       1.              0               0               0
0                       0               0               0               0
0                       0               0               0               1.
0                       0               0               0               0
0                       0               0               0               0
0                       0               1.              0               0
0                       0               0               0               0
0                       0               0               0               0
1.                      0               0               0               0
0                       0               0               0               0
0                       0               0               1.              0
0                       0               0               0               0
0                       0               0               0               0
0                       1.              0               0               0
0
 T              1                1
.5
ND              1                1
2.5
Z1              1                1
0
```

Fig. 1

```
JULY 30, 1965
BEGIN
LOAD   S1 S2 S3 S4 X1 X2 V1 V2 V3 V4 P1 P2 HR B1 B2 DI LM
MULT   S1,X1,    G1,
MULT   S2 X1     G2
MULT   S3 X1     G3
MULT   S4 X1     G4
JUXTC  G1 G2      1
JUXTC   1,G3      2
JUXTC   2 G4      G
MULT   S1 X2     Q1
MULT   S2 X2     Q2
MULT   S3 X2     Q3
MULT   S4 X2     Q4
JUXTC  Q1,V1,     3
JUXTC   3 Q2      4
JUXTC   4 V2      5
JUXTC   5 Q3      6
JUXTC   6 V3      7
JUXTC   7 Q4      8
JUXTC   8 V4     SB
JUXTC  S1,S2,     9
JUXTC   9 S3     10
JUXTC  10 S4     SR
MULT   DI SR     SR
MULT   SR LM     SR
TRANP  SR,    ST,
PIZER  P1,P2,
MULT   SR,ST,    BQ,
PSEUO  BQ,+ 1BI,RJ,PRINT
DECOM  BQ    1 S   T ER   P   E RK
RINT   ER,ER SB,SB SR,SR   G,GR HR HR
TRANP   T,    TT,
MULT    P, T,     T,
TRANP   P       P
MULT   ST,TT,    12
MULT   12, P,     Q,
MULT    T DI      T
MULT   LM   Q     Q
MULT    T,SB,    13
MULT   13, Q,    14
MULT    T, G     15
MULT   HR, Q,    16
RINT   14, F 15   G 16   H
MULT   14,B1,    17
MULT   B2 17      F
MULT   16 B1      H
MULT   B2 15     GX
RINT    F   F   H   H GX   G
MULT    H GX     18
RINT   18   S
MULT    F GX     GX
MULT    H GX     18
```

Fig. 2

```
RINT  18  S
MULT   F GX    GX
MULT   H GX    18
RINT  18  S
MULT   F GX    GX
MULT   H GX    18
RINT  18  S
MULT   F GX    GX
MULT   H GX    18
RINT  18  S
MULT   F GX    GX
MULT   H GX    18
RINT  18  S
MULT   F GX    GX
MULT   H GX    18
RINT  18  S
MULT   F GX    GX
MULT   H GX    18
RINT  18  S
MULT   F GX    GX
MULT   H GX    18
RINT  18  S
MULT   F GX    GX
MULT   H GX    18
RINT  18  S
MULT   F GX    GX
MULT   H GX    18
RINT  18  S
END
```

```
S1        15      5
3.                3.              -18.            60.             -192.
3.                -18.            60.             -192.           648.
-18.              60.             -192.           648.            -2328.
60.               -192.           648.            -2328.          8760.
-192.             648.            -2328.          8760.           -33912.
0                 2.              -16.            98.             -544.
2.                -16.            98.             -544.           2882.
-16.              98.             -544.           2882.           -14896.
98.               -544.           2882.           -14896.         75938.
-544.             2882.           -14896.         75938.          -384064.
2.                -4.             26.             -172.           1010.
-4.               26.             -172.           1010.           -5524.
26.               -172.           1010.           -5524.          29066.
-172.             1010.           -5524.          29066.          -149692.
1010.             -5524.          29066.          -149692.        761570.
S2        15      5
6.                -30.            132.            -552.           2256.
-30.              132.            -552.           2256.           -9120.
132.              -552.           2256.           -9120.          36672.
-552.             2256.           -9120.          36672.          -147072.
```

Fig. 2

```
2256.          -9120.          36672.          -147072.          589056.
1.             -3.             9.              -27.              81.
-3.            9.              -27.            81.               -243.
9.             -27.            81.             -243.             729.
-27.           81.             -243.           729.              -2187.
81.            -243.           729.            -2187.            6561.
-2.            8.              -26.            80.               -242.
8.             -26.            80.             -242.             728.
-26.           80.             -242.           728.              -2186.
80.            -242.           728.            -2186.            6560.
-242.          728.            -2186.          6560.             -19682.
S3        15    5
2.             -7.             25.             -91.              337.
-7.            25.             -91.            337.              -1267.
25.            -91.            337.            -1267.            4825.
-91.           337.            -1267.          4825.             -18571.
337.           -1267.          4825.           -18571.           72097.
0              2.              -22.            110.              -430.
2.             -22.            110.            -430.             1502.
-22.           110.            -430.           1502.             -4942.
110.           -430.           1502.           -4942.            15710.
-430.          1502.           -4942.          15710.            -48910.
1.             -3.             9.              -27.              81.
-3.            9.              -27.            81.               -243.
9.             -27.            81.             -243.             729.
-27.           81.             -243.           729.              -2187.
81.            -243.           729.            -2187.            6561.
S4        15    5
2.             -5.             13.             -35.              97.
-5.            13.             -35.            97.               -275.
13.            -35.            97.             -275.             793.
-35.           97.             -275.           793.              -2315.
97.            -275.           793.            -2315.            6817.
0              8.              -56.            320.              -1712.
8.             -56.            320.            -1712.            8888.
-56.           320.            -1712.          8888.             -45416.
320.           -1712.          8888.           -45416.           230000.
-1712.         8888.           -45416.         230000.           -1158752.
10.            -36.            162.            -780.             3834.
-36.           162.            -780.           3834.             -18996.
162.           -780.           3834.           -18996.           94482.
-780.          3834.           -18996.         94482.            -470940.
3834.          -18996.         94482.          -470940.          2350314.
X1        5     1
1.
X2        5     4
0                                                                1.
0                                                                1.
0                                                                1.
0                                                                1.
V1        15    1
648.           -2328.          8760.           -33912.           133368.
2882.          -14896.         75938.          -384064.          1933442.
```

Fig. 2

```
-5524.          29066.          -149692.        761570.         -3847204.
V2      15      1
-9120.          36672.          -147072.        589056.         -2357760.
-243.           729.            -2187.          6561.           -19683.
728.            -2186.          6560.           -19682.         59048.
V3      15      1
-1267.          4825.           -18571.         72097.          -281827.
1502.           -4942.          15710.          -48910.         150302.
-243.           729.            -2187.          6561.           -19683.
V4      15      1
-275.           793.            -2315.          6817.           -20195.
8888.           -45416.         230000.         -1158752.       5820008.
-18996.         94482.          -470940.        2350314.        -11738436.
P1      1       1
.000001
P2      1       1
1.
HR      3       20
3.              3.              -18.            60.             -192.
6.              -30.            132.            -552.           2256.
2.              -7.             25.             -91.            337.
2.              -5.             13.             -35.            97.
0               2.              -16.            98.             -544.
1.              -3.             9.              -27.            81.
0               2.              -22.            110.            -430.
0               8.              -56.            320.            -1712.
2.              -4.             26.             -172.           1010.
-2.             8.              -26.            80.             -242.
1.              -3.             9.              -27.            81.
10.             -36.            162.            -780.           3834.
B1      15      12
1.

                                                1.

        1.                                                      1.

                1.

1.                                              1.

        1.                                                      1.
```

Fig. 2

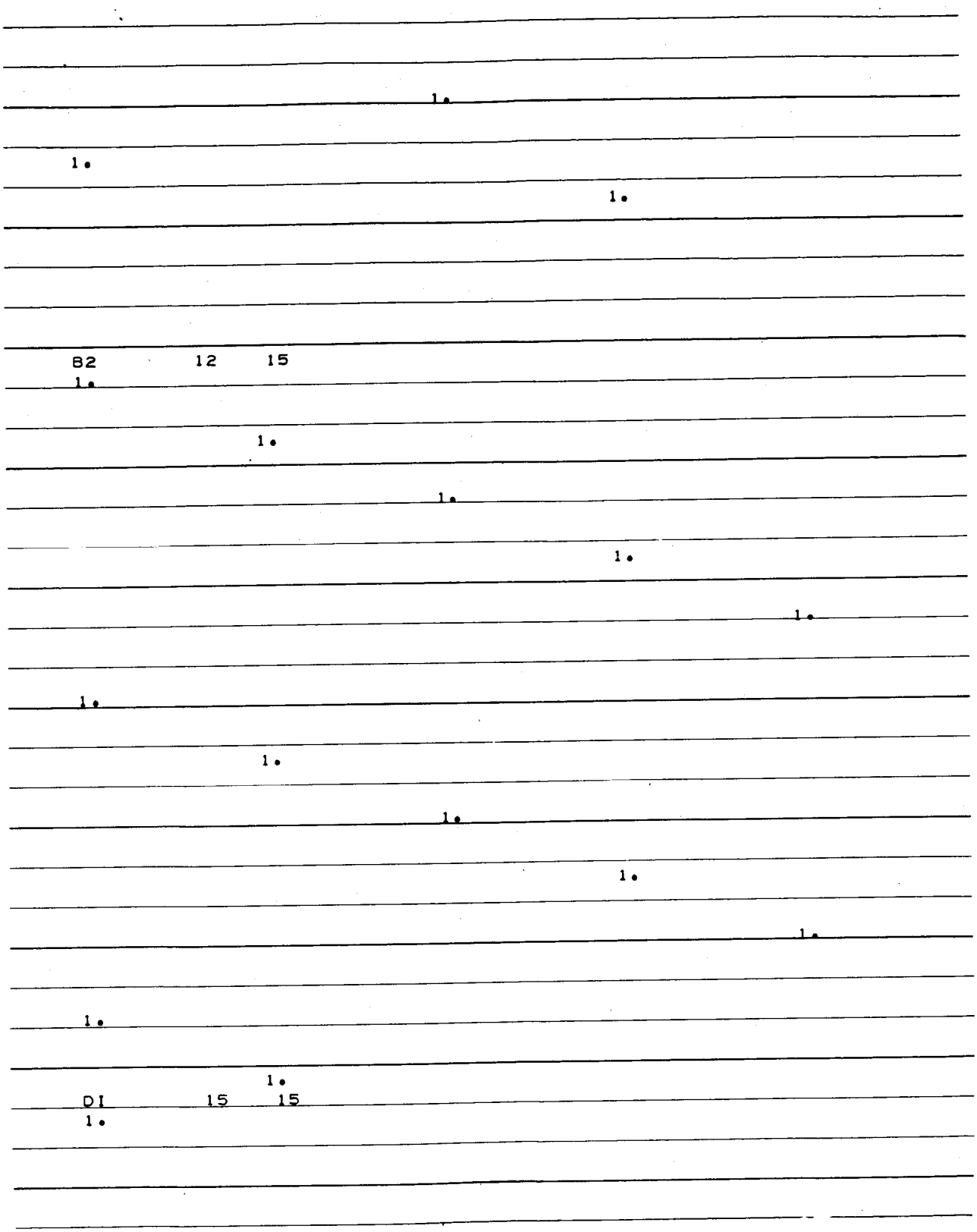

Fig. 2

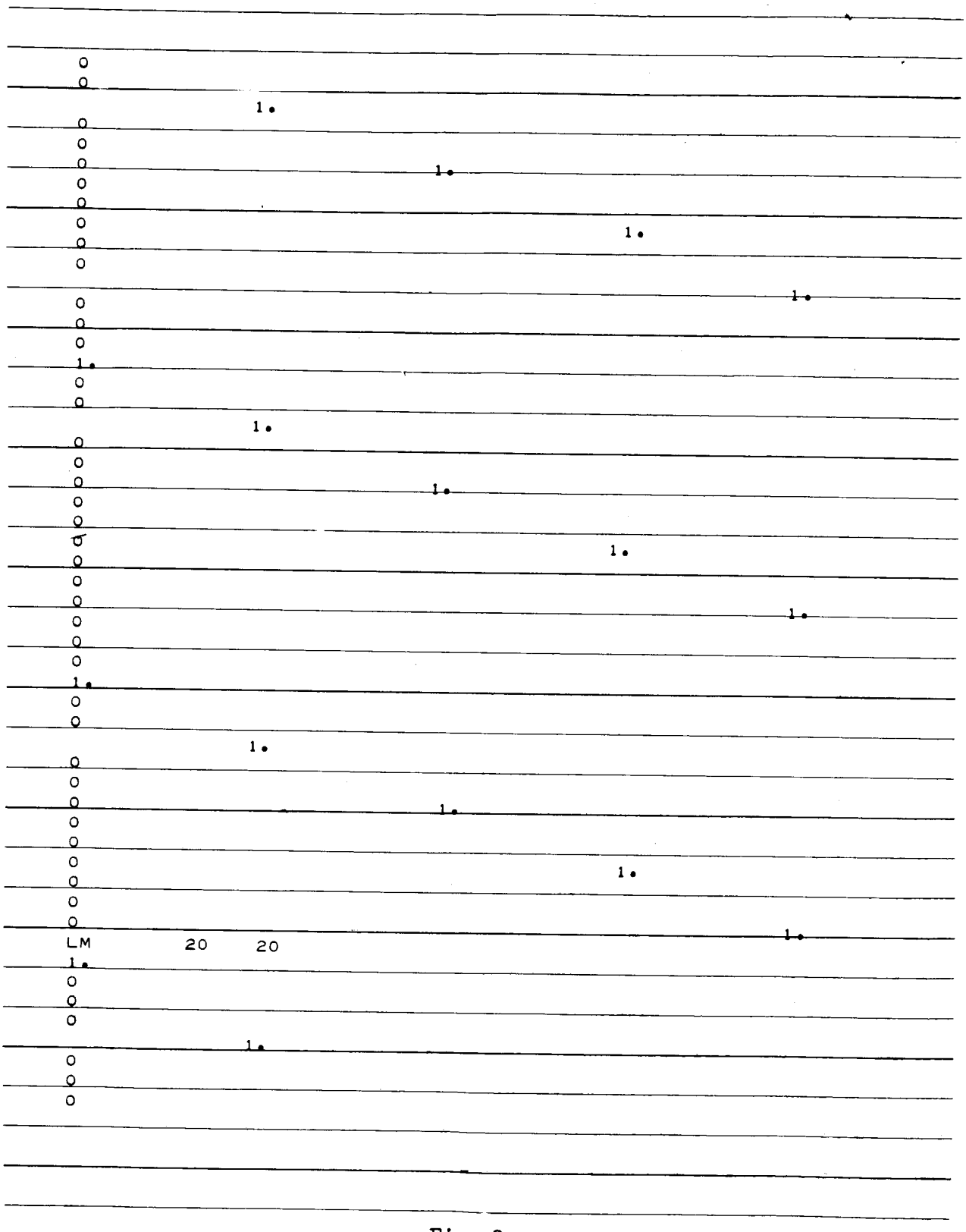

Fig. 2

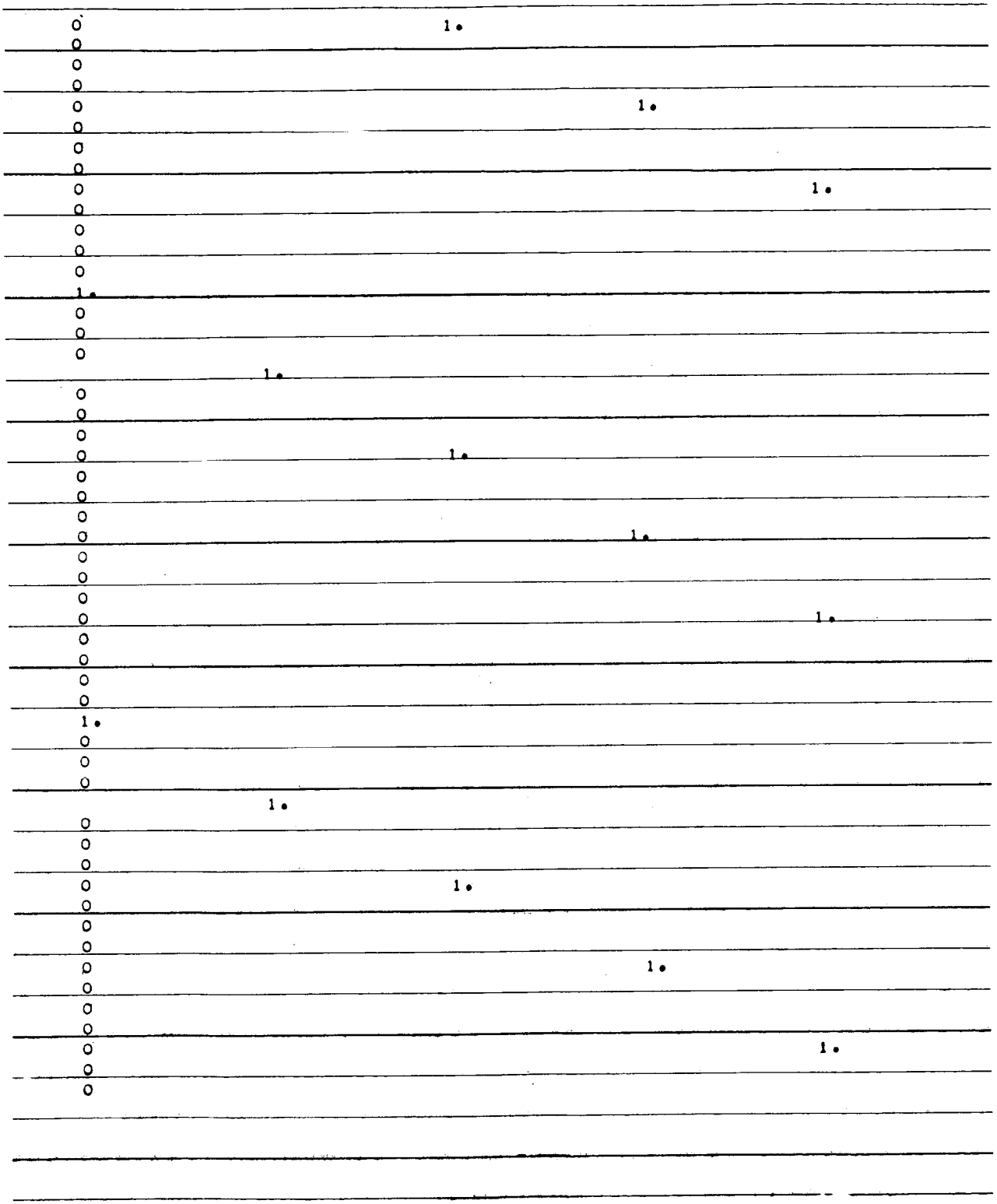

Fig. 2

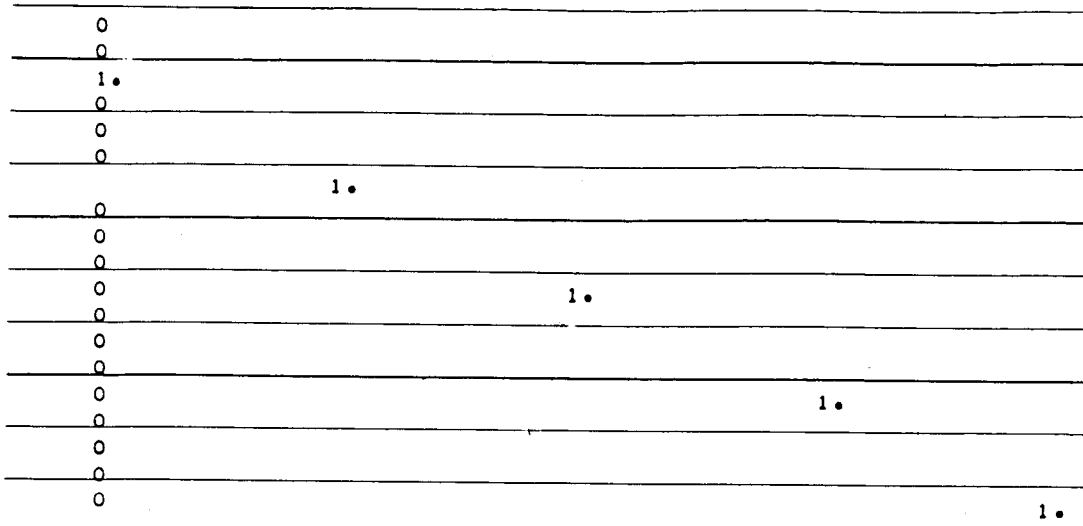

Fig. 2

MATRIX S

NUMBER OF ROWS 3 NUMBER OF COLUMNS 4

-0.60633675E 05	-0.82538218E 05	-0.31000812E 05	0.81135172E 05
0.48906875E 05	0.52736579E 05	0.61705748E 05	-0.40652142E 05
-0.97309955E 05	-0.12957074E 06	-0.53789978E 05	0.12663344E 06

MATRIX S

NUMBER OF ROWS 3 NUMBER OF COLUMNS 4

-0.15341415E 11	-0.25053193E 11	-0.46605802E 09	0.25663784E 11
0.11584865E 11	0.19123507E 11	-0.19607167E 09	-0.19742244E 11
-0.24571605E 11	-0.40141921E 11	-0.69948262E 09	0.41134847E 11

Fig. 3

```
BEGIN
LOAD  S1 S2 S3 S4 X1 X2 V1 V2 V3 V4 P1 P2 HR B1 B2 DI LM
MULT  S1,X1,   G1,
MULT  S2 X1    G2
MULT  S3 X1    G3
MULT  S4 X1    G4
JUXTC G1 G2    1
JUXTC 1,G3     2
JUXTC 2 G4     G
MULT  S1 X2    Q1
MULT  S2 X2    Q2
MULT  S3 X2    Q3
MULT  S4 X2    Q4
JUXTC Q1,V1,   3
JUXTC 3 Q2     4
JUXTC 4 V2     5
JUXTC 5 Q3     6
JUXTC 6 V3     7
JUXTC 7 Q4     8
JUXTC 8 V4     SB
JUXTC S1,S2,   9
JUXTC 9 S3     10
JUXTC 10 S4    SR
MULT  DI SR    SR
MULT  SR LM    SR
TRANP SR,  ST,
PIZER P1,P2,
MULT  SR,ST,   BQ,
PSEUO BQ,+ 1BI,RJ,PRINT
DECOM BQ  1 S  T ER  P  E RK
RINT  ER,ER SB,SB SR,SR  G,GR HR HR
TRANP T,   TT,
MULT   P, T,   T,
TRANP  P     P
MULT  ST,TT,   12
MULT  12, P,   Q,
MULT   T DI    T
MULT  LM  Q    Q
MULT   T,SB,   13
MULT  13, Q,   14
MULT   T, G    15
MULT  HR, Q,   16
RINT  14, F 15  G 16  H
MULT  14,B1,   17
MULT  B2 17    F
MULT  16 B1    H
MULT  B2 15    GX
RINT   F  F  H  H GX  G
MULT   H GX    18
RINT  18  S
MULT   F GX    GX
MULT   H GX    18
RINT  18  S
```

Fig. 4

```
        MULT   F GX     GX
        MULT   H GX     18
        RINT   18  S
        MULT   F GX     GX
        MULT   H GX     18
        RINT   18  S
        MULT   F GX     GX
        MULT   H GX     18
        RINT   18  S
        MULT   F GX     GX
        MULT   H GX     18
        RINT   18  S
        MULT   F GX     GX
        MULT   H GX     18
        RINT   18  S
        MULT   F GX     GX
        MULT   H GX     18
        RINT   18  S
        MULT   F GX     GX
        MULT   H GX     18
        RINT   18  S
        MULT   F GX     GX
        MULT   H GX     18
        RINT   18  S
        MULT   F GX     GX
        MULT   H GX     18
        RINT   18  S
        END
```

```
S1          15      5
3.          3.      -18.    60.     -192.
3.          -18.    60.     -192.   648.
-18.        60.     -192.   648.    -2328.
60.         -192.   648.    -2328.  8760.
-192.       648.    -2328.  8760.   -33912.
0           2.      -16.    98.     -544.
2.          -16.    98.     -544.   2882.
-16.        98.     -544.   2882.   -14896.
98.         -544.   2882.   -14896. 75938.
-544.       2882.   -14896. 75938.  -384064.
2.          -4.     26.     -172.   1010.
-4.         26.     -172.   1010.   -5524.
26.         -172.   1010.   -5524.  29066.
-172.       1010.   -5524.  29066.  -149692.
1010.       -5524.  29066.  -149692. 761570.
S2          15      5
6.          -30.    132.    -552.   2256.
-30.        132.    -552.   2256.   -9120.
132.        -552.   2256.   -9120.  36672.
-552.       2256.   -9120.  36672.  -147072.
2256.       -9120.  36672.  -147072. 589056.
```

Fig. 4

```
1.          -3.          9.          -27.         81.
-3.         9.           -27.        81.          -243.
9.          -27.         81.         -243.        729.
-27.        81.          -243.       729.         -2187.
81.         -243.        729.        -2187.       6561.
-2.         8.           -26.        80.          -242.
8.          -26.         80.         -242.        728.
-26.        80.          -242.       728.         -2186.
80.         -242.        728.        -2186.       6560.
-242.       728.         -2186.      6560.        -19682.
S3          15    5
2.          -7.          25.         -91.         337.
-7.         25.          -91.        337.         -1267.
25.         -91.         337.        -1267.       4825.
-91.        337.         -1267.      4825.        -18571.
337.        -1267.       4825.       -18571.      72097.
0           2.           -22.        110.         -430.
2.          -22.         110.        -430.        1502.
-22.        110.         -430.       1502.        -4942.
110.        -430.        1502.       -4942.       15710.
-430.       1502.        -4942.      15710.       -48910.
1.          -3.          9.          -27.         81.
-3.         9.           -27.        81.          -243.
9.          -27.         81.         -243.        729.
-27.        81.          -243.       729.         -2187.
81.         -243.        729.        -2187.       6561.
S4          15    5
2.          -5.          13.         -35.         97.
-5.         13.          -35.        97.          -275.
13.         -35.         97.         -275.        793.
-35.        97.          -275.       793.         -2315.
97.         -275.        793.        -2315.       6817.
0           8.           -56.        320.         -1712.
8.          -56.         320.        -1712.       8888.
-56.        320.         -1712.      8888.        -45416.
320.        -1712.       8888.       -45416.      230000.
-1712.      8888.        -45416.     230000.      -1158752.
10.         -36.         162.        -780.        3834.
-36.        162.         -780.       3834.        -18996.
162.        -780.        3834.       -18996.      94482.
-780.       3834.        -18996.     94482.       -470940.
3834.       -18996.      94482.      -470940.     2350314.
X1          5     1
1.
X2          5     4
0                                                  1.
0                                                  1.
0                                                  1.
0                                                  1.
V1          15    1
648.        -2328.       8760.       -33912.      133368.
2882.       -14896.      75938.      -384064.     1933442.
-5524.      29066.       -149692.    761570.      -3847204.
```

Fig. 4

```
V2          15      1
-9120.              36672.          -147072.        589056.         -2357760.
-243.               729.            -2187.          6561.           -19683.
728.                -2186.          6560.           -19682.         59048.
V3          15      1
-1267.              4825.           -18571.         72097.          -281827.
1502.               -4942.          15710.          -48910.         150302.
-243.               729.            -2187.          6561.           -19683.
V4          15      1
-275.               793.            -2315.          6817.           -20195.
8888.               -45416.         230000.         -1158752.       5820008.
-18996.             94482.          -470940.        2350314.        -11738436.
P1          1       1
.000001
P2          1       1
1.
HR          3       20
3.                  3.              -18.            60.             -192.
6.                  -30.            132.            -552.           2256.
2.                  -7.             25.             -91.            337.
2.                  -5.             13.             -35.            97.
0                   2.              -16.            98.             -544.
1.                  -3.             9.              -27.            81.
0                   2.              -22.            110.            -430.
0                   8.              -56.            320.            -1712.
2.                  -4.             26.             -172.           1010.
-2.                 8.              -26.            80.             -242.
1.                  -3.             9.              -27.            81.
10.                 -36.            162.            -780.           3834.
B1          15      12
1.
                                                    1.
                    1.
                                                                    1.
                                    1.
1.
                                                    1.
                    1.
                                                                    1.
```

Fig. 4

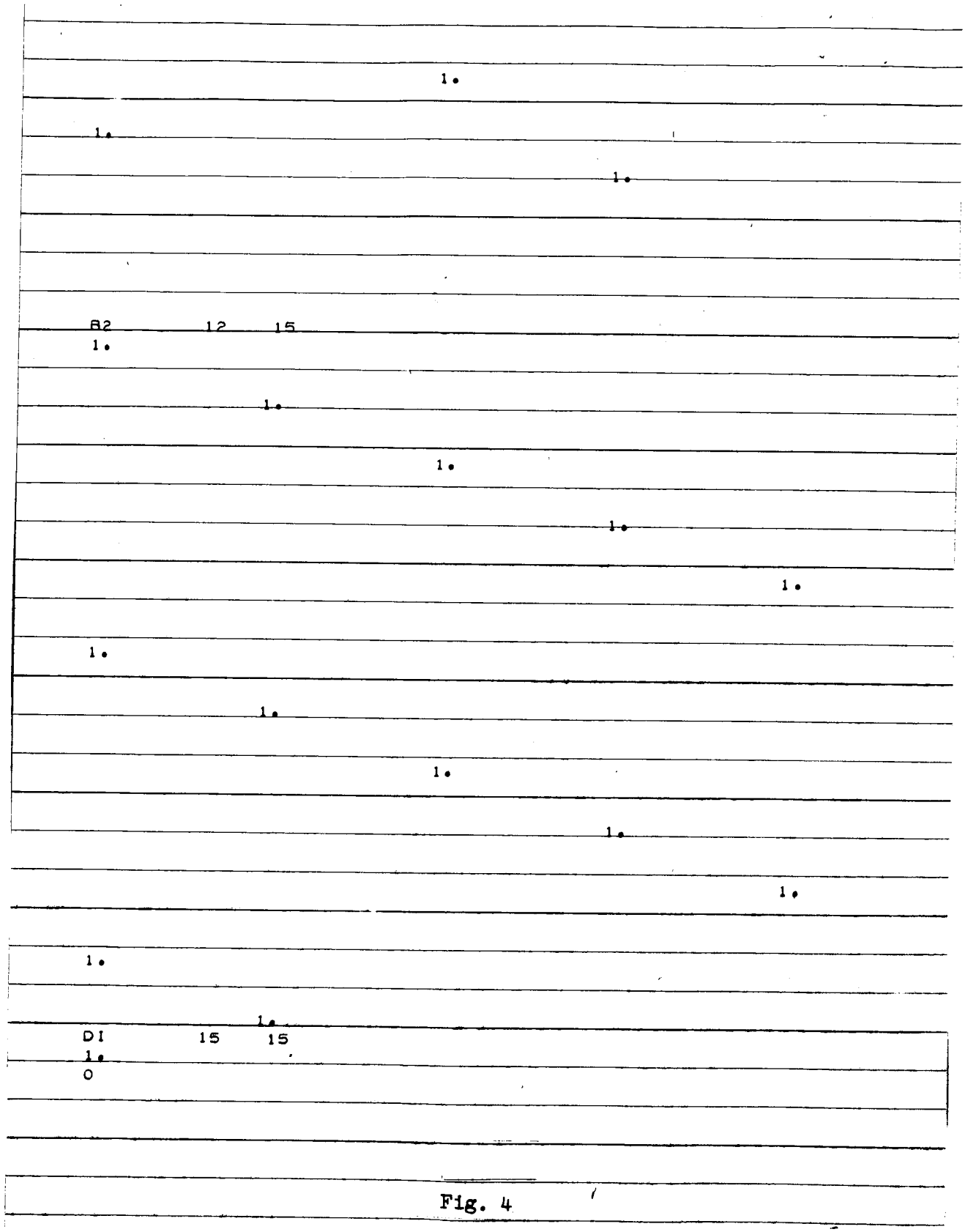

Fig. 4

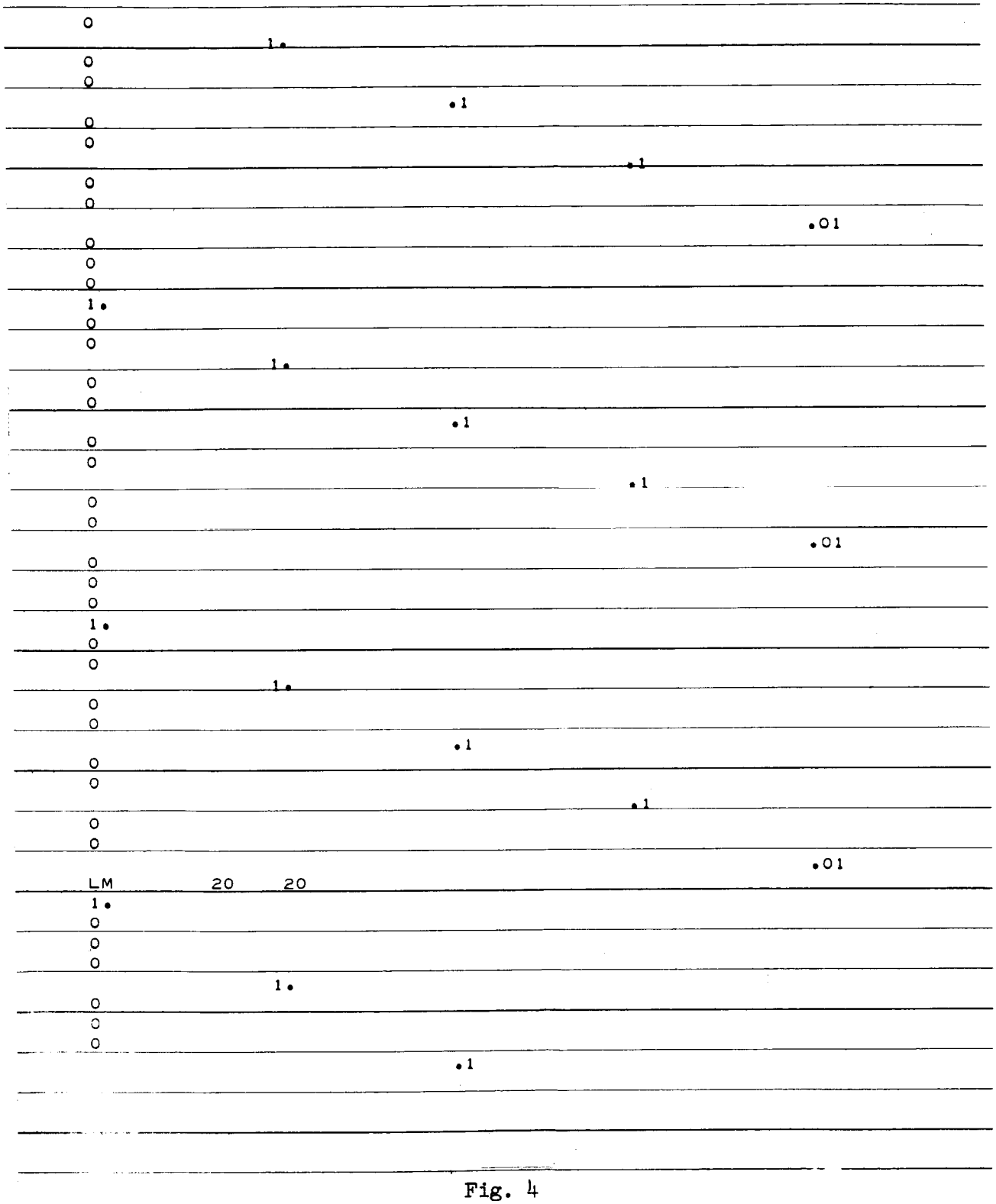
0
1.
0
0
.1
0
0
.1
0
0
.01
0
0
0
1.
0
0
1.
0
0
.1
0
0
.1
0
0
.01
0
0
0
1.
0
0
1.
0
0
.1
0
0
.1
0
0
.01
LM 20 20
1.
0
0
0
1.
0
0
0
.1

Fig. 4

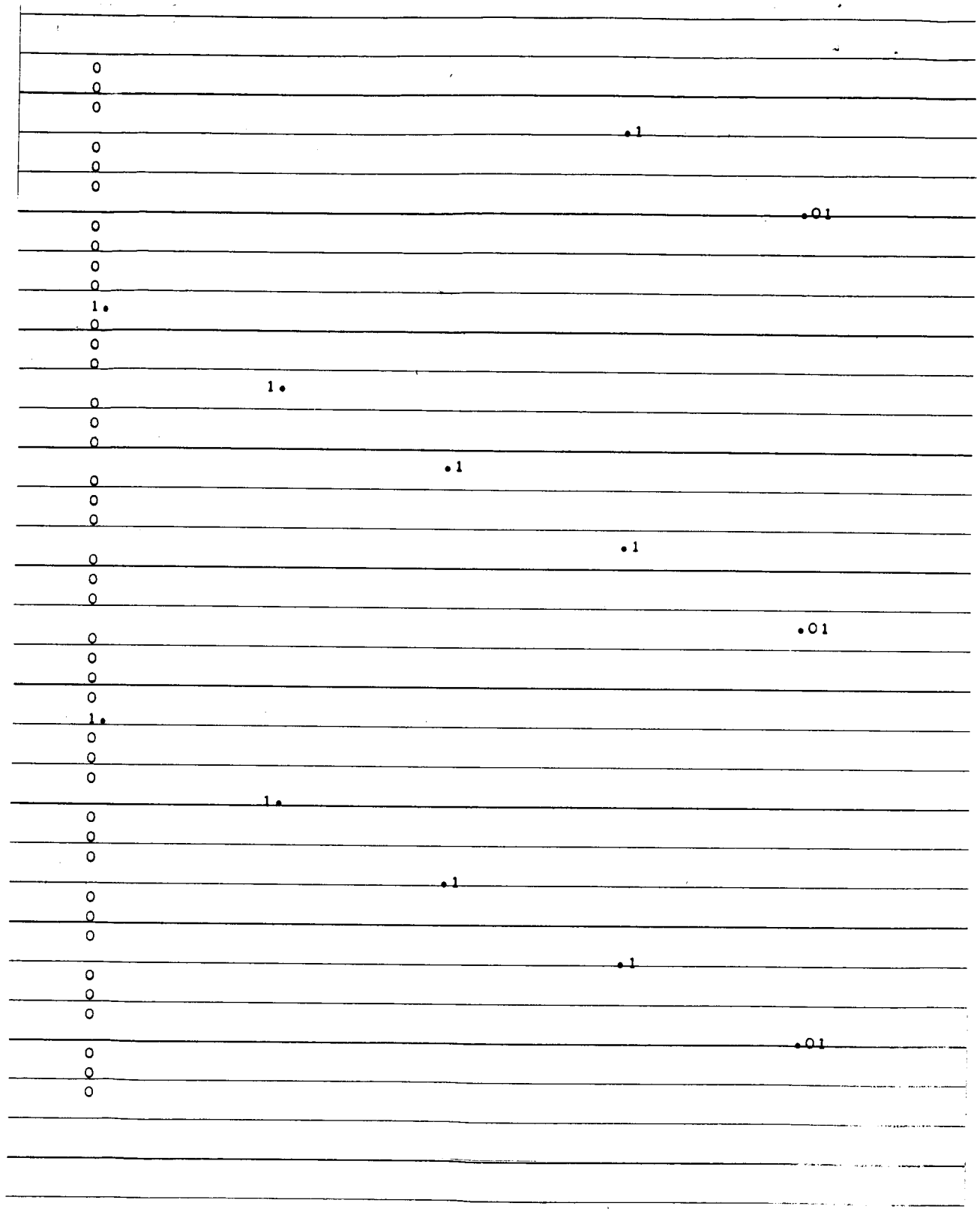

Fig. 4

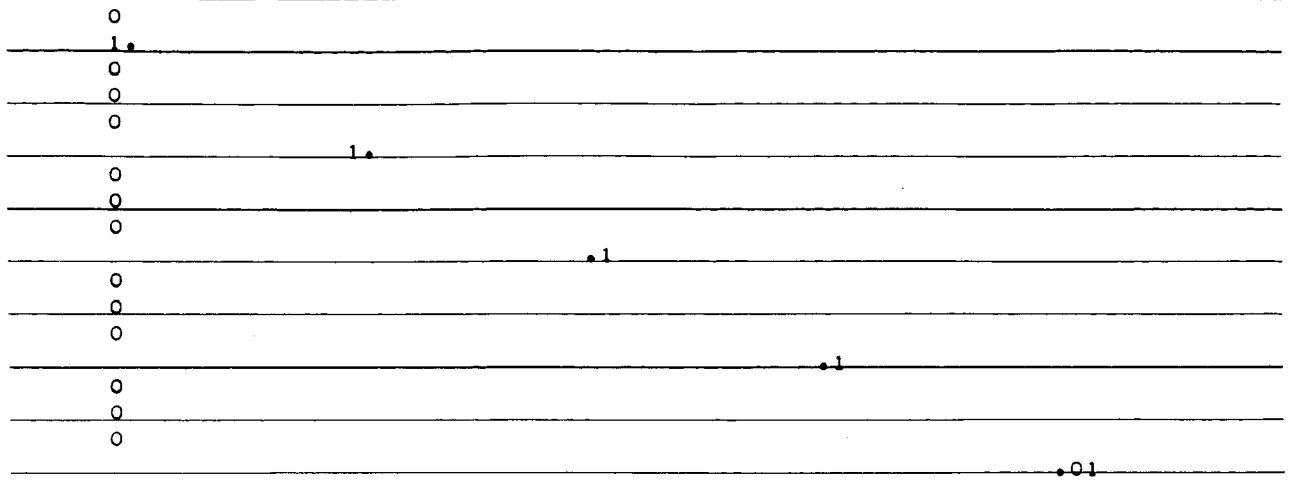

Fig. 4

CR-475

MATRIX S

NUMBER OF ROWS 3 NUMBER OF COLUMNS 4

0.30000562E 01	0.60000406E 01	0.19999760E 01	0.19999504E 01
0.56246562E-05	0.99998961E 00	-0.76427437E-04	-0.24703038E-04
0.19999890E 01	-0.19999588E 01	0.10003050E 01	0.10000072E 02

MATRIX S

NUMBER OF ROWS 3 NUMBER OF COLUMNS 4

0.29998252E 01	-0.30000342E 02	-0.69998919E 01	-0.49996567E 01
0.19999714E 01	-0.29999191E 01	0.20003461E 01	0.79999965E 01
-0.39999356E 01	0.79998185E 01	-0.30008943E 01	-0.36000139E 02

--Fig. 5

CR475

MATRIX S

NUMBER OF ROWS 3 NUMBER OF COLUMNS 4

-0.23547876E 04	0.36666603E 05	0.48401163E 04	0.69591645E 03
-0.14896059E 05	0.72841801E 03	-0.49452196E 04	-0.45416312E 05
0.29066005E 05	-0.21845774E 04	0.73730381E 03	0.94482138E 05

MATRIX S

NUMBER OF ROWS 3 NUMBER OF COLUMNS 4

0.88658751E 04	-0.14704803E 06	-0.18653721E 05	-0.19579727E 04
0.75938398E 05	-0.21837629E 04	0.15728378E 05	0.23000103E 06
-0.14969205E 06	0.65527389E 04	-0.22299267E 04	-0.47094142E 06

Fig. 6

CHAPTER XVI

APPROXIMATION OF AN IMPULSE RESPONSE

1. Description of the Problem: Given a sampled signal (i.e. a sequence of scalars), we wish to approximate it in the least squares sense by the impulse response of some other system (preferably perhaps smaller), whose dynamics are specified. The approximation procedure is to be applied to signals on any finite interval; as the interval increases, we want our approximation to change accordingly. We will treat this problem in close analogy with the general statistical filtering problem.

2. Theory and References: See, in addition to Chapter VIII

[1] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", Journal of Basic Engineering, March, 1960.

Given the signal $\{z_k\}$, $k = 0, 1, \ldots$, a vector h, and a matrix F, we wish to determine a vector $\hat{G}_N$ such that

$$\sum_{k=0}^{N} (z_k - \hat{H}e^{kF}\hat{G}_N)^2 \quad \text{is minimum for each } N = 0, 1, \ldots .$$

This problem has a well-defined solution, which is readily derived by least-squares fitting techniques (see Section 5). We will approach the problem from a slightly different point of view and obtain a solution by using an optimal filter. This will not give us the true minimum because we cannot put in the correct initial covariance matrix, but from the practical point of view the results will be very satisfactory.

3. Specific Problem: We are given a sequence of scalars $\{z_k\}$ $k = 0, 1, \ldots$ where z_k is the response at $t = 0.5k$ of a dynamical system having impulse response

$$\frac{1}{\prod_{j=1}^{8} (s+j)}.$$

CR-475

We will approximate this as the output of a system with

$$\hat{F} = \text{diag}\,(-.8,\ -1.5,\ -3.),\quad H = [1,\quad 1,\quad 1] \tag{3.1}$$

by determining the optimal estimate $\hat{x}_N$ of $x(.05N)$ with the aid of discrete filtering. Then we relate this back to $t = 0$ via the transition matrix to obtain an estimate of $\hat{x}(0)$. We may regard $\hat{x}(0)$ as equivalent to the matrix G for the system (3.1). Thus we will estimate the given impulse of a known finite-dimensional linear dynamical system, in which $\hat{F}$ and $\hat{H}$ are fixed (the latter without loss in generality) and G is to be determined, optimally.

4. Preliminary Computations:

$$F = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix},\quad G = \begin{bmatrix} 1 \\ -1 \end{bmatrix},\quad H = [1 \quad 1],$$

$$T = .1.$$

We generated a tape of the impulse response of (4.1) using the program in Fig. 1.

This specified the given signal. Then using $\hat{H} = 1$, $\hat{F} = .5$ $(R = 1)$, we ran the filtering program Fig. 2, to obtain $\hat{x}(0|t)$. The initial variance was $P_o = 100$, initial estimate $\hat{x}_o = \hat{x}(0|-1) = 0$. It follows that

$$P_{N+1} = e^{-.1}\,\frac{P_n}{P_N + 1}$$

$$K_N = e^{-.05}\,\frac{P_N}{P_N + 1}$$

$$\hat{x}_{N+1} = e^{-.05}\hat{x}_N + K_N(y_N - H\hat{x}_N)$$

$$G_{N-1} = e^{.05N}\hat{x}_N = e^{.05N}\hat{x}(0.5N|.05(N-1)).$$

The exact values of $\hat{g}_N$ are given by

$$\hat{G}_o = 0$$
$$\hat{G}_1 = .04727$$
$$\hat{G}_2 = .09311$$
$$\hat{G}_3 = .13746$$

compared with the machine results

$$\hat{G}_o = 0$$
$$\hat{G}_1 = 0.47273731$$
$$\hat{G}_2 = .093116675$$
$$\hat{G}_3 = .13746906.$$

The small loss in accuracy is apparently caused by the fact that P_1 is only accurate to six decimal places.

A second check was made using

$$F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3.6 & -8.1 & -5.3 \end{bmatrix},$$

$$G = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad H = [1 \quad 0 \quad 0],$$

to specify the impulse response

$$\frac{1}{(s + .8)(s + 1.5)(s + 3)}.$$

This was approximated with

$$F = \begin{bmatrix} -.8 & 0 & 0 \\ 0 & -1.5 & 0 \\ 0 & 0 & -3 \end{bmatrix}, \quad H = [1, \quad 1, \quad 1].$$

The two systems have the same eigenvales. $\hat{G}$ can be calculated to be

$$\hat{G} = \begin{bmatrix} .64935065 \\ -.95238095 \\ .30303030 \end{bmatrix}.$$

Using $P_o = 1000I$ $(R = 1)$, the approximation was run and converged to

$$\hat{G} = \begin{bmatrix} .64159 \\ -.93345 \\ .29022 \end{bmatrix}$$

by $t = 7.1$.

5. Procedure and Results for Specific Problem: We computed the impulse response at intervals of .05 for the transfer function (3.1) by applying the transformation

$$T = \text{diag}\ (10^4, 10^4, 10^4, 10^4, 10^4, 10^3, 100, 10)$$

to the system

$$H = [10^4,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]$$

$$G' = [0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 10^2]$$

$$F = \text{companion } [-40{,}320.,\ -109584.,\ -118{,}124.,\ -67{,}284.,$$
$$-22{,}449.,\ -4536.,\ -546.,\ -36.].$$

The corresponding signal is plotted in Fig. 3. It is obtained using the program in Fig. 4, which is essentially the same as in Fig. 1, except that the response is printed.

Then we approximated the signal with the system

$$\hat{F} = \begin{bmatrix} -.8 & 0 & 0 \\ 0 & -1.5 & 0 \\ 0 & 0 & -3. \end{bmatrix}, \qquad H = [1,\quad 1,\quad 1],$$

using the program in Fig. 5, which gives us $\hat{x}(t|t-1)$, $\hat{G}$, and the cumulative square errors. This was done by obtaining a sequence $\hat{x}(.05N|.05(N-1)) = \hat{x}_N$ which would be translated back to the origin as $\hat{G}_N$ by using the transition matrix $\Phi(0,.05N)$. The three components of $\hat{G}$ are plotted in Figs. 6 and 7.

It is of interest to have $\sum_{k=0}^{N} (z(.05k) - \hat{z}(.05k|.05k))^2$. This is is tabulated in Fig. 8, along with

$$\sum_{k=0}^{N} (z(0.5k) - \hat{z}(0.5k|.05(k^{-1})))^2.$$

It is really amazing how much better estimate is obtained by including an extra reading. The period of small increase around 4.5-5.0 seems to correspond to the region where the optimally approximating curve crosses the signal curve, see Fig. 9.

There only remains the task of displaying the approximating impulse response computed in this manner and the square error given by it. For this we can choose any value of $\hat{G}$. In Fig. 9 we have graphed the impulse response corresponding to $\hat{G}(7.)$, a point where $\hat{G}$ has attained steady state. The error is tabulated in Fig. 10. These results were obtained using the program in Fig. 12.

The response and error were also obtained for $\hat{G}(2.0)$, which is approximately two-thirds the final value of $\hat{G}$. The impulse reponse is graphed in Fig. 11, the error tabulated in Fig. 10. These results were obtained using the program in Fig. 12.

As an experiment, we let the initial variance matrix VO be $10^{17}I$. This gave a run which differed drastically from the previous run with VO = $10^{3}I$. The terminal value of $\hat{G}$ (compare with Fig. 6 and 7) was

$$\hat{G}(7) = \begin{bmatrix} 101.93869 \\ -235.47173 \\ 196.84500 \end{bmatrix}.$$

Immediately we see that the curve generated by this $\hat{G}(7)$ will have an enormous sum square error. The initial error $(63.311958)^2$ above is larger than the total error for the previous estimates (see Fig. 10); however the error given by $\hat{z}(t|t)$ is extremely small (see Fig. 13 and compare with Fig. 8). The response curve appears in Fig. 14; it has very good fit at the peak and rather poor fit at the ends.

In an attempt to improve the fit in the vicinity of the peak for the problem with VO = $10^{3}I$, we let R = 0 on the interval [1, 3], using the program in Fig. 15. This gave behavior very much like that with the large VO. The terminal value of $\hat{G}$ was

$$\hat{G}(10) = \begin{bmatrix} 117.91459 \\ -299.13265 \\ 315.48727 \end{bmatrix}.$$

The errors are given in Fig. 16 and the curve plotted in Fig. 17.

This procedure was extremely successful in giving us good results, very good results, at the peak, but associated with this is a much poorer performance at each end. These results are quite reasonable considering that we have penalized errors on [1, 3] so heavily.

6. Least Square Procedure: We will now analyze the same problem from a strictly least square approach.

Define

$$\hat{W}_N = \sum_{i=0}^{N} \hat{\Phi}^{i\prime}\hat{H}'\hat{H}\hat{\Phi}^{i}.$$

Then the problem stated in Sec. 2 has the solution

$$\hat{G}_N = \hat{W}_N^{\#} \sum_{i=0}^{N} \hat{\Phi}^{i\prime}\hat{H}'z_i.$$

Observing that

$$\hat{W}_{N+1} = \hat{H}'\hat{H} + \hat{\Phi}'\hat{W}_N\hat{\Phi} \tag{6.1}$$

or

$$\hat{W}_{N+1} = \hat{W}_N + \hat{\Phi}^{N+1\prime}\hat{H}'\hat{H}\hat{\Phi}^{N+1}, \tag{6.2}$$

we will try to obtain a recursive definition of $\hat{G}_N$.

$$\begin{aligned}\hat{G}_{N+1} &= \hat{W}_{N+1}^{\#} \sum_{i=0}^{N+1} \hat{\Phi}^{i\prime}\hat{H}'z_i \\ &= \hat{W}_{N+1}^{\#}\hat{W}_N\hat{G}_N + \hat{W}_{N+1}^{\#}\hat{\Phi}^{N+1\prime}\hat{H}'z_{N+1};\end{aligned}$$

because $\hat{W}_N\hat{W}_N^{\#}$ is the identity operator on the range of $[\hat{H}', \hat{\Phi}'\hat{H}', \hat{\Phi}^{2\prime}\hat{H}', \ldots, \hat{\Phi}^{n\prime}\hat{H}']$

$$\hat{G}_{N+1} = \hat{W}_{N+1}^{\#}(\hat{W}_{N+1} - \hat{\Phi}^{N+1\prime}\hat{H}'\hat{H}\hat{\Phi}^{N+1})\hat{G}_N + \hat{W}_{N+1}^{\#}\hat{\Phi}^{N+1\prime}H'z_{n+1}$$

from (6.2). Now $\hat{W}_{N+1}^{\#}$, $\hat{W}_{N+1}$ is the identity operator on range $\hat{W}_{N+1}$ $\supset$ range W_N. But $\hat{G}_N \in$ range $\hat{W}_N^{\#}$, so we obtain the recursion formula

$$\hat{G}_{N+1} = \hat{G}_N + \hat{W}^{\#}_{N+1}\hat{\Phi}^{N+1\prime}\hat{H}'(z_{N+1} - \hat{H}\hat{\Phi}^{N+1}\hat{G}_N)$$

which holds at every step.

Unfortunately we will find no such formula for $W^{\#}_{N+1}$. A recursion formula exists for $\hat{W}^{-1}_{N+1}$, but it requires that $\hat{\Phi}^{-1}$ and $\hat{W}^{-1}_{N}$ exist. This will mean, of course that before starting the recursion formula we must obtain n readings, find $\hat{W}^{-1}_{N}$ and then proceed.

$$\hat{W}^{-1}_{N+1} = \hat{\Phi}^{-1}[\hat{W}^{-1}_{N} - \hat{W}^{-1}_{N}\hat{\Phi}^{-1\prime}\hat{H}'(\hat{H}\hat{\Phi}^{-1}\hat{W}^{-1}_{N}\hat{\Phi}^{-1\prime}\hat{H}' + I)^{-1}\hat{H}\hat{\Phi}^{-1}\hat{W}^{-1}_{N}]\hat{\Phi}^{-1\prime}.$$

The following example shows that the recursion formula need not hold, even if $\hat{W}^{-1}_{N-1}$ exists.

Consider

$$\hat{H} = [1 \quad 0] \qquad \hat{\Phi} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

This system is completely observable since $\det \begin{bmatrix} \hat{H} \\ \hat{H}\hat{\Phi} \end{bmatrix} \neq 0.$

$$\hat{W}_o = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \hat{W}^{\#}_o = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \hat{\Phi}^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \quad \hat{\Phi}^{-1\prime} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}.$$

$$\hat{W}_1 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}, \quad \hat{W}^{-1}_1 = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}.$$

But using the recursion formula, we obtain

$$\hat{W}_1^{\#} = \begin{bmatrix} 1/2 & 0 \\ 0 & 0 \end{bmatrix}.$$

Not only is $\hat{W}_2^{\#}$ incorrect, but it is singular, and in fact all the subsequent $\hat{W}_1^{\#}$ will be singular (because $[\hat{\Phi}^{-1}, \hat{H}]$ is not completely controllable).

Using this procedure and the programs appearing in Figs. 18 and 19 we computed

$$\hat{G}(2.) = [63.124303, -118.11554, 58.159163]$$

and

$$\hat{G}(7.) = [79.965998, -153.58624, 78.573582].$$

These gave the response appearing in Fig. 20 and the errors tabulated in Fig. 21. Notice that the filter gave results quite comparable to the true minimum.

In Fig. 18 we have used 9 steps before inverting the observability matrix $\hat{W}_N$ with inversion errors of about 10^{-4}. This was required because inverting at three steps gave inversion errors of 10^{-1}.

As a test we put the same system in companion form with $\hat{H} = [1 \ 0 \ 0]$ and were able to invert at three steps with errors less than 10^{-4}. Using F in companion form with $\hat{H} = [1, 1, 1]$ we obtained inversion errors of 10^{-2}. The cumulative error was not significantly affected by these small errors. This is an interesting point which can perhaps be resolved as follows: If knowledge of the system had been perfect i.e. if we had used the $\hat{F} = F$, then our use of $10^{17}I$ as an initial covariance would probably not have affected the limit value of $\hat{G}$. However by using an incorrect value of $\hat{F}$ we were insensitive to small

errors in the sense that their contribution would be small to a total which, because of the mismatched dynamics, must be large.

On the other hand, because of the mismatched dynamics, information about the whole curve must be available. Theoretically if the dynamics are matched we need only any three points on the curve.

7. Digression: To the person interested in using digital computers to apply optimal system theory, one of the most frustrating and lingering problems is that of matrix inversion.

In discrete problems, many engineers have felt that the matrix P_{k+1} should be computed stepwise by using one column of Γ (row of H) at a time, in order to guarantee that $[\Gamma'P\Gamma + R]^{\#}$ can be accurately calculated. This procedure can be used if R is diagonal; we can prove this by remembering that Γ and Φ need not be constant matrices. In the filtering problem this corresponds to treating the output vector one component at a time with no dynamics between, a procedure which is clearly feasible if the noise in the different components is uncorrelated, i.e. if R is diagonal.

If an inversion routine is used routinely then problems can arise (e.g. $[\Gamma'P\Gamma + R]$ is always nonnegative definite but an inversion routine may very well treat "noise" of the wrong sign as legitimate numbers, rendering the result $[\Gamma'P\Gamma + R]^{-1}$ indefinite). This particular problem can be avoided by using a pseudo-inversion routine for nonnegative definite symmetric matrices which will guarantee symmetry and nonnegativity of the output. More serious is the question of the subtraction occurring in

$$P - PM'(MPM' + N)^{\#}MP.$$

If $P_k \to 0$ as $k \to \infty$, it will be difficult computationally to preserve nonnegativity even with the use of PSEUO. Using a stepwise process it may well happen that

$$P - \frac{PM'MP}{MPM'+N} \qquad (M = \text{row vector}) \tag{7.1}$$

will be numerically nonnegative. In particular if $N \neq 0$, it seems that (7.1) should always be ≥ 0.

It turns out that this process also uses less machine time and therefore appears from every viewpoint to be the desirable means of computation if N is diagonal.

In what follows we give a strictly algebraic proof that the two methods of computation are equivalent.

Let

$$S = P - PM'(MPM' + N)^{\#}MP$$

where

$$M = \begin{bmatrix} H \\ h \end{bmatrix} \quad \text{and} \quad N = \operatorname{diag}(R, r).$$

Let

$$S_1 = P - PH'(H'H' + R)^{\#}HP$$

$$S_2 = S_1 - S_1h'(hS_1h' + r)^{\#}hS_1.$$

We want to show that $S = S_2$. This reduces to proving that

$$PM'(MPM' + N)^{\#}MP = PH'(HPH' + R)^{\#}H'P$$

$$+ P[h'-H'(H'H' + R)\ H'Hh'][h(P-PH'(H'H' +R)^{\#}H')h'+r]^{\#}(h-hPH'(HPH'+R)^{\#}H)P.$$

The psuedo-inverse of a matrix is, unfortunately, not a continuous function of the elements of the matirx, hence we will need to consider several cases

1) If $r = 0$ and $hPh' = 0$, then $hP = 0$ and we prove quickly that

$$]MPM' + N]^{\#} = \begin{bmatrix} HPH' + R & HPh' \\ hPH' & hPH' + r \end{bmatrix}^{\#} = \begin{bmatrix} HPH' + R & 0 \\ 0 & 0 \end{bmatrix}^{\#}$$

$$= \begin{bmatrix} (HPH' + R)^{\#} & 0 \\ 0 & 0 \end{bmatrix}$$

2) In the most important case, when $hS_1h' + r \neq 0$, we can still prove that $(MPM' + R) = \Theta$, where

$$\Theta = \begin{bmatrix} (HPH'+R)^{\#}[I+HPh'(hS_1h'+r)^{-1}hPH'(HPH'+R)^{\#}] & -(HPH'+R)^{\#}HPh'(hS_1h'+r)^{-1} \\ \text{symmetric} & h(hS_1h'+r)^{-1} \end{bmatrix}.$$

We do this by recourse to the pseudo-inverse axioms:

$$MPM' + R = \begin{bmatrix} HPH' + R & HPh' \\ hPH' & hPh' + r \end{bmatrix}$$

$$\Theta = \begin{bmatrix} (HPH' + R)(HPH' + R)^{\#} & 0 \\ 0 & 0 \end{bmatrix}.$$

This is symmetric, which proves that axiom 4 holds.

In order to obtain this form it is necessary to know that

$$(HPH' + R)(HPH' + R)^{\#}H = H \tag{6.2}$$

This follows from Lenna (2.14), Chapter IV.

It is clear now that $\Theta(MPM' + N)\Theta = \Theta$, which proves that axiom 2 holds. Using (2.1) we see that $(MPM' + R)\Theta(MPM' + N) = (MPM' + N)$, which proves that axiom 1 holds.

There remains to show that $\Theta(MPM' + N)$ is symmetric

$$\Theta(MPM' + N) = \begin{bmatrix} (HPH' + R)^{\#}(H'H' + R) & 0 \\ 0 & 1 \end{bmatrix}$$

3) The final case is considerably less important. We have $hS_1h' + r = 0$ and $hPh' \neq 0$. Unfortunately it is no longer true that

$$(MPM' + N) = \begin{bmatrix} (HPH' + R)^{\#} & 0 \\ 0 & 0 \end{bmatrix}$$

so we will have to prove that

$$PM'(MPM' + N)^{\#}MP = PH'(HPH' + R)^{\#}HP.$$

There seems to be no royal way of doing this, so we will compute $(MPM' + N)$

$$\begin{bmatrix} HPH' + R \\ hPH' \end{bmatrix} = \begin{bmatrix} [1 - \frac{(HPH' + R)^{\#2}HPh'hPH'}{1 + HPH'(HPH' + R)^{\#2}HPH'}] (HPH' + R)^{\#} & \frac{(HPH' + R)^{\#2}HPH'}{1 + \sim} \end{bmatrix}$$

This follows from the fact that

$$(HPH' + R)(HPH' + R)^{\#\leftarrow} = (HPH' + R) .$$

(An interesting result obtained in this investigation is that $S^{\#2} = S^{2\#}$ and $S^{\#}S = SS^{\#}$ for any symmetric S). We wish to compute

$$\begin{bmatrix} HPH' + R & HPh' \\ hPH' & hPh \end{bmatrix}^{\#}$$

and before we can do so, we must determine if

$$\begin{bmatrix} HPh' \\ hPH' \end{bmatrix} = \begin{bmatrix} HPH' + R \\ hPH' \end{bmatrix} \begin{bmatrix} HPH' + R \\ hPH' \end{bmatrix}^{\#} \begin{bmatrix} HPh' \\ \end{bmatrix}$$

$$= \begin{bmatrix} HPH' + R \\ hPH' \end{bmatrix} \left[\left[I - \frac{(HPH'+R)^{\#2} HPH'hPH'}{1 + \sim} \right] (HPH'+R)\, HPh' + \frac{(HPH+R)^{\#2} HPh'hPH'}{1 + \sim} \right]$$

$$= \begin{bmatrix} HPh' - \dfrac{(HPH'+R)^{\#} HPh'hPH'(HPH'+R)^{\#} HPh' - (HPH'+R)^{\#} HPh'hPh'}{1 + hPH'(HPH' + R)^{\#2} HPh'} \\ hPH'(HPH'+R)^{\#} HPh' - \dfrac{hPH'(HPH'+R)^{\#2} HPh'(hPH'(HPH'+R)^{\#} HPh' - hPh'}{1 - hPH'(HPH'+R)^{\#2} HPh'} \end{bmatrix}.$$

If we now remember that $hS_1h' = h(P - PH'(HPH' + R)^{\#} HP)h' = 0$ we obtain

$$\begin{bmatrix} HPh' \\ hPH'(HPH' + R)^{\#} HPh' \end{bmatrix} = \begin{bmatrix} HPh' \\ hPh' \end{bmatrix}.$$

This shows why the condition $hS_1h' + r = 0$ is a natural one though it does not appear to be at first; it is important to the choice of which formula can be used in the iterative definition of the pseudo-inverse.

$$\begin{bmatrix} HPH' + R & HPh' \\ hPH' & hPh' \end{bmatrix}^{\#} = \begin{bmatrix} \begin{bmatrix} HPH'+R \\ hPH' \end{bmatrix}^{\#} \left\{ I - \begin{bmatrix} HPh' \\ hPh' \end{bmatrix} \frac{[hPH' \quad hPh']}{\alpha} \begin{bmatrix} HPH'+R \\ hPH' \end{bmatrix}^{\#'} \begin{bmatrix} HPH'+R \\ hPH' \end{bmatrix}^{\#} \right\} \\ \frac{[hPH' \quad hPh']}{\alpha} \begin{bmatrix} HPH' + R \\ hPH' \end{bmatrix}^{\#'} \begin{bmatrix} HPH' + R \\ hPH' \end{bmatrix} \end{bmatrix}$$

where

$$\alpha = 1 + [hPH' \quad hPh'] \begin{bmatrix} HPH' + R \\ hPH' \end{bmatrix}^{\#'} \begin{bmatrix} HPH' + R \\ hPH' \end{bmatrix} \begin{bmatrix} HPh' \\ hPh' \end{bmatrix}$$

but

$$\begin{bmatrix} HPH' + R \\ hPH' \end{bmatrix}^{\#} \begin{bmatrix} HPh' \\ \end{bmatrix} = (HPH' + R)^{\#}HPh' - \frac{(HPH'+R)^{\#2}HPh'[hPH'(HPH'+R)^{\#}HPh'-hPH']}{1 + \sim}$$

$$= (HPH' + R)^{\#}HPh'.$$

So $\alpha = 1 + \sim$, and

$$(MPM' + N)^{\#} = \begin{bmatrix} \begin{bmatrix} HPH' + R \\ hPH' \end{bmatrix}^{\#} - \frac{(HPH' + R)^{\#}HPh'hPH'(HPH'+R)^{\#}}{1 + \sim} \begin{bmatrix} HPH' + R \\ hPH' \end{bmatrix} \\ \frac{hPH'(HPH' + R)}{1 + \sim} \begin{bmatrix} HPH'+R \\ hPH' \end{bmatrix} \end{bmatrix}$$

$$P[H' \quad h'](MPM'+R) = \left[PH' - \frac{[PH'(HPH'+R)^{\#}HPh'-Ph']hPH'(HPH'+R)}{1+\sim} \right] \begin{bmatrix} HPH'+R \\ h'PH \end{bmatrix}$$

$$\begin{bmatrix} HPH' + R \\ hPH' \end{bmatrix}^{\#} \begin{bmatrix} H \\ h \end{bmatrix} P = (HPH' + R)^{\#}HP - \frac{(HPH'+R)^{\#2}HPh'[hPH'(HPH'+R)^{\#}HP-hP]}{1+\sim}.$$

Now consider $h[P - PH'(HPH' + R)^{\#}HP] = h\theta$.

We know that $h\theta h' = 0$ but since θ is nonnegative definite, it must be that $h\theta = 0$.

Therefore

$$P[H' \quad h'](MPM + N)^{\#} \begin{bmatrix} H \\ h \end{bmatrix} P = PH'(HPH' + R)^{\#}HP.$$

7. Adaptation by Error Minimization: This technique may be used to determine the parameters $\hat{F}$ by trial and error to minimize the least-squares fitting. As seen in Fig. 21 we had an error of 221.4 when using $\lambda(F) = -.8, -1.5, -3$. The following table shows the evolution of the system as this error is decreased.

$-\lambda_1$	$-\lambda_2$	$-\lambda_3$	ERR
.8	1.5	3.	221.4
.7	1.4	2.9	148.9
.6	1.3	2.8	147.2
.7	1.2	2.7	121.2
.7	1.2	2.5	115.5
.71	1.25	2.4	109.5
.73	1.25	2.3	103.4
.75	1.25	2.1	95.9
.77	1.27	1.9	90.0
.88	1.27	1.9	78.7

If we allow $\hat{F}$ to have one more eigenvalue, we obtain the following results:

$-\lambda_1$	$-\lambda_2$	$-\lambda_3$	$-\lambda_4$	
.6	1.	2.	2.9	176.2
.8	1.1	1.9	2.9	102.3
1.	1.2	2.	3.	41.4
1.1	1.4	2.	3.	13.2
1.2	1.4	2.	3.	8.45
1.3	1.4	2.	3.	7.78
1.28	1.4	2.	3.	7.63
1.28	1.4	2.	2.82	7.10

Like other adaptive schemes, this parameter search procedure is not as easy as the table above makes it appear. Nevertheless the search procedure can be mechanized to obtain an optimal fit over some chosen interval of the impulse response. This sequence of systems can then be patched together to provide an approximating time varying linear system.

The impulse responses of the "optimized" approximating systems appear in Fig. 22 and 23.

The following is a very rough estimate of the arithmetic operation required to compute P_{k+1}. First by treating m outputs and then by treating 1 output m times. Floating addition, multiplication and division are considered equally time consuming.

The inner product of two n vectors is assumed to required 2n operations, actually n multiplication and n-1 addition. Matrix inversion is assumed to require n^3 operations.

Stage	(i) Inversion	(ii) Single output
MP	$2mn^2$	$2n^2$
MPM'	$2mn^2 + 2m^2n$	$2n^2 + 2n$
$MPM' + N$	$2mn^2 + 2m^2n + m$	$2n^2 + 2n+1$
$(MPM' + N)^{-1}$	$2mn^2 + 2m^2n + m+m^3$	$2n^2 + 2n + 2$
$(MPM' + N)^{-1}MP$	$2mn^2 + 4m^2n + m+m^3$	$2n^2 + 3n + 2$
$PM'(MPM' + N)^{-1}MP$	$4mn^2 + 4m^2n + m+m^3$	$3n^2 + 3n + 2$
$P - PM'(MPM' + N)^{-1}MP$	$4mn^2 + 4m^2n + m+m^3+n^2$	$4n^2 + 3n + 2$

Now multiply the operations in the single stage process by m and subtract, the difference is

$$4m^2n - 3mn - m + m^3 + n^2 > 0.$$

```
        BEGIN
        LOAD    H, F, G, T,ND,Z1, I,
        REW         5
        WEF         5
        ETPHI   F, T,PH
        MULT    I, G,     1,
        REW         5
HEAD 1MULT    H, 1,     Y,
        SAVE        5 Y, Y,Z1, T,
        ADD     Z1, T,    Z1
        MULT    PH, 1,    1,
        IF      ND,Z1,HEAD 1
        END
```

H	1	1			
	1.		1.		
F	1	1			
	0		0	0	-1.
G	2	1			
	1.		-1.		
T	1	1			
	.1				
ND	1	1			
	3.				
Z1	1	1			
	0				
I	2	2			
	1.		0	0	1.

Fig. 1

```
          BEGIN
          REW          5
          LOAD   Z1, D,PC,T1,LF,V0,LH, R, Q, I,XC,ND,
          EQUAT   I, 3,
          ETPHI LF,T1,PH
          SUBT   Z1,T1,    MT,
          ETPHI LF,MT,HP,
     HEAD 1BRING       5 Y, Y,
          SAMPL PH,V0,LH, R, Q, D,PC,    V0, K,AL,
          MULT   LH,XC,    YC
          SUBT    Y,YC,    YT,
          MULT    K,YT,     1,
          MULT   PH,XC,     2,
          ADD     1, 2,    XC,
          MULT   HP, 3,     3,
          MULT    3,XC,    GC,
          RINT   Z1,    GC,GC
          ADD    Z1,T1,    Z1
          IF     ND,Z1,HEAD 1
          END
     Z1          1      1
                   0
      D          1      2
                   0                .1
     PC          1      4
                   0                0              0              1.
     T1          1      1
                  .1
     LF          1      1
                 -.5
     V0          1      1
                100.
     LH          1      1
               1.
      R          1      1
             1.
      Q          1      1
           0
      I          1      1
               1.
     XC          1      1
             0
     ND          1      1
           2.
```

Fig. 2

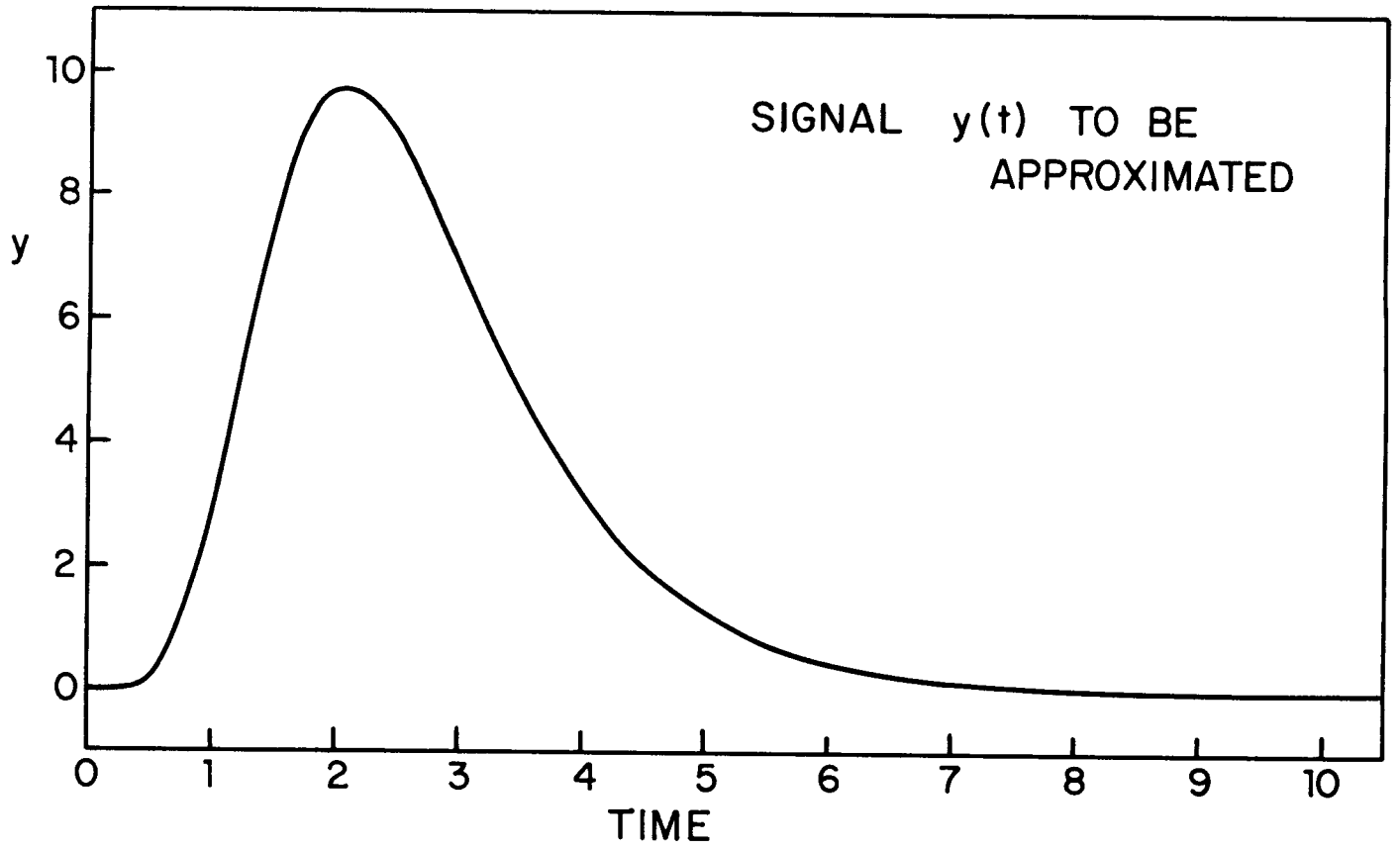

Fig. 3

```
        BEGIN
        LOAD    H, F, G, T,ND,Z1, I, J, K, R,GM,TT
        REW         5
        WEF         5
        ETPHI   F, T,PH
        MULT    I, G,     1,
        TRNSI   J, K, R, G,PH,GM, H,TT
        REW         5
HEAD 1MULT      H, 1,     Y,
        SAVE        5 Y, Y,Z1, T,
        ADD     Z1, T,    Z1
        MULT    PH, 1,    1,
        IF      ND,Z1,HEAD 1
        END
H               1       8
                1.              0               0               0               0
                0               0               0
F               8       8
                0               1.              0               0               0
                0               0               0               0               0
                1.              0               0               0               0
                0               0               0               0               1.
                0               0               0               0               0
                0               0               0               1.              0
                0               0               0               0               0
                0               0               10.             0               0
                0               0               0               0               0
                0               10.             0               0               0
                0               0               0               0               0
                10.     -40.32          -109.584        -118.124        -67.284
-22.449         -45.36          -54.6           -36.
G               8       1
                0               0               0               0               0
        0               0               1000.
T               1       1
            .05
ND              1       1
        10.
Zi              1       1
                0
I8              8       8
                1.              0               0               0               0
                0               0               0               0               1.
                0               0               0               0               0
                0               0               0               1.              0
                0               0               0               0               0
                0               0               1.              0               0
                0               0               0               0               0
                0               1.              0               0               0
                0               0               0               0               0
                1.              0               0               0               0
                0               0               0               0               1.
                0               0               0               0               0
                0               0               0               1.
J               1       1
                0
K               1       8
                0               0               0               0               0
                0               0               0
R               1       1
                0
GM              8       1
```

Fig. 4

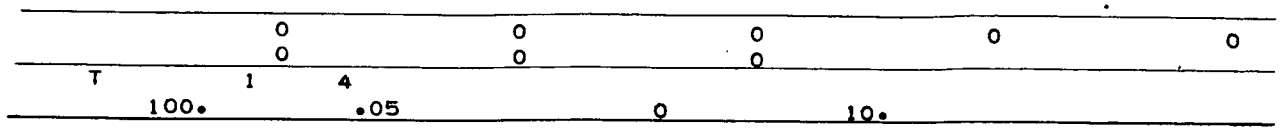

Fig. 4

```
        JAN. 22, 1965
        BEGIN
        REW          5
        LOAD  Z1, D,PC,T1,LF,V0,LH, R, Q, I,XC,ND,
        EQUAT  1, 3,Z1,Z2,
        ETPHI LF,T1,PH
        SUBT  Z1,T1,    MT,
        ETPHI LF,MT,HP,
HEAD 1BRING       5 Y, Y,
        SAMPL PH,V0,LH, R, Q, D,PC,   V0, K,AL,
        MULT  LH,XC,    YC
        SUBT   Y,YC,    YT,
        MULT   K,YT,    1,
        MULT  PH,XC,    2,
        ADD    1, 2,    XC,
        MULT  HP, 3,    3,
        MULT   3,XC,    GC,
        MULT  HP XC     7
        MULT  LH  7     YC
        SUBT   Y YC     YT
        MULT  YT,YT,    Y2
        ADD   Z2,Y2,    Z2,
        TRANP GC,    GT,
        TRANP XC,    XT,
        JUXTC GT,XT,    4,
        JUXTC  4,Z2,    5,
        RINT  Z1,    5, X
        ADD   Z1,T1,   Z1
        IF    ND,Z1,HEAD 1
        END
Z1          1      1
              0
D           1      2
              0              .05
PC          1      4
              0              0              0              1.
T1          1      1
            .05
LF          3      3
            -.8              0              0              0              -1.5
              0              0              0             -3.
V0          3      3
          1000.              0              0              0              1000.
              0              0              0           1000.
LH          1      3
             1.              1.             1.
R           1      1
             1.
Q           3      3
              0              0              0              0              0
              0              0              0              0
I           3      3
             1.              0              0              0              1.
              0              0              0             1.
XC          3      1
              0              0              0
ND          1      1
           10.
```

Fig. 5

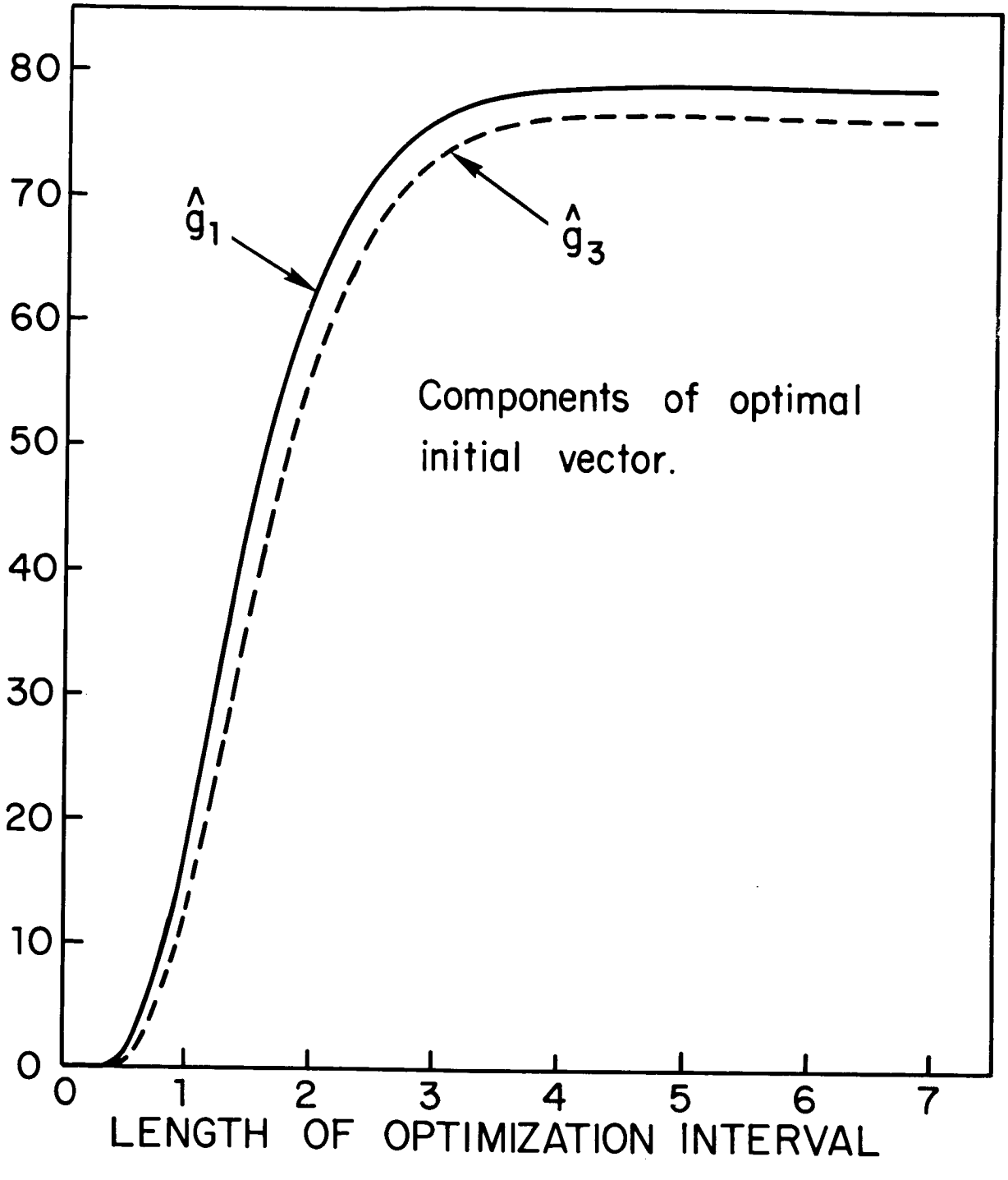
80
70
60
50
40
30
20
10
0
0
1
2
3
4
5
6
7
$\hat{g}_1$
$\hat{g}_3$
Components of optimal initial vector.
LENGTH OF OPTIMIZATION INTERVAL

Fig. 6

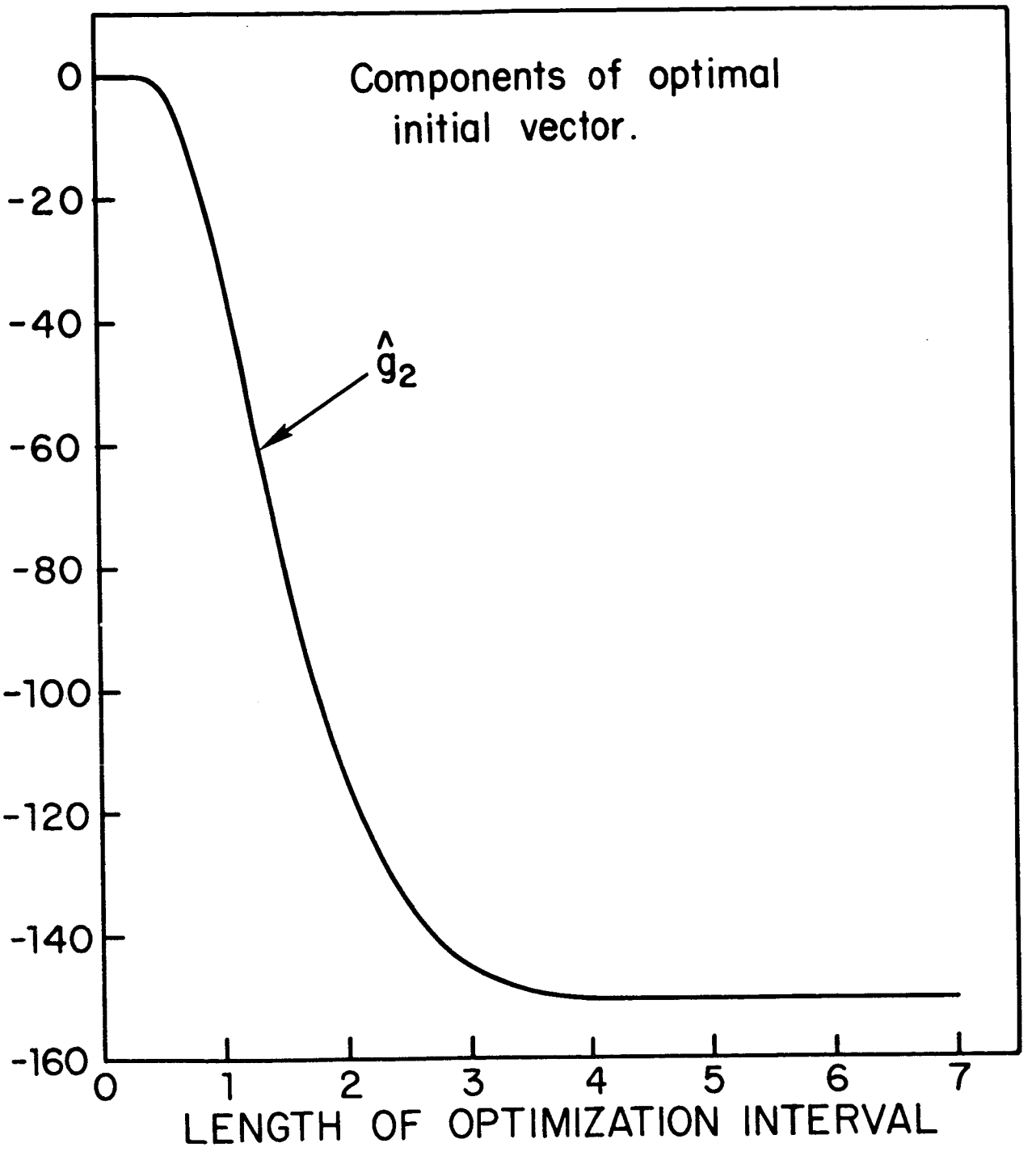
Components of optimal
initial vector.
$\hat{g}_2$
0
-20
-40
-60
-80
-100
-120
-140
-160
0
1
2
3
4
5
6
7
LENGTH OF OPTIMIZATION INTERVAL

Fig. 7

$.05N = T$	$\sum_{k=0}^{N} (z(.05k) - \hat{z}(.05k\|(k-1)))^2$	$\sum_{k=0}^{N} (z(.05k) - \hat{z}(.05k\|.05k))^2$
.5	.02	.01
1.0	5.37	3.42
1.5	48.59	36.93
2.0	134.88	112.23
2.5	212.03	184.01
3.0	253.13	223.59
3.5	267.85	238.03
4.0	271.32	241.47
4.5	271.72	241.86
5.0	271.74	241.89
5.5	271.89	242.03
6.0	272.10	242.25
6.5	272.29	242.44
7.0	272.43	242.58
7.5	272.52	242.66

Fig. 8

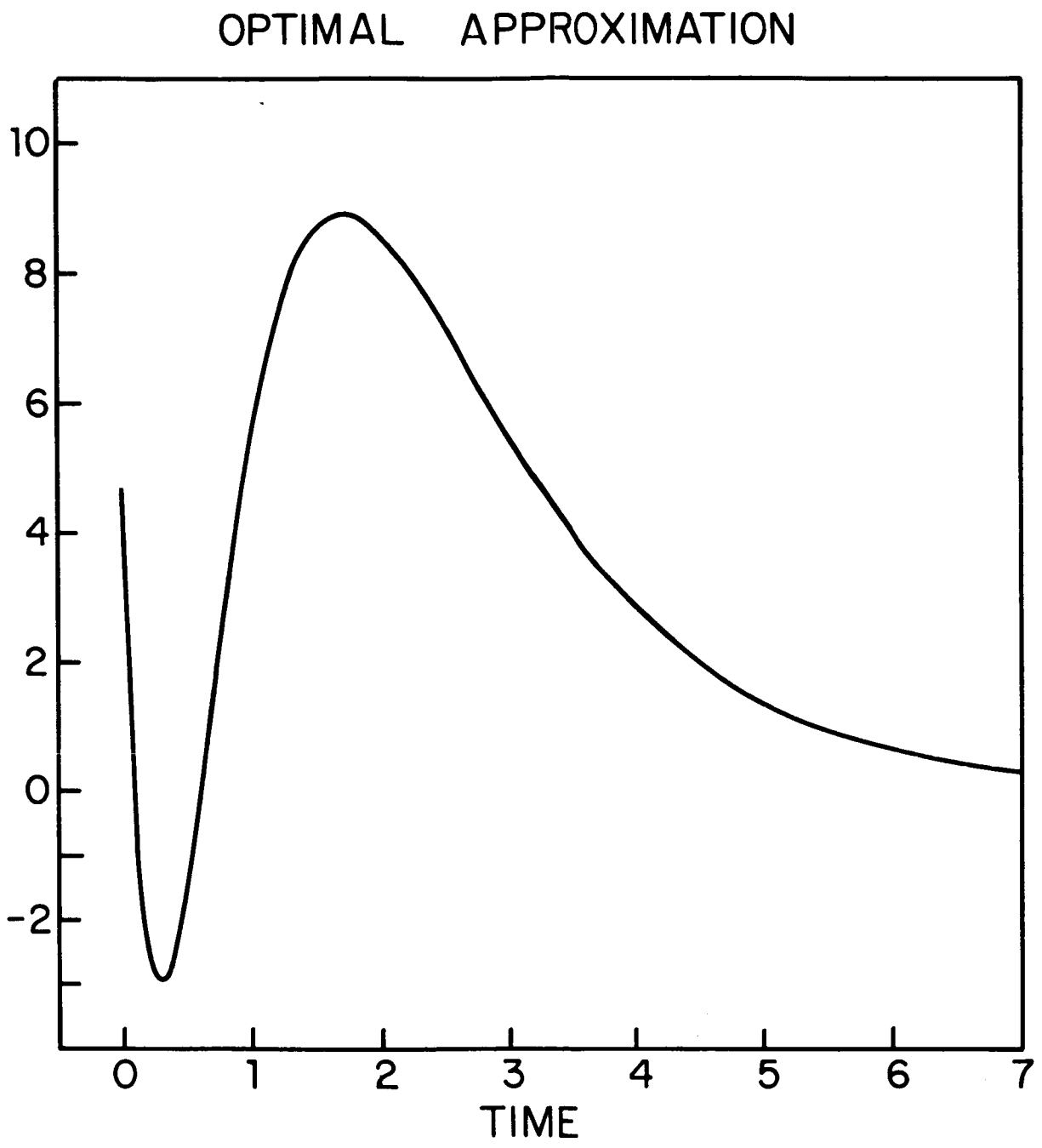

Fig. 9

$.05N = T$	$\sum_{k=0}^{N} (z(.05k) - \hat{z}(.05k \mid 7.))^2$	$\sum_{k=0}^{N} (z(.05k) - \hat{z}(.05k \mid 2))^2$
.5	68.59	26.43
1.0	93.42	48.18
1.5	146.19	60.57
2.0	150.75	103.18
2.5	175.06	208.78
3.0	203.22	297.47
3.5	217.07	342.65
4.0	220.94	359.62
4.5	221.46	364.64
5.0	221.49	365.80
5.5	221.62	365.97
6.0	221.83	365.98
6.5	222.02	365.99
7.0	222.17	366.00
7.5	222.26	366.02

Fig. 10

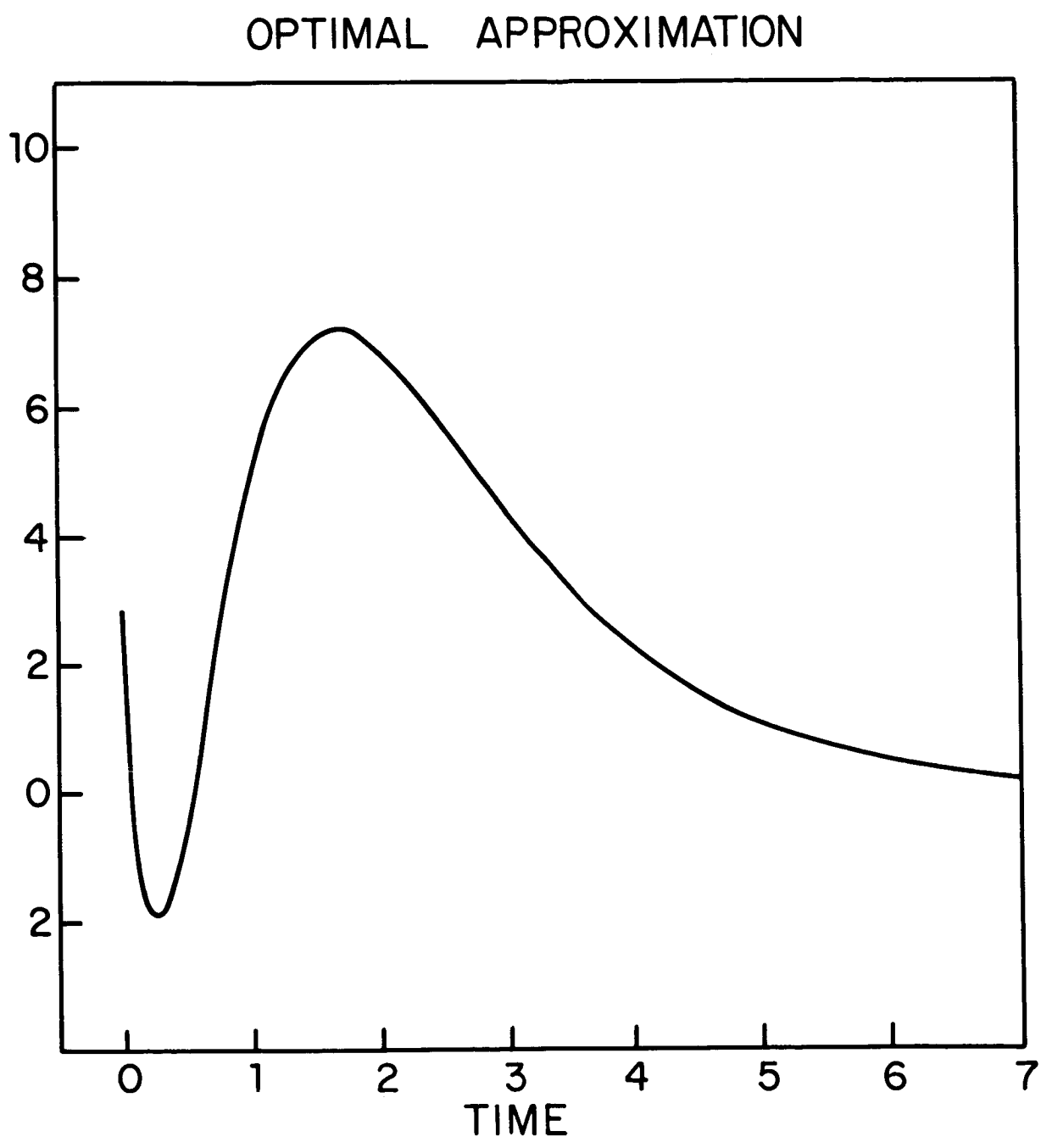

Fig. 11

```
        BEGIN
        REW          5
        LOAD   Z1, T,ND, F, H,XC,
        ETPHI  F, T,PH,
        EQUAT Z1,Z2,
HEAD 1BRING        5 Y, Y,
        MULT    H,XC,    YC,
        SUBT    Y,YC,    YT,
        MULT   YT,YT,    Y2,
        ADD    Z2,Y2,    Z2,
        JUXTC YC Z2      Z3
        RINT   Z1     Z3 ERR
        MULT   PH XC     XC
        ADD    Z1, T,    Z1,
        IF     ND,Z1,HEAD 1
        END
   Z1          1      1
                 0
    T          1      1
              .05
   ND          1      1
              7.5
    F          3      3
             -.8                  0                0              0          -1.5
                 0                0                0          -3.
    H          1      3
              1.                 1.               1.
   XC          3      1
   61.21139        -113.73065        55.368904
```

Fig. 12

.05N = T	$\Sigma \tilde{z}^2(t\|t)$	$\Sigma \tilde{z}^2(t\|7)$
.5	$.2.10^{-5}$	9365.
1.0	.18	9368.
1.5	1.20	9371.
2.0	5.45	9372.
2.5	9.10	9375.
3.0	9.86	9377.
3.5	10.1	9377.
4.0	11.5	9378.
4.5	13.9	9380.
5.0	16.3	9382.
5.5	18.2	9384.
6.0	19.5	9385.
6.5	20.3	9386.
7.0	20.8	9387.
7.5	21.1	9387.

$V0 = 10^{17}$ R = 1

Fig. 13

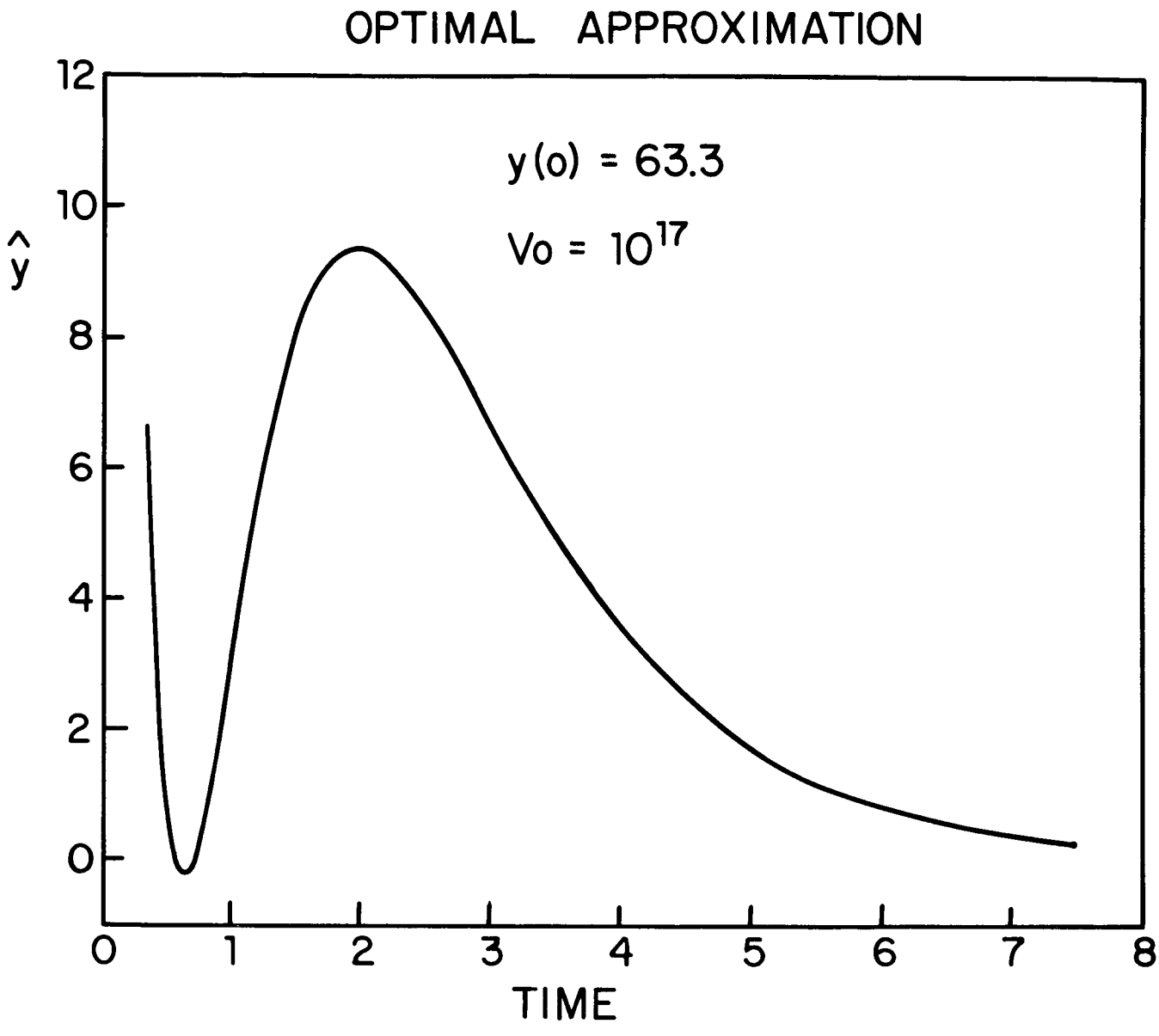
OPTIMAL APPROXIMATION
y(o) = 63.3
Vo = 10^17
ŷ
12
10
8
6
4
2
0
0
1
2
3
4
5
6
7
8
TIME

Fig. 14

```
        BEGIN
        REW          5
        LOAD   Z1, D,PC,T1,LF,V0,LH, R, Q,  I,XC,ND,N2,N3,R1,R2
        EQUAT   I, 3,Z1,Z2,Z1,Z3,
        ETPHI LF,T1,PH
        SUBT   Z1,T1,    MT,
        ETPHI LF,MT,HP,
HEAD 1BRING       5 Y, Y,
        SAMPL PH,V0,LH, R, Q, D,PC,    V0, K,AL,
        MULT   LH,XC,    YC
        SUBT    Y,YC,    YT,
        MULT   YT,YT,    Y2
        ADD    Z3 Y2     Z3
        MULT    K,YT,     1,
        MULT   PH,XC,     2,
        ADD     1, 2,    XC,
        MULT   HP, 3,     3,
        MULT    3,XC,    GC,
        MULT   HP XC      7
        MULT   LH  7      YC
        SUBT    Y YC      YT
        MULT   YT,YT,     Y2
        ADD    Z2,Y2,     Z2,
        TRANP GC,    GT,
        TRANP XC,    XT,
        JUXTC GT,XT,      4,
        JUXTC  4,Z2,      5,
        JUXTC  5 Z3,      6,
        RINT   Z1,     6, X,
        ADD    Z1,T1,     Z1
        IF     ND,Z1,HEAD 1
        IF     ND,N2,HEAD 2
        EQUAT N2,ND,R1, R,
        IF     Z1,Z1,HEAD 1
HEAD 2IF       ND,N3,HEAD 3
        EQUAT N3,ND R2, R
        IF     Z1,Z1,HEAD 1
HEAD 3END
Z1          1       1
              0
 D          1       2
              0                .05
PC          1       4
              0                0                0                1.
T1          1       1
            .05
LF          3       3
            -.8              0                0                0               -1.5
              0              0                0               -3.
V0          3       3
          1000.              0                0                0             1000.
              0              0                0            1000.
LH          1       3
```

Fig. 15

	1•		1•	1•		
R	1	1				
	1•					
Q	3	3				
	0		0	0	0	0
	0		0	0	0	
I	3	3				
	1•		0	0	0	1•
	0		0	0	1•	
XC	3	1				
	0		0	0		
ND	1	1				
	1•					
N2	1	1				
	3•					
N3	1	1				
	10•					
R1	1	1				
	0					
R2	1	1				
	1•					

Fig. 15

.05N = T	$\Sigma \tilde{z}^2(t\|t)$	$\Sigma \tilde{z}^2(t\|10)$
.5	.011	47457.
1.0	3.42	47501.
1.5	3.42	47503.
2.0	3.42	47503.
2.5	3.48	47503.
3.0	4.77	47504.
3.5	9.26	47509.
4.0	16.4	47516.
4.5	23.8	47524.
5.0	29.8	47530.
5.5	34.0	47534.
6.0	36.6	47536.
6.5	38.1	47538.
7.0	38.9	47539.
7.5	39.3	47539.

V0 = 1000 R = 0 on [1, 3]

Fig. 16

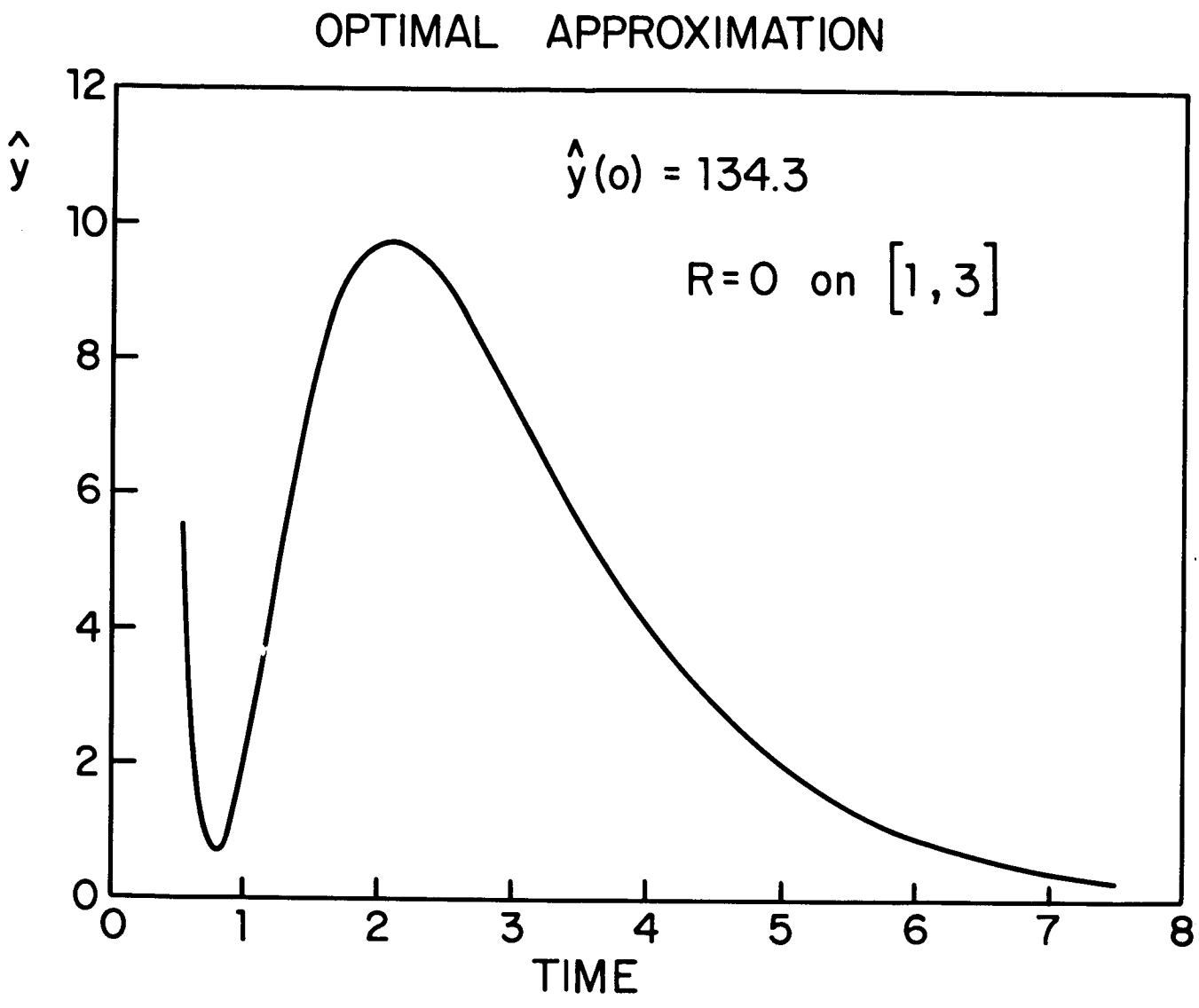
OPTIMAL APPROXIMATION
$\hat{y}$
$\hat{y}(o) = 134.3$
R=O on [1,3]
12
10
8
6
4
2
0
0
1
2
3
4
5
6
7
8
TIME

Fig. 17

```
BEGIN
LOAD    T, F,Z1, H, P,P2,X0,I1,Z3, D,PC,ND,
REW          5
PIZER Z1,P2,
ETPHI   F, T,PH,
MULT    H,PH,     G,
MULT    G,PH      2
MULT    2 PH     10
MULT   10 PH     20
MULT   20 PH     30
MULT   30 PH     35
MULT   35 PH     40
MULT   40 PH     45
JUXTR   H  G      3
JUXTR   3  2      4
JUXTR   4 10     11
JUXTR  11 20     21
JUXTR  21 30     31
JUXTR  31 35     36
JUXTR  36 40     41
JUXTR  41 45     46
TRANP  46      5
MULT    5 46      6
PSEUO   6,+    P RK PRINT
INVRS   6,     9
MULT    6  P     PP
MULT    6  9     PQ
BRING        5 Y,Y0, Y,Y1, Y,Y2, Y,Y3, Y,Y4, Y,Y5, Y,Y6, Y,Y7, Y,Y8
JUXTR Y0,Y1,      7
JUXTR   7 Y2      8
JUXTR   8 Y3     12
JUXTR  12 Y4     22
JUXTR  22 Y5     32
JUXTR  32 Y6     37
JUXTR  37 Y7     42
JUXTR  42 Y8     47
MULT    5 47      8
MULT    P  8     X0
RINT   PP IP PQ II X0 X0, 6, W
BLOT    2
TRANP  PH,    PT
TRANP   H     HT
MULT   PT HT     TR
MULT   PT TR     TR
MULT   PT TR     TR
MULT   PT TR     TR
MULT   PT TR     TR
MULT   PT TR     TR
MULT   PT TR     TR
MULT   PT TR,    TR,
SUBT   Z1  T     MT
ETPHI   F MT HP
MULT    H HP      G
```

Fig. 18

```
        ADD     Z1   T      Z1
        ADD     Z1   T      Z1
        ADD     Z1   T      Z1
        ADD     Z1   T      Z1
        ADD     Z1   T      Z1
        ADD     Z1   T      Z1
        ADD     Z1   T      Z1
        ADD     Z1   T      Z1
HEAD 1BRING          5 Y    Y
        SAMPL HP     P    G I1 Z3, D,PC,      P, K,AL
        MULT  PT TR       TR
        MULT   P TR        K
        TRANP TR       TT
        MULT  TT X0       YC
        SUBT   Y YC       ER
        MULT   K ER       XE
        ADD   X0 XE       X0
        TRANP X0       XT
        ADD     Z1   T      Z1
        RINT    Z1      XT X0
        IF      ND Z1 HEAD 1
        END
 T              1       1
.05
 F              3       3
-.8                                                          -1.5
0                       0                   0          -3.
Z1              1       1
0
 H              1       3
1.                      1.                  1.
 P              3       3
0
0
P2              1       1
1.
X0              3       1
0
I1              1       1
1.
Z3              3       3
0
0
 D              1       2
0                      .05
PC              1       4
0                                                    1.
ND              1       1
7.
```

Fig. 18

```
        BEGIN
        LOAD   Z1, T,ND, F, H,XC,
        REW            5
        ETPHI   F  T PH
        EQUAT Z1,Z2,
HEAD 1BRING          5 Y, Y
        MULT    H XC      YC
        SUBT    Y YC      YT
        MULT   YT YT      Y2
        ADD    Z2 Y2      Z2
        JUXTC  YC Z2      Z3
        JUXTC  Z3  Y      Z4
        RINT   Z1      Z4 ERR
        MULT   PH XC      XC
        ADD    Z1  T      Z1
        IF     ND Z1 HEAD 1
        END
Z1              1       1
0
 T              1       1
.05
ND              1       1
7.
 F              3       3
-.8                                                        -1.5
                                                    -3.
 H              1       3
1.                    1.                  1.
XC              3       1
79.965997          -153.58624          78.573582
```

Fig. 19

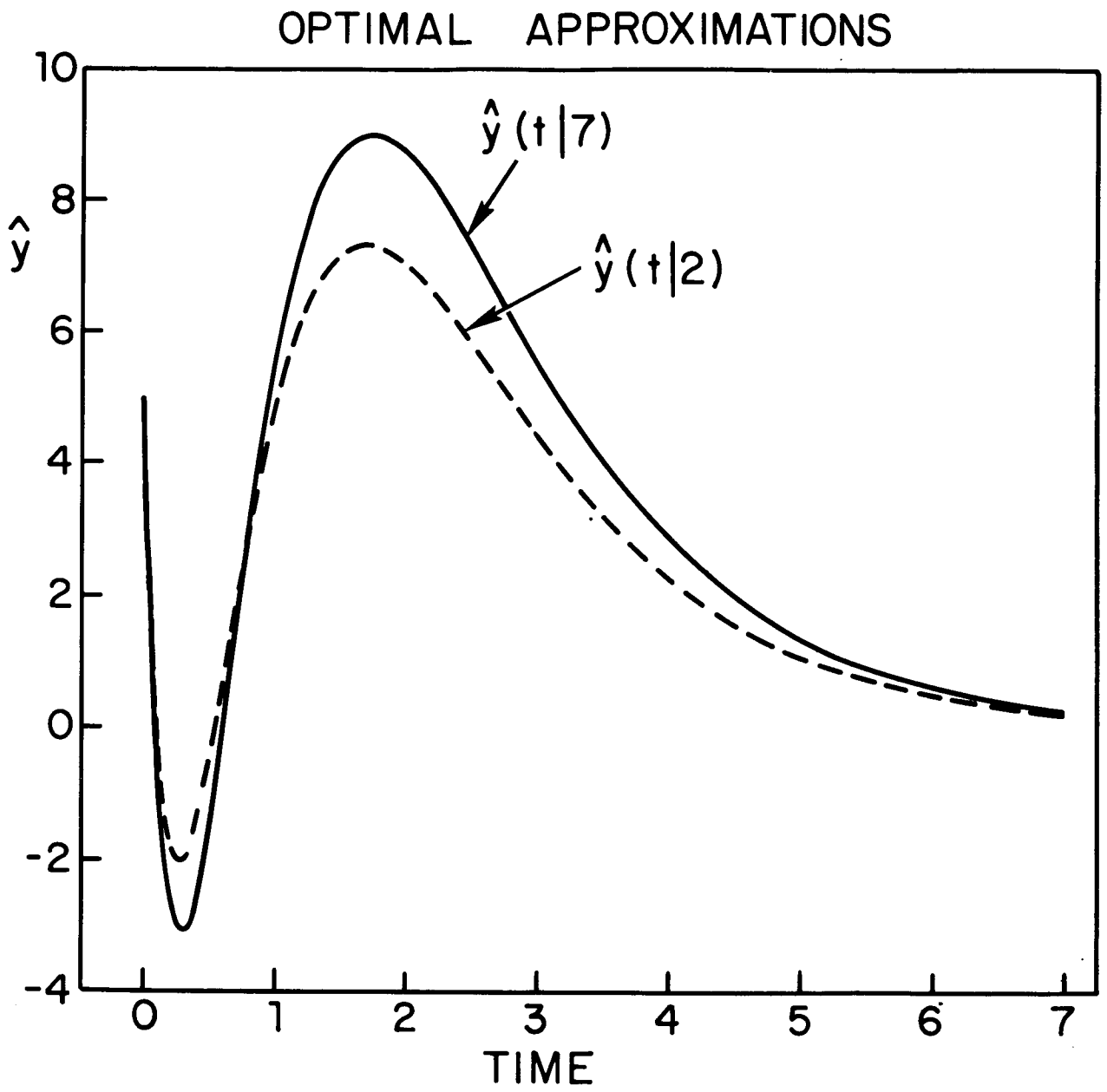

Fig. 20

.05N = T	$\Sigma \tilde{z}^2(t\|2)$	$\Sigma \tilde{z}^2(t\|7)$
.5	31.8	78.5
1.0	52.6	107.0
1.5	66.2	155.4
2.0	102.4	159.5
2.5	197.4	182.5
3.0	278.0	206.6
3.5	318.7	217.6
4.0	333.7	220.3
4.5	337.9	220.6
5.0	338.8	220.6
5.5	338.9	220.8
6.0	338.9	221.1
6.5	339.0	221.3
7.0	339.0	221.4

Fig. 21

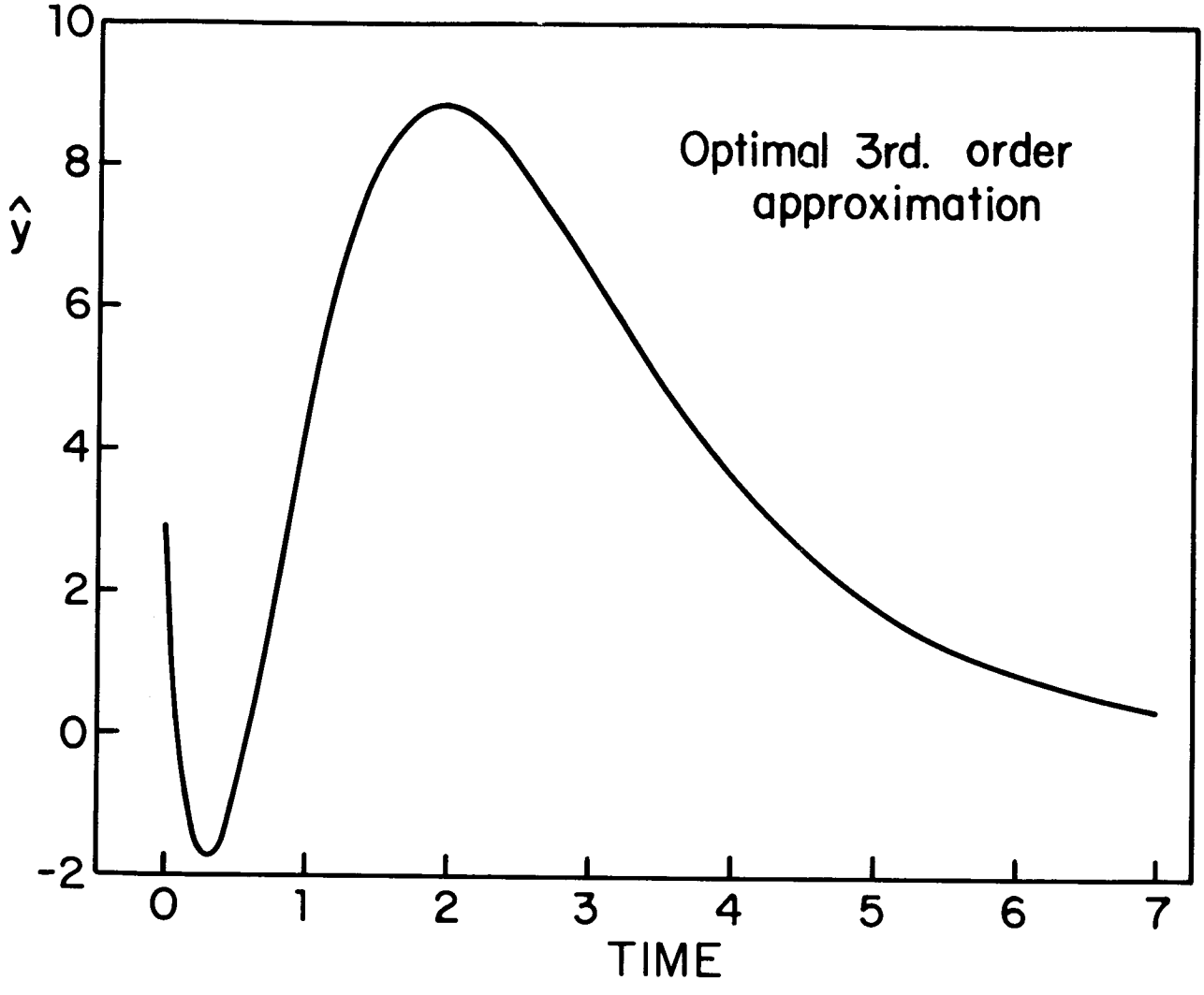

Fig. 22

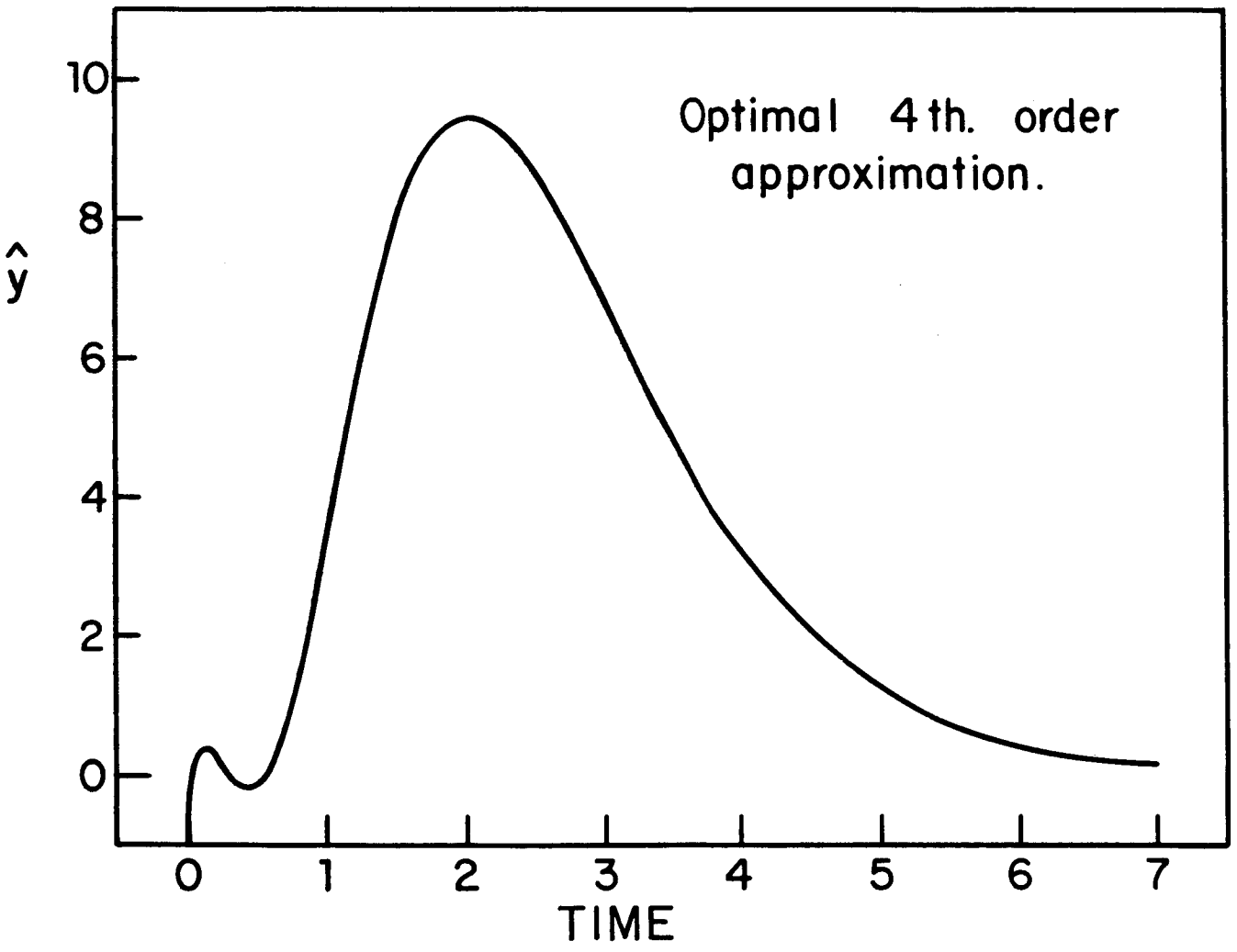

Fig. 23

CHAPTER XVII

MULTI-RATE SAMPLING

1. Description of the Problem: Frequently information about the state of a dynamical system is available only at a discrete set of points in time. This fact has given rise to the study of discretized systems. To complicate matters, however, there are times when sample and hold elements in a system operate at two or more different frequencies.

In this chapter we will examine how we may produce an optimal control for a system having an internal sampling period distinct from the period at which control can be changed.

Parenthetically, we will examine the relation of the continuous performance index to sampled ones, with and without the additional (internal) sampler.

2. Theory and references: In addition to Chapter VIII, see R. E. Kalman and J. E. Bertram, "A Unified Approach to the Theory of Sampling Systems", Jour. Franklin Institute, vol. 267, May 1959.

3. The Specific Problem: We take the flow diagram of Fig. 1, where

$$S_1 = \frac{-10(s-1)}{s(s^2+2s+8)}$$

$$S_2 = \frac{10}{2}$$

$$S_3 = \frac{1}{(s+3)(s+4)},$$

and synthesize it by the flow diagram of Fig. 2, having

$$F = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -8 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -12 & 0 & 1 \\ 0 & 0 & 0 & 1 & -7 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} 10 \\ -10 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

We wish to optimize this continuous system with respect to the performance index

$$x'(0)Px(0) = \int_0^\infty [(y + 2\dot{y})^2 + .01u^2]dt,$$

having

$$Q = \text{diag}\,(0,\ 0,\ 0,\ 0,\ 0,\ \begin{bmatrix} 1 & .2 \\ .2 & .04 \end{bmatrix})$$

$$R = .01.$$

Having done this we will introduce a sampler on the control with period τ and examine the performance index matrix P_τ,

$$x'(0)P_\tau x(0) = \sum_{i=1}^{\infty} x'(i\tau)\tau Q x(i\tau) + u'((i-1)\tau)\tau R u((i-1)\tau)$$

as τ goes from $\frac{2}{e} = .73575888$, to zero. By the reasoning given in Chapter XI we realize that in the continuous control problem we may replace Q by $Q = \text{diag}\,(0, 0, 0, 0, 0, \begin{bmatrix} 1 & 0 \\ 0 & .04 \end{bmatrix})$; we will see if this can be done in the discrete case as well.

We will then, using our multi-rate program introduce a new state variable x_8 to describe a sample and hold element operating at a period of 2 between x_7 and x_4. Because the two sampling periods are incommensurable, the transition matrix between control points is neither convergent nor periodic, so the optimal gains do not converge. We will take a total interval of 25 over which to optimize.

The open loop transfer function is

$$\frac{x_7(s)}{u(a)} = -\frac{100(s-1)(s+3)(s+4)}{s(s^2 + 2s + 8)(s^4 + 7s^3 + 125^2 - 10)}$$

having poles at $0, \quad -1 \pm \sqrt{-7}, \sim .75, \sim -4.75$.

We will assume that both sample and hold elements operate at $t = 0$.

4. Results: Using the program appearing in Fig. 3, we ran the free motion which gave, as expected, stable behavior in x_1, x_2, and x_3 and instability in the other components.

We then computed the optimal control for the continuous system, using the program in Fig. 4. This converged in 9.47 seconds to a performance index with diagonal elements:

$$p_{11} = 16.233963$$

$$p_{22} = 16.121959$$

$$p_{33} = 14.866118$$

$$p_{44} = 4.857859$$

$$p_{55} = 4.407092$$

$$p_{66} = 1822.1928$$

$$p_{77} = 12.586129.$$

Using the program in Fig. 5 we computed sampled control for various values of τ. This generated the following table illustrating how the sampled performance index converges to the continuous one

τ	p_{11}	p_{66}	p_{77}	Time for convergence
.7257588	39.500149	3872.7168	24.452950	12.51
.3678944	20.613631	2244.5554	15.730421	11.40
.18393972	17.365082	1935.2903	14.137003	11.04
.09196986	16.639765	1864.3951	13.795817	10.66
.04598493	16.454767	1846.0598	13.716736	10.19
.022992414	16.384925	1838.8439	13.683984	9.78
.011496207	16.340233	1833.9453	13.655727	9.46

It appears from the experimental results in this case, that the sampled control is also independent of whether the error is $(y + 2\dot{y})^2$ or $y^2 + 4\dot{y}^2$.

Our multi-rate program appears in Fig. 6, three runs were made with this and checked by using the ordinary sampled program.

The first, with T1 = T2 = .73575888 was checked with Fig. 7 and showed agreement to five decimal places. This simulates operation of the internal sampler infinitesimally prior to the control sampler.

The diagonal terms of P for this system were,

$$p_{11} = 23.758$$
$$p_{22} = 22.105$$
$$p_{33} = 14.325$$
$$p_{44} = 6.5906$$
$$p_{55} = 4.3283$$
$$p_{66} = 2103.97$$
$$p_{77} = 7.9393$$
$$p_{88} = 1.9325.$$

The remarkable point here is the decrease in the performance index caused merely by providing an extra, constant state variable prior to x_4.

Here the diagonal terms of P were much smaller, probably because, once the loop is broken, the system is asymptotically stable.

$$p_{11} = 9.9194$$
$$p_{22} = 8.8411$$
$$p_{33} = 4.7264$$
$$p_{44} = 2.6170$$
$$p_{55} = 1.4585$$
$$p_{66} = 776.93$$
$$p_{77} = 2.5821$$
$$p_{88} = 2.9742.$$

The second with T1 = .73575888 and T2 = 50 was checked with Fig. 8 and showed agreement to four decimal places. This simulates failure of the internal sampler even to operate. Notice that even though the state of the system will not go to zero (if $x_8(0) \neq 0$), the performance index is finite.

The third case is that of our problem, T1 = .73575888, T2 = 2.

This gave a performance index of

$$p_{11} = 12.631$$
$$p_{12} = 11.175$$
$$p_{33} = 5.6149$$
$$p_{44} = 3.3069$$
$$p_{55} = 1.7685$$
$$p_{66} = 971.03$$
$$p_{77} = 3.0681$$
$$p_{88} = 2.8250.$$

In Fig. 9 appear the transients, from unit initial condition on x_7, of the output as given by the free system, the controlled continuous system, the controlled discrete system, and the controlled multi-rate system.

In Fig. 10 appear the optimal control function required for continuous control and ordinary discrete control of the transients in Fig. 9, with the given performance index.

t	Ordinary	Multi-rate
0	-1.091	-.219
$\frac{2}{e}$	1.610	.475
$\frac{4}{e}$	-1.110	-.514
$\frac{6}{e}$	.954	.421
$\frac{8}{e}$	-.622	-.304
$\frac{10}{e}$	.404	.221
$\frac{12}{e}$	-.237	-.123
$\frac{14}{e}$	.146	.071
$\frac{16}{e}$	-.085	-.045
$\frac{18}{e}$	.052	.025
$\frac{20}{e} = 7.358$	-.030	-.015.

Surprisingly enough, the gains for the multi-rate did not converge, to the extent that variations were only about 10% of some average value. It is possible therefore to compare the steady-state gains for the three controlled runs.

	Continuous	Sampled	Multi-rate
k_1	56.65	1.727	.9915
k_2	55.36	1.464	.7168
k_3	46.99	1.054	.2558
k_4	30.26	.8225	.4106
k_5	25.58	.6168	.1835
k_6	555.1	14.63	6.209
k_7	40.95	1.091	.2186
k_8			.4637.

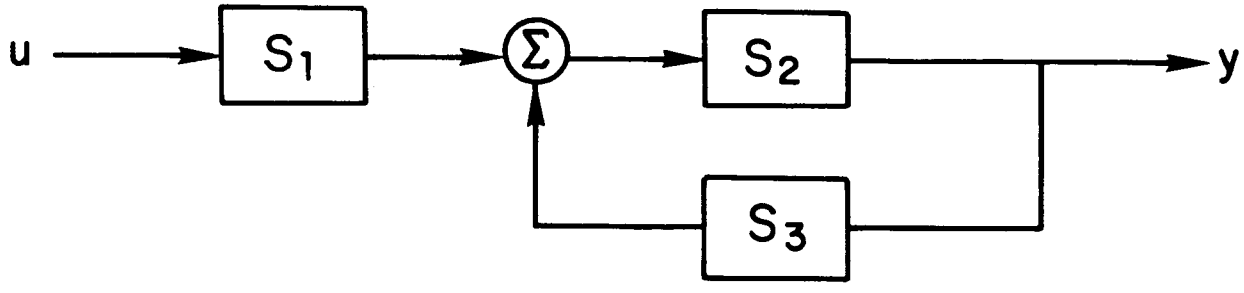

Fig. 1

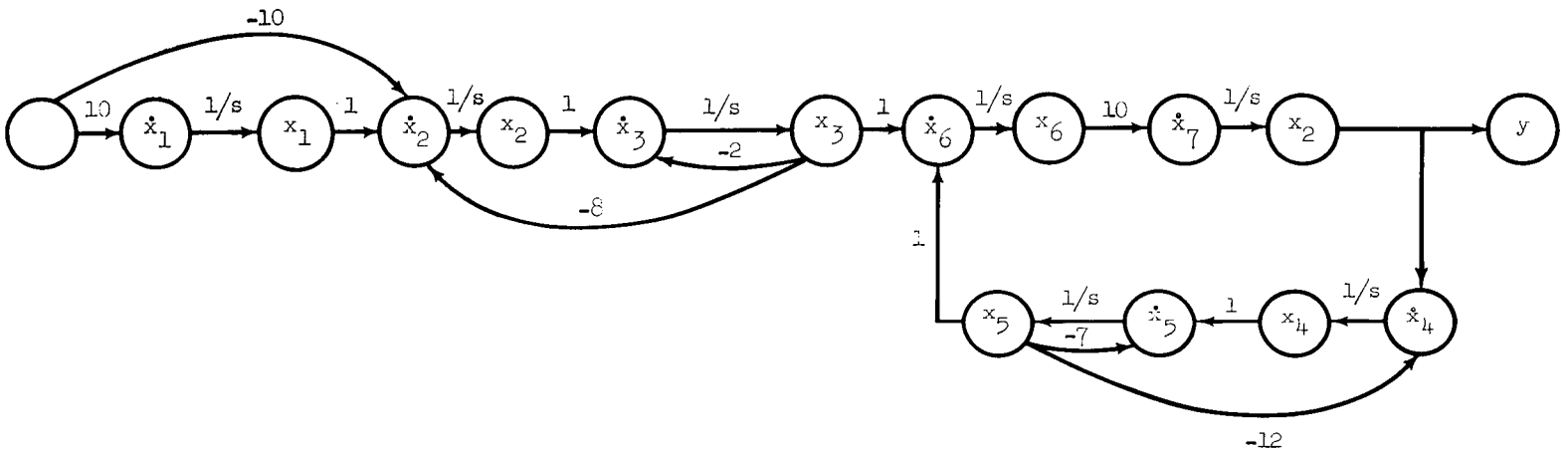

Fig. 2

```
        BEGIN
        LOAD    F   T   I Z1 ND
        ETPHI   F   T PH     PRINT
HEAD 1RINT  Z1.     I   X
        MULT    PH  I       I
        ADD     Z1  T       Z1
        IF      ND Z1 HEAD 1
        END
 F              7       7
0
0                                       1.                      -8.
0
1.                      -2.
0
-12.                                    1.
0                       1.              -7.
0                                       1.                      1.
0
0                                       10.
 T              1       1
   .25
17              7       7
1.
                                                        1.
0
                        1.
                                                                1.
0
0                                       1.
0
1.
                                                        1.
Z1              1       1
0
ND              1       1
24.9995
```

Fig. 3

```
BEGIN
LOAD  TT, D,D1,PC, F, G, Q,RI,ZR,T1,XO,ND,R1,RZ,XX
SUBT  ZR, F,     1,
TRANP  F,    FT,
TRANP  G,    GT,
MULT  RI,GT,     C,
MULT   G, C,     5,
NORM   Q,    NQ,
NORM   5,    N5,
PSEUO NQ,    PQ,RK
MULT  PQ,N5,    M1,
DECOM M1,     S,SJ,ER,PE, E,RK,
MULT   S, Q,     Q,
MULT  SJ, 5,     5,
MULT  SJ, C,     C,
BLOT  NQ,
JUXTC  1, 5,     2,
JUXTC  Q,FT,     3,
JUXTR  2, 3,    PH,
NORM  PH,    NP,
PSEUO NP,    NP,RK
MULT  NP,TT,     T
RINT  PH,PH
ETPHI PH, T,PH
MULT   T,D1,     4,
ADD    4, D,     D,
RICAT  Q,PH, C, D,PC,XX, P, K,AL,
RINT   P,PER K, K
WRITE       THE PRECEDING MATRICES WERE THE MATRIX P OF THE
WRITE    PERFORMANCE INDEX AND THE FEEDBACK GAIN MATRIX K.
MULT   G, K,    GK,
SUBT   F,GK,    CF,
ETPHI CF,T1,P1,
HEAD 1MULT   K,XO,    KX,
JUXTR XO,RZ,     6,
JUXTR  6,KX,     7,
RINT  R1,     7, X
MULT  P1,XO,    XO,
ADD   R1,T1,    R1,
IF    ND,R1,HEAD 1
IF    TT,TT,HEAD 2
WRITE       RICAT CONVERGENCE TEST REMOVED BECAUSE WITH Q = 4,1
WRITE    RI = 100. CONVERGENCE WAS NOT ACHIEVED.
HEAD 2END
          1      1
       8.
          3      1
       .0001              0              0
          3      1
             0           -1.         -1000.
          4      1
100.           100.           1.           1000.
 F        7      7
```

Fig. 4

```
0
0                                   1.                                  -8.
0
1.              -2.
0
-12.                                1.
0               1.                  -7.
0                                   1.                                  1.
0
0                                   10.
 G          7       1
10.             -10.                0
0
 0          7       7
0
0
0
0
0
0
0
0
.4
                                                                        10.
RI          1       1
10.
Z7          7       7
0
0
0
0
0
0
0
0
0
T1          1       1
.25
I7          7       7
1.
                                                                        1.
0
                    1.                                                  1.
0
0                                   1.
0
1.
                                                                        1.
ND          1       1
24.9995
Z1          1       1
```

Fig. 4

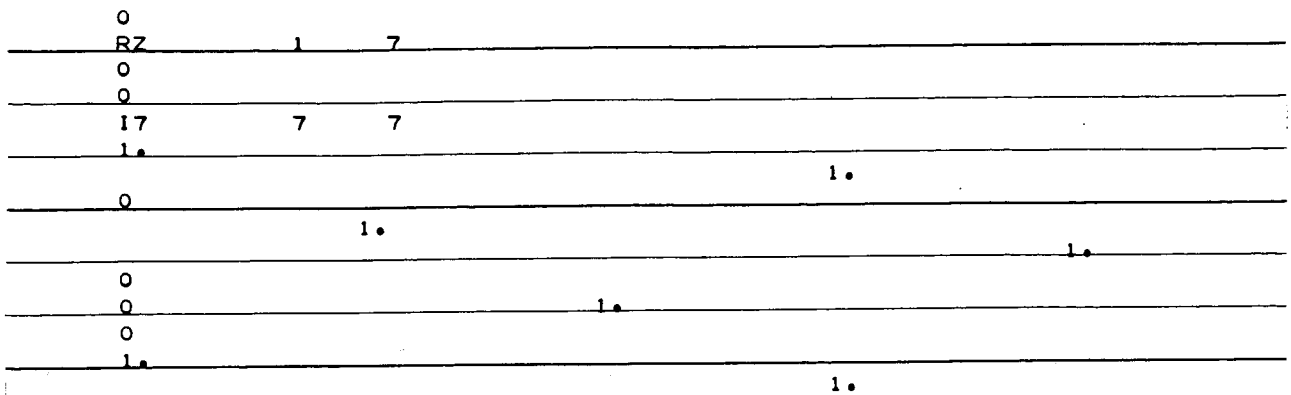

Fig. 4

```
          JULY 10, 1965
          BEGIN
          LOAD    T, F, G, D,PC,R1,Q1,RZ,Z1,ND,X0
HEAD 2MULT    T,R1,     R
          MULT    T,Q1,     Q
          EAT     F, T,PH,IN
          MULT   IN   G     GM
          TRANP PH,    FT
          TRANP GM     GT
          EQUAT  Q   P
          SAMPL FT, P,GT, R, Q, D,PC,    PE,KT,AL
          TRANP KT      K
          SUBT   PE   Q      P
          RINT    P PER K   K
          LOAD    T
          IF      T, T,HEAD 2
          END
 T              1      1
.09196986
 F              7      7
0
0                                        1.                                    -8.
0
1.                    -2.
0
-12.                                     1.
0                     1.                 -7.
0                                        1.                                    1.
0
0                                        10.
 G              7      1
10.                   -10.               0
0
 D              1      2
.0001                 .18393972
PC              1      4
10.                   10.                1.                 1000.
 R              1      1
.01
 Q              7      7
0
0
0
0
0
0
0
0
.04
0                                                                   1.
RZ              1      7
0
0
```

Fig. 5

```
Z1          1     1
0
ND          1     1
26.
I7          7     7
1.
                                                    1.
0
                    1.
                                                              1.
0
0                                  1.
0
1.
                                                    1.
 T          1     1
.04598493
 T          1     1
.022992415
 T          1     1
.0114962076
 T          1     1
.0057481038
```

Fig. 5

```
          BEGIN
          LOAD  S1,S2,ND,T1,T2, Q, R, F,,5,T0, D,PC,K0, G,X0,RZ D2 EP P1 CD
          LOAD  CE
          REW          5
          WEF          5
          REW          5
          EQUAT S1, 1,S2, 2,ND, T,ND,C2,
HEAD 1    ADD   T1, 1,     1,
          IF    ND, 1,HEAD 1                BE CAREFUL HERE WITH T3 IF ND
          SUBT   1,T1,     1                IS A MULTIPLE OF T1.
          SUBT  ND, 1,     T3
          MULT  T3, Q,     P0,
          MULT  T3  R      R1
          MULT  T1  Q,     Q1
HEAD 2    ADD   T2  2      2
          IF    ND  2 HEAD 2
          SUBT   2,T2,     2,
          EAT    F,T2,P6,P7,
          MULT  P7  G,     P8
          MULT  D2 P6      P9                THESE TWO INSTRUCTIONS ARE A
          MULT  D2 P8      P5                SAVING IF T1 IS GREATER THAN T2.
          IF     2, 1,HEAD 3
          SUBT   T, 1      T3
          EAT    F,T3,PH,IN
          MULT  IN  G      G3
          TRANP PH,    PH
          TRANP G3     GM
          SAMPL PH,P0,GM,R1,Q1, D,PC,    P0,K1,AL,
          TRANP K1       K
          MULT  T1, R      R1
          SAVE         5 K, K
          EQUAT  1, T,
          SUBT   1 T1      1
          IF    ,5,,5,HEAD 4
HEAD 3    SUBT   T, 2,     T3,
          EAT    F,T3,PH,IN,
          MULT  IN  G      G3,
          EQUAT  2  T
          SUBT   2 T2,     2,
HEAD 6    IF     1  2,HEAD 5
          MULT  PH,P5,     G2,
          ADD   G2 G3      G3
          MULT  PH P9      PH
          EQUAT  2  T
          SUBT   2 T2      2
          IF    ,5,,5,HEAD 6
HEAD24    MULT  T1, R,     R1
          EQUAT T0 C2
          IF    ,5 ,5 HEAD 4
HEAD 5    SUBT   T, 1,     T3,
          EAT    F,T3,P1  IN,
          MULT  PH,D2,     P2
          MULT  IN  G,     G1
```

Fig. 6

```
        MULT   P2 G1,    G2,
        ADD    G2 G3     G3,
        MULT   P2 P1     PH
        EQUAT   1  T
        TRANP PH     PH
        TRANP G3,    GM
        SAMPL PH P0 GM R1 Q1   D PC,    P0,K1,AL,
        TRANP K1       K
        SAVE          5 K, K
        SUBT    1,T1,     1,
        IF     C2 ND HEAD24
HEAD 4IF      T0, 1,HEAD 7
HEAD10IF       2  1 HEAD 8
        SUBT    T, 1,    T3,
        EAT     F T3 PH IN
        MULT   IN  G     G3,
        TRANP PH     PH
        TRANP G3     GM
        SAMPL PH P0 GM R1 Q1   D PC,    P0,K1,AL
        TRANP K1       K
        SAVE          5 K   K
        EQUAT   1  T
        SUBT    1 T1,     1
        IF     ,5,,5,HEAD 4
HEAD 8SUBT     T, 2,    T3
        EAT     F T3,PH,IN,
        MULT   IN, G,    G3,
        EQUAT   2, T,
        SUBT    2 T2,     2
HEAD23IF       1  2 HEAD22
        MULT   PH P5     G2
        ADD    G2 G3     G3
        MULT   PH P9     PH
        EQUAT   2  T
        SUBT    2 T2      2
        IF     ,5,,5 HEAD23
HEAD22SUBT     T, 1,    T3,
        EAT     F T3 P1 IN
        MULT   PH D2     P2
        MULT   IN  G     G1
        MULT   P2 G1     G2
        ADD    G2 G3     G3
        MULT   P2 P1     PH
        EQUAT   1  T
        TRANP PH,    PH
        TRANP G3,    GM
        SAMPL PH,P0,GM R1 Q1, D,PC,    P0,K1,AL
        TRANP K1       K
        SAVE          5 K, K
        SUBT    1,T1,     1
        IF     ,5,,5,HEAD 4
HEAD 7SUBT    P0,Q1,    19,
        RINT   19 19
```

Fig. 6

```
        IF      D S1,HEAD 9
        SAVE        5K0, K
        IF     .5,.5,OUT 1                    TRANSFER TO TRANSIENT.
HEAD41EQUAT T0, 1,
        IF      D S2,HEAD42HEAD10
HEAD42MULT  .5 T2,   T3,
        MULT  EP T2      S
        SUBT  T0, S      S
HEAD44SUBT    2  S     T4
        IF     T3,T4,HEAD43
        ADD    T2  S     S
        IF     .5 .5 HEAD44
HEAD43EQUAT   S  2
        IF     .5 .5,HEAD10
HEAD 9MULT    .5 T1     T3
        SUBT    T T0,    T4
        IF     T4 T3 HEAD41
OUT  1IF       D,S1,HEAD11                    BEGIN TRANSIENT.
        EQUAT S1, 1
        IF     .5,.5,HEAD12
HEAD11EQUAT T1, 1,
HEAD12IF       D S2,HEAD13
        EQUAT S2  2
        IF     .5 .5 HEAD14
HEAD13EQUAT T2, 2
HEAD14BSR      1  5
        EQUAT T0, T,T0,PT,
        BRING       5 K, K,
        ADD    PT PI     PT
        MULT    K X0,    U
        SUBT   RZ  U     U
        BSR      2  5
HEAD19JUXTR   K RZ     X1
        JUXTR X1 X0     X2
        JUXTR X2 RZ     X3
        JUXTR X3  U     X4
        RINT    T,    X4,MAT
HEAD18IF      PT, 1,HEAD15
        IF     PT, 2,HEAD16
        IF     PT,ND HEAD17
        SUBT   PT  T,    T3,
        EQUAT PT  T
        ADD    PI,PT,    PT,
        EAT     F,T3,PH,IN
        MULT   IN  G     G3
        MULT   G3  U     GU
        MULT   PH X0     XT
        ADD    XT GU     X0
        IF     .5 .5 HEAD19
HEAD16IF       2 ND HEAD17
        SUBT    2  T     T3
        EQUAT   2  T
        EAT     F T3 PH IN
```

Fig. 6

```
          ADD     2 T2       2
          MULT   IN  G      G3
          MULT   G3, U,     GU,
          MULT   PH X0      XT
          ADD    XT GU      XT
          MULT   D2 XT      X0
          IF      D CD HEAD18HEAD19
HEAD15IF        1  2,HEAD16
          IF      1 ND HEAD17
          SUBT    1  T,     T3
          EQUAT   1  T
          ADD     1 T1       1
          EAT     F T3 PH IN
          MULT   IN  G      G3
          MULT   G3  U      GU
          MULT   PH X0      XT
          ADD    XT GU      X0
          BRING        5 K, K
          BSR       2  5
          MULT    K X0,     U
          SUBT   RZ, U,     U
          IF      D CE HEAD18HEAD19
HEAD17SUBT   ND   T,     T3,
          EAT     F T3,PH IN
          MULT   IN  G      G3
          MULT   G3  U      GU
          MULT   PH X0      XT
          ADD     T GU      X0
          JUXTR   K RZ      X1
          JUXTR  X1 X0      X2
          JUXTR  X2 RZ      X3
          JUXTR  X3  U      X4
          RINT   ND      X4,MAT
          END
Z1             1       1
                0
Z1             1       1
0
ND             1       1
25.
T1             1       1
.73575888
T2             1       1
2.
 Q             8       8
0
0
0
0
0
0
0
0
```

Fig. 6

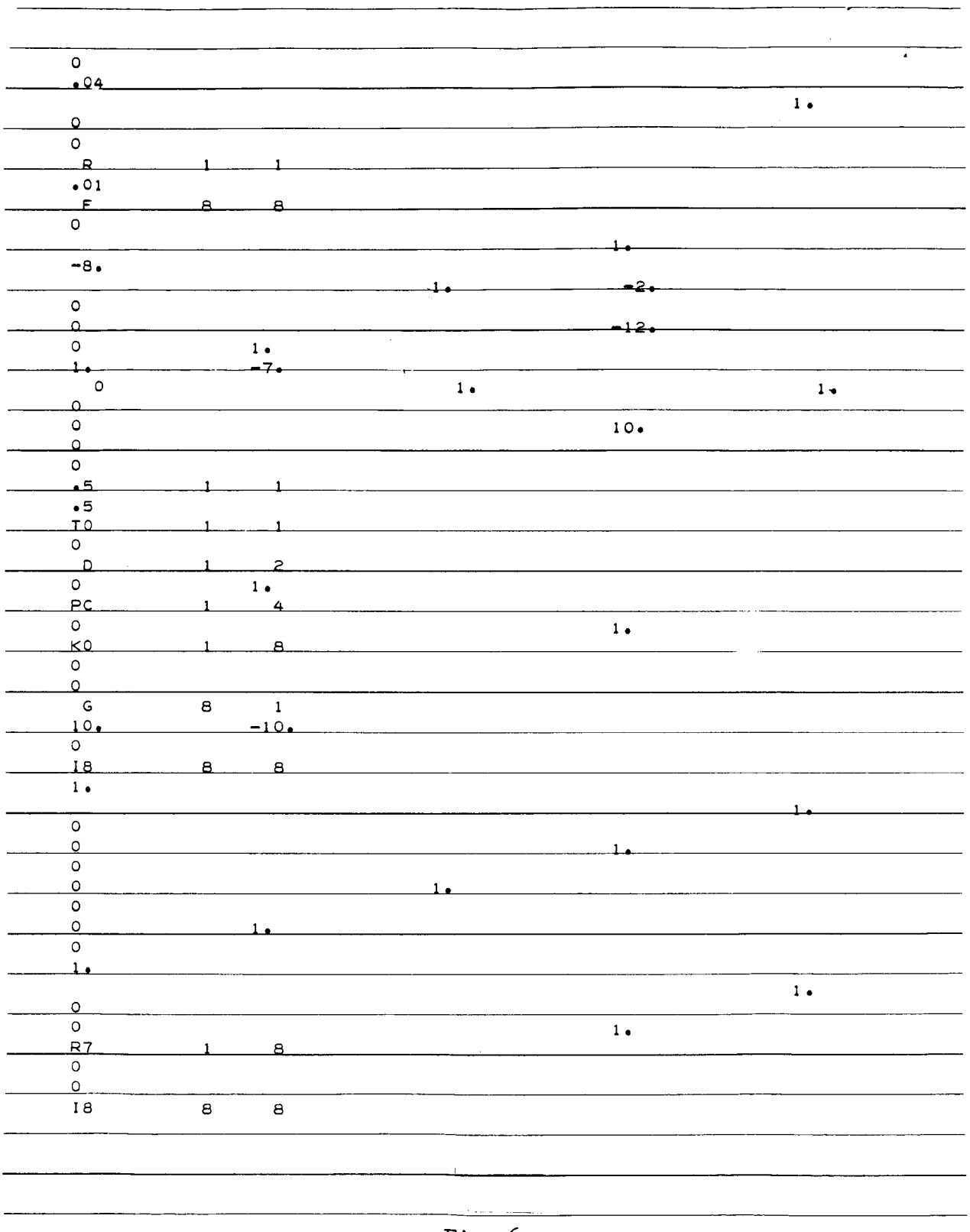
0
.04
1.
0
0
R 1 1
.01
F 8 8
0
1.
-8.
1. -2.
0
0 -12.
0 1.
1. -7.
0 1. 1.
0
0 10.
0
0
.5 1 1
.5
T0 1 1
0
D 1 2
0 1.
PC 1 4
0 1.
K0 1 8
0
0
G 8 1
10. -10.
0
I8 8 8
1.
1.
0
0 1.
0
0 1.
0
0
1.
1.
0
0 1.
R7 1 8
0
0
I8 8 8

Fig. 6

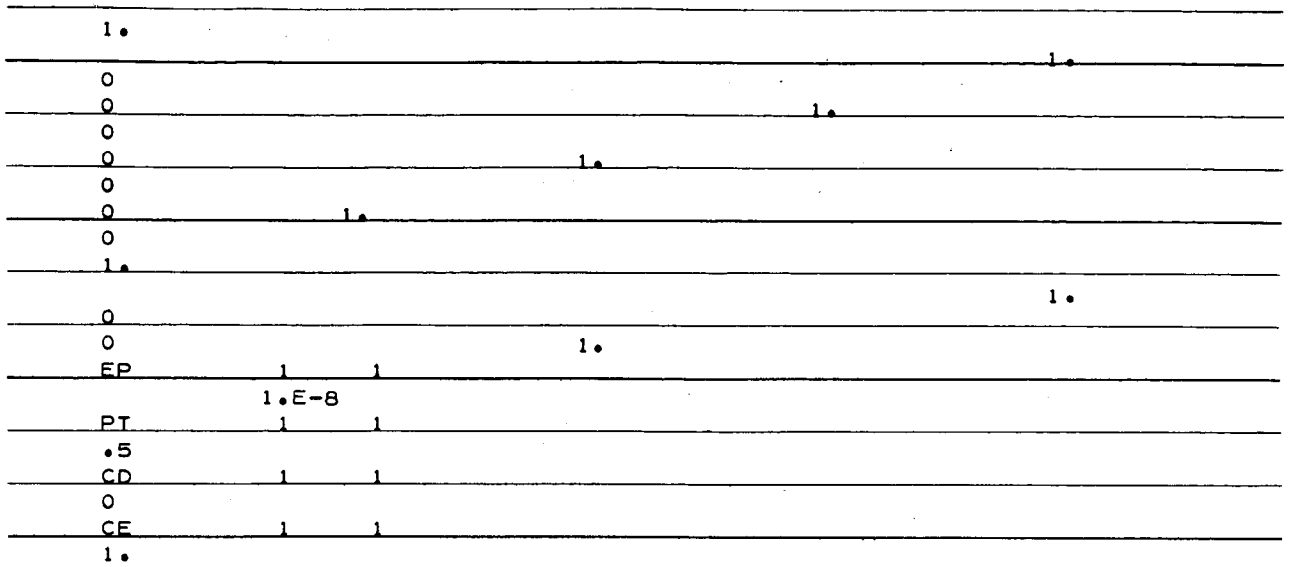

Fig. 6

```
        BEGIN
        LOAD    T, F, G, D,PC,R1,Q1,RZ,Z1,ND,X0 D2
HEAD 2MULT    T,R1,     R
        MULT    T,Q1,     Q
        EAT     F, T,PH,IN
        MULT   IN  G      GM
        MULT   D2 PH      PH
        MULT   D2 GM      GM
        TRANP  PH,    FT
        TRANP  GM     GT
        EQUAT   Q   P
        SAMPL  FT, P,GT, R, Q, D,PC,    PE,KT,AL
        TRANP  KT      K
        SUBT   PE  Q       P
        RINT    P PER K   K
        LOAD    T
        IF      T, T,HEAD 2
        END
 T              1       1
.73575888
 F              8       8
0
                                                          1.
-8.
                                        1.                -2.
0
0                                                        -12.
0                   1.
1.                  -7.
  0                                     1.                                1.
0
0                                                        10.
0
0
 G              8       1
10.                 -10.
0
 D              1       2
.0001               .18393972
PC              1       4
10.                 10.             1.                1000.
 R              1       1
   .01
 Q              8       8

0
0
0

0
0
0
0
```

Fig. 7

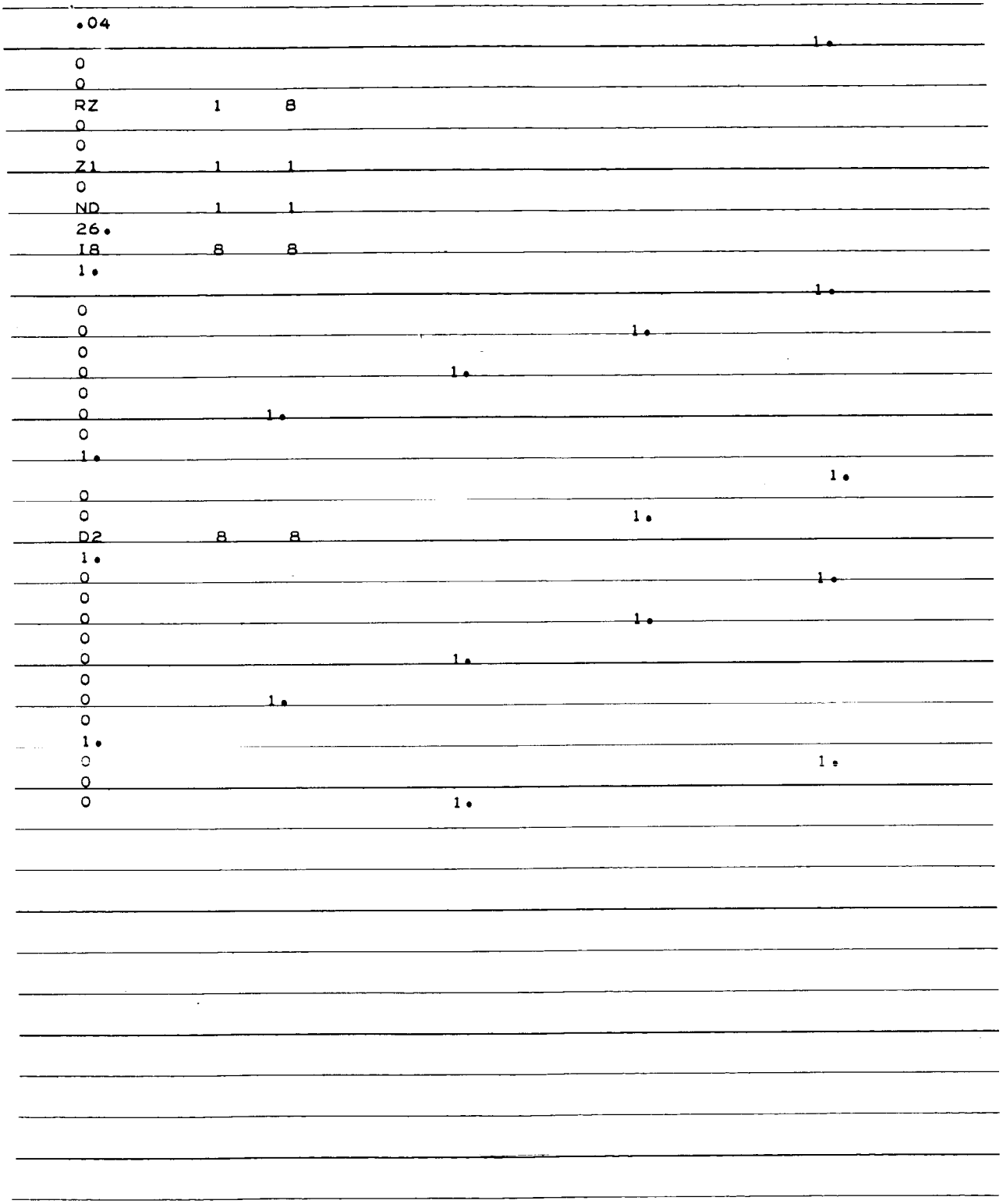
.04
0
0
RZ 1 8
0
0
Z1 1 1
0
ND 1 1
26.
I8 8 8
1.
0
0
0
0
0
0
0
1.
0
0
D2 8 8
1.
0
0
0
0
0
0
0
1.
0
0
0
1.
1.
1.
1.
1.
1.
1.
1.
1.
1.
1.
1.
1.
1.

Fig. 7

```
          BEGIN
          LOAD    T, F, G, D,PC,R1,Q1,RZ,Z1,ND,XO D2
HEAD 2MULT    T,R1,      R
          MULT    T,Q1,      Q
          EAT     F, T,PH,IN
          MULT  IN  G      GM
          MULT  D2 PH      PH
          MULT  D2 GM      GM
          TRANP PH,    FT
          TRANP GM     GT
          EQUAT  Q   P
          SAMPL FT, P,GT, R, Q, D,PC,   PE,KT,AL
          TRANP KT       K
          SUBT  PE   Q       P
          RINT    P PER K   K
          LOAD    T
          IF      T, T,HEAD 2
          END
 T              1       1
.73575888
 F              8       8
0
                                                          1.
-8.
                                          1.              -2.
0
0                                                         -12.
0               1.
1.              -7.
  0                             1.                                  1.
0
0                                                         10.
0
0
 G              8       1
10.             -10.
0
 D              1       2
.0001           .18393972
PC              1       4
10.             10.             1.              1000.
 R              1       1
  .01
 Q              8       8

0
0
0

0
0
0
0
```

Fig. 8

```
.04                                                          1.
0
0
RZ          1       8
0
0
Z1          1       1
0
ND          1       1
26.
I8          8       8
1.
                                                             1.
0
0                                                    1.
0
0                                  1.
0
0                       1.
0
1.                                                                1.
0
0                                                    1.
D2          8       8
1.
0                                                            1.
0
0                                                    1.
0
0                                  1.
0
0                       1.
0
1.                                                           1.
0
0
0                                                    1.
```

Fig. 8

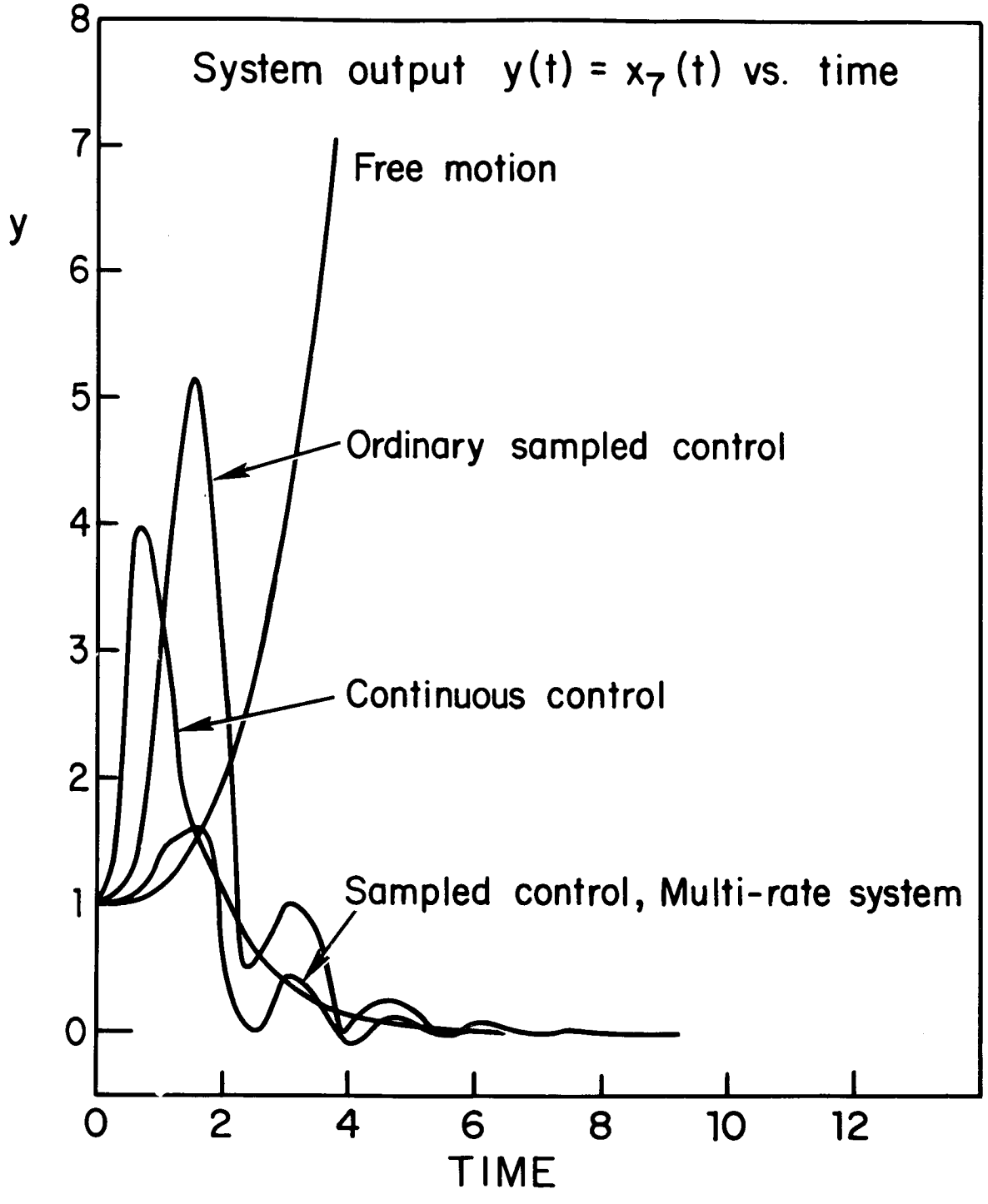
System output y(t) = x7 (t) vs. time
Free motion
Ordinary sampled control
Continuous control
Sampled control, Multi-rate system
y
TIME
0
1
2
3
4
5
6
7
8
0
2
4
6
8
10
12

Fig. 9

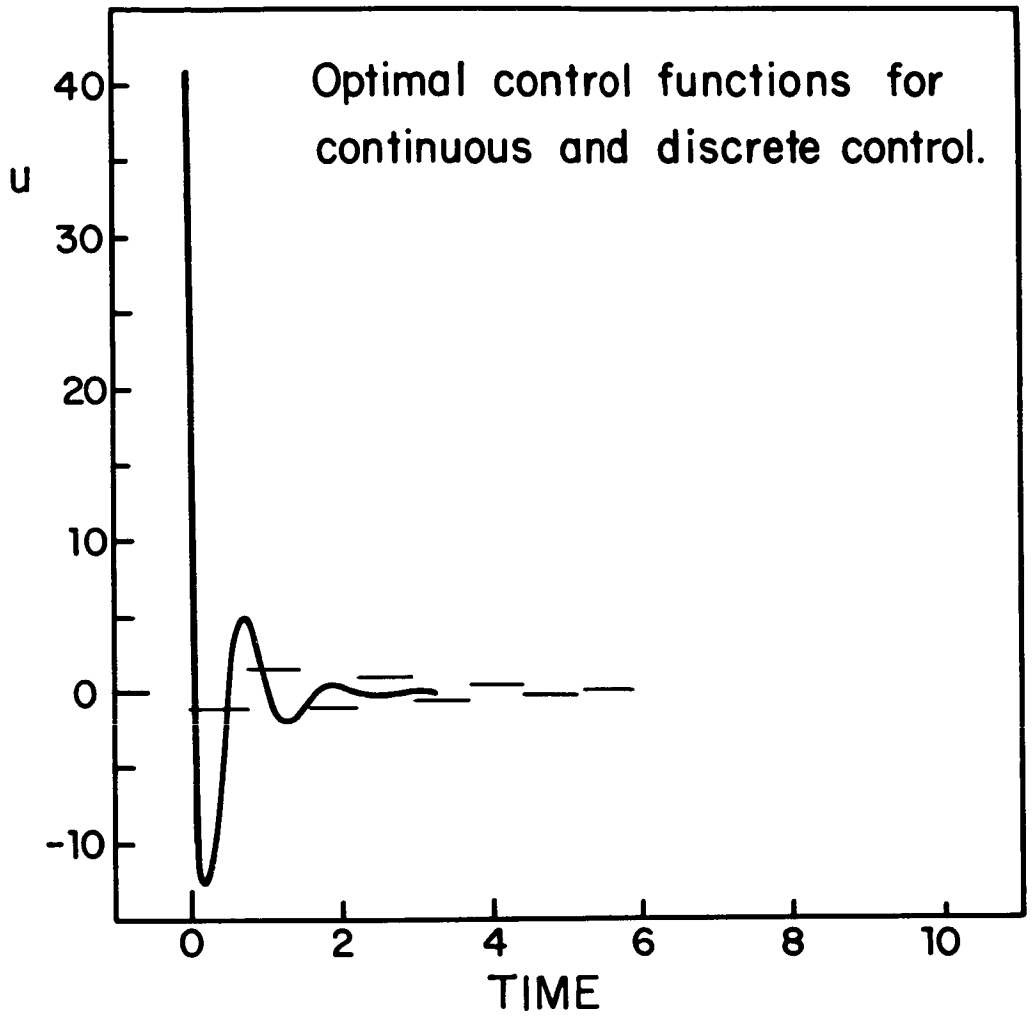
Optimal control functions for
continuous and discrete control.
u
40
30
20
10
0
-10
0
2
4
6
8
10
TIME

Fig. 10

CHAPTER XVIII

MODEL-FOLLOWER CONTROL

1. Description of the Problem: Given a system

$$\begin{aligned} \dot{x} &= Fx + Gu \\ y &= Hx \end{aligned} \tag{1.1}$$

and another, homogeneous system

$$z = Lz \tag{1.2}$$

we wish to obtain a control $u(t)$ which minimizes the performance index

$$\int_0^T [\|y - z\|_Q^2 + \|u\|_R^2]dt.$$

2. Theory and References: Our method will be to augment the system matrix by L, then $\|y - z\|_Q^2$ can be represented as $\|\hat{x}\|_{\hat{Q}}^2$ where

$$\hat{x} = \begin{bmatrix} x \\ z \end{bmatrix}, \quad \dot{\hat{x}} = \begin{bmatrix} F & 0 \\ 0 & L \end{bmatrix}\hat{x} + \begin{bmatrix} G \\ 0 \end{bmatrix}u$$

$$\hat{Q} = \begin{bmatrix} H'QH & -H'Q \\ -QH & Q \end{bmatrix}.$$

Notice that this augmented system is not completely controllable and we may expect problems with the behavior of the performance index because of this. Fortunately, in this application L is a stable matrix which will mitigate the problem.

3A. The Specific Problem: This problem is taken from the paper "Synthesis of Feedback Controls Using Optimization Theory" by F. J. Ellert and C. M. Merriam III.

Examination of the aircraft system leads us to

$$F = \begin{bmatrix} -.6 & -.76 & .00296875 & 0 \\ 1. & 0 & 0 & 0 \\ 0 & 102.4 & -.4 & 0 \\ 0 & 6 & 1 & 0 \end{bmatrix} \qquad G = \begin{bmatrix} -2.375 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\dot{x} = Fx + Gu$.

In addition we have a time-varying model to follow which necessitates enlarging the system by the direct sum with

$$\begin{bmatrix} \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} -.2 & 0 \\ 0 & -.2 \end{bmatrix} \begin{bmatrix} x_5 \\ x_6 \end{bmatrix} \qquad \begin{array}{l} 0 \leq t \leq 15 \\ x_5(0) = 100 \\ x_6(0) = -20 \end{array}$$

and

$$\begin{bmatrix} \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_5 \\ x_6 \end{bmatrix}. \qquad 15 \leq t \leq 20$$

The requirement that a second order system be used after $t = 15$ is the only reason for using one prior to that time. It may appear that this does not satisfy the altitude requirements as specified in the paper. However a small amount of analysis shows that the differences are less than one percent since

$$100\, e^{-.3} = 4.9787071$$

and

$$-20\, e^{-.3} = -.99574141$$

compared with the stated desiderata of 5 and -1. Furthermore the system as we simulate it does have $x_5(20) = 0$.

The error function may be written as

$$L = \varphi_4(x_5 - x_4)^2 + \varphi_3(x_3 - x_6)^2 + \varphi x_1^2 + u^2 = \frac{1}{2}[\|x\|_Q^2 + u^2]$$

with a terminal weighting of

$$\frac{1}{2}\|x\|_S^2 = \frac{1}{2}(\varphi_{4T}(x_5 - x_4)^2 + \varphi_{3T}(x_r - x_6)^2 + \varphi_{2T}(x_2 - 2^o)^2)$$

where 2^o can be written as $\frac{2}{57.2957795}$ radians and hence we may write the term as

$$\varphi_{2T}(x_2 + .034906586\, x_6)^2.$$

Using x_6 to specify the 2^o terminal value of x_2 is of course purely coincidental, we can do this because $x_6(20)$ is a known constant.

Considering Case III in the paper, we define

$$Q = \begin{bmatrix} 99. & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .0001 & 0 & 0 & -.0001 \\ 0 & 0 & 0 & .00005 & -.00005 & 0 \\ 0 & 0 & 0 & -.00005 & .00005 & 0 \\ 0 & 0 & -.0001 & 0 & 0 & .0001 \end{bmatrix}$$

for $15 \leq t \leq 20$,

$$Q = \begin{bmatrix} 99 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .00005 & -.00005 & 0 \\ 0 & 0 & 0 & -.00005 & .00005 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

for $0 \leq t \leq 15$,

and

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10. & 0 & 0 & 0 & .34906586 \\ 0 & 0 & .01 & 0 & 0 & -.01 \\ 0 & 0 & 0 & .001 & -.001 & 0 \\ 0 & 0 & 0 & -.001 & .001 & 0 \\ 0 & .34906586 & -.01 & 0 & 0 & .022184688 \end{bmatrix}$$

4A. Results: Running this on the program appearing in Fig. 1, we achieved results agreeing with those appearing in the paper. For instance

$$k_{11}(0) = 4.18 \qquad k_{11}(16) = 4.22$$

$$k_{12}(0) = 2.20 \qquad k_{12}(16) = 3.25$$

$$k_{13}(0) = .00675 \qquad k_{13}(16) = .0136$$

$$k_{14}(0) = .00298 \qquad k_{14}(16) = .00642$$

The graph of altitude for the high, nominal, and low initial condition is in Fig. 2. As would be expected with a relatively inaccurate integration tool such as ASP, we do not get the same pitch angles at termination as the paper. However, the errors are less than $.003^{\circ}$.

I.C.	$\lvert u\rvert_{max}$ deg.	$\theta(20.)$ deg.	h(20) ft.	$\dot{h}(20)$ ft/sec.	$\lvert\alpha\rvert_{max}$ deg.
1	.4	- .033	- .103	- .76	1.0
2	16.3	.003	- .025	- .82	1.8
3	32.8	.038	.053	- .87	3.

Having obtained these results, we attempted to improve upon them. There were two points at which improvement was possible. One was at the beginning of the approach where the initial (.5 sec) transient gave extremely large control values compared with the remainder of the trajectory. The other was at the terminal time, when it appeared that state variables could be closer to the desired values. We felt that these desiderata could be accomplished because of the extremely low magnitude of control used everywhere except initially.

We mention here one confusing point in the article. Initial angle of attack, α, is supposed to be 14°, yet in each of the three initial conditions given, $\theta = \gamma$, hence $\alpha = 0$.

In fact the condition that

$$\ddot{h}(0) = \frac{V}{T_s}\left(\theta(0) - \frac{\dot{h}(0)}{V}\right)$$

be zero requires that

$$\alpha(0) = \theta(0) - \gamma(0) = \theta(0) - \frac{\dot{h}(0)}{V} = 0.$$

Under these circumstances we deleted any consideration of α, though it will be found that if $\alpha(.)$ is satisfactorily bounded in the article, it is also satisfactory in our runs.

To attack these problems, we needed a more flexible program; one capable of modifying the index weighting and the control gains. In addition we wanted more printout at the critical initial and terminal times. Using some hindsight we reran the problem with the program in Fig. 3, to check agreement with results using Fig. 1.

We then increased S (terminal weighting) by a factor of 100 by removing the two M U L T 0 T, S, S instructions, to improve terminal conditions. This affected the initial transient very little but gave the following terminal values.

I.C.	$\lvert u\rvert_{max}$	$\theta(20)$	h(20)	$\dot{h}(20)$	$\lvert\alpha\rvert_{max}$
1	.4	1.72	- .007	- .93	1.1
2	16.1	1.73	- .005	- .93	1.8
3	32.5	1.74	- .003	- .94	3.

This now looks very satisfactory except for the large control excursions. Notice that with increased weighting terminal conditions appear independent of initial conditions.

We then attempted to reduce the control requirement by reducing Q in the early part of the run, using the program in Fig. 4. This reduced Q by a factor of 10^{-3} on [1, 2] and by an additional 10^{-3} on [0, 1], and gave the following results:

I.C.	$\lvert u\rvert_{max}$	$\theta(20)$	$h(20)$	$\dot{h}(20)$	$\lvert\alpha\rvert_{max}$
1	- 13.0	1.73	- .006	- .93	- 4.8
2	3.3	1.73	- .007	- .93	2.5
3	19.1	1.73	- .007	- .93	6.2

Since we have lowered Q by a factor of 10^{-6} and still have a fairly significant gain matrix, we infer that F has at least one unstable eigenvalue and therefore K cannot be indefinitely reduced by reducing Q. Despite the fact that u is within the bounds, $-35^{\circ} \leq u \leq 15$, prescribed, we would like to show what can be done in such a case if the bounds were, e.g. $-8^{\circ} \leq u \leq 8^{\circ}$.

An optimal system cannot be made to satisfy because K cannot be sufficiently reduced, but an ad hoc procedure of reducing u to the bound may very well provide the answer. This is particularly true in a case such as this where most of the control magnitude is used in the initial surge. To approximate the effect of control stops we used the program in Fig. 5, multiplying K by $\lambda(t)$ ($\lambda(0) = .37$, $\lambda(.7) = 1$) before computing control. This gave the following results:

I.C.	$\lvert u\rvert_{max}$	$\theta(20)$	$h(20)$	$\dot{h}(20)$	$\lvert\alpha\rvert_{max}$
1	- 4.6	1.73	- .006	- .933	- 2.2
2	2.8	1.73	- .007	- .933	2.6
3	7.8	1.74	- .007	- .934	5.4

The altitude profiles for this run appear in Fig.6, in Fig.7 and 8 we give plots of k_3 and k_4 as we computed them from Fig. 1 and for the final run (K modified) as computed from Fig. 5.

3B. The Specific Problem: This is an aircraft problem received in a private communication from J. S. Tyler, Jr. It is very similar to the one described in J. S. Tyler, Jr., "The Characteristics of Model-Following Systems as Synthesized by Optimal Control," IEEE Trans. on Automatic Control, vol. AC-9, No. 4, October, 1964.

The plant is specified by

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -2.93 & -4.75 & .78 \\ .086 & 0 & -.11 & -1. \\ 0 & -.049 & 2.59 & -.39 \end{bmatrix}$$

$$G = \begin{bmatrix} 0 & 0 \\ 0 & -3.91 \\ .035 & 0 \\ -2.53 & .31 \end{bmatrix},$$

and the model by

$$L = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1. & -73.14 & 3.18 \\ .086 & 0 & -.11 & -1. \\ .0086 & .086 & 8.95 & -.49 \end{bmatrix}$$

We define the performance index by

$$\int_0^\infty [\|y\|_a^2 + \|u\|^2]dt$$

where $y = Hx$,

$H = [I_4, -I_4]$ and

$$Q = 5\, I_4$$

4.B Results: Let K_{fb}, K_{ff} denote respectively the gain matrices from the plant and from the model. Then Tyler gives the values

$$K_{fb} = \begin{bmatrix} .253 & -.185 & 1.58 & -2.34 \\ -2.21 & -1.83 & .7 & .01 \end{bmatrix}$$

$$K_{ff} = \begin{bmatrix} .104 & .377 & -3.63 & 4.16 \\ 2.035 & 2.211 & -15.3 & 2.59 \end{bmatrix}$$

Using the program appearing in Fig. 9, we calculate the matrices

$$K_{fb} = \begin{bmatrix} -.253 & -.185 & 1.584 & -2.342 \\ -2.214 & -1.827 & .700 & .010 \end{bmatrix}$$

$$K_{ff} = \begin{bmatrix} .104 & .377 & -3.630 & 4.161 \\ 2.035 & 2.211 & -15.309 & 2.589 \end{bmatrix}$$

```
      JUNE 28, 1965
      BEGIN
      REW          5
      REW          6
      WEF          5
      LOAD    F, G,T1,T2,F1,G1,T3,20,DT,TS, Q,Q1,TQ,DS,
      REW          5
      WEF          6
      SAVE         5T1, T, F, F, G, G, Q, Q,T2, T,F1, F,G1, G,Q1, Q,T3, T,
      REW          6
HEAD21SUBT   20,DT,    20
      SAVE         620, T,
      IF     20,TS,HEAD21
HEAD22SUBT   20,DS,    20
      SAVE         620, T,
      IF     20,TQ HEAD22
      LOAD   ZO,.5,ON,D2,D3,PC,RZ,ZR,TW,TL,RI, S,XX,TR,XO,TI  R  2
      WRITE          OPTIMAL FEEDBACK CONTROL
      REW          5
      REW          6
      REW          7
      WEF          7
      EQUAT TL, T,TL,TF,
      TRANP TI,    IT,
      BRING        5 T,T1, F, F, G, G, Q, Q, T,T2,
      MULT   IT, G,    32,
      MULT   32,TI,    G,
      MULT    2, S,    S
      MULT   IT  S     32
      MULT   32 TI     S
      BSR      1  3
      MULT   TR, F,    32,
      MULT   32,TI     F,
      MULT   TR, G     G,
      REW          7
      TRANP  G,    GT
      MULT   RI,GT,    C,
      MULT    C, S,    K,
      MULT    G, K,    GK,
      SUBT    F,GK,    CF,
      SAVE         7CF,CF, K, K, T, T,
      BRING        6 T,PI,
      SUBT   ZR, F,    MF,
      TRANP  G,    GT,
      MULT   RI,GT,    C,
      MULT    G, C,    GC,
      JUXTC MF,GC,     TP,
      TRANP  F,    FT,
      JUXTC  Q,FT,     BT,
      JUXTR TP,BT,     PH,
      ADD    T1 T2     T1,
      MULT   .5,T1     T2,
      EQUAT PI, T,ON,CD,          A PRINT TIME MUST COME BETWEEN TL
```

Fig. 1

```
        SUBT   TF, T,    TU,              AND THE LAST BREAK TIME.
        EQUAT   T,TF,
        ETPHI PH TU,P1,HT,
        MULT   TU D3     34
        MULT   HT,D2,    35,
        ADD    34 35     34
        RICAT   S P1, C,34,PC,XX, S, K,AL,
        RINT    S, P
        IF     ZO,ZO,HEAD 8
HEAD 5BRING        5 T,T1, F, F  G  G  Q  Q  T T2
        RINT   T1, T
        MULT   IT, Q,    32,
        MULT   32 TI      Q,
        BSR      1  5
        MULT   TR, F,    32
        MULT   32 TI      F
        MULT   TR  G      G
        MULT    G  K     GK
        SUBT    F GK     CF
        SAVE         7CF,CF, K, K, T, T,
HEAD 1SUBT   ZR, F,    MF,
        TRANP   G     GT,
        MULT   RI,GT,     C,
        MULT    G, C,    GC,
        JUXTC MF,GC,    TP
        TRANP   F,    FT,
        JUXTC   Q,FT,    BT,
        JUXTR TP,BT,    PH,
        ADD    T1,T2,    T1,
        MULT   .5,T1,    T2
HEAD 6IF     PI,T2,HEAD 2
        EQUAT T2,  T,ZO,CD,
        IF     ZO,ZO,HEAD 3
HEAD 2EQUAT PI,  T,ON,CD
HEAD 3SUBT   TF, T,    TU,
        EQUAT   T,TF,
        ETPHI PH,TU,P1,HT,
        MULT   TU,D3,    34
        MULT   HT D2     35
        ADD    34 35     34
        RICAT   S P1, C,34,PC,     S, K,AL,
HEAD 8MULT    G, K,    GK,
        SUBT    F,GK,    CF,
        SAVE         7CF,CF, K, K, T, T,
        IF     TW, T,HEAD 4
HEAD10IF     .5,CD,HEAD 5
        BRING        6 T,PI,
        IF     ZO,ZO,HEAD 6
HEAD 4BSR      4  7
        BLOT   IT
        BRING        7 T,T1,CF,CF, K, K, T,T2,
        BSR      7  7
        ADD    T1,T2,    T3,
```

Fig. 1

```
        MULT   .5 T3     HT,
        SUBT   HT T2     TU
        MULT   TR XO     XO
        EQUAT  HT  T
        ETPHI  CF,TU,PH,
        MULT    K,XO,    -U,
        MULT    K,TR,    KR,
        MULT   TI,XO,     X,
        TRANP  TR,    ZK,
        MULT   ZK, S,    41,
        MULT   41 TR      S
        RINT   T2,     S, P KR, K  X,XO -U,-U
        WRITE      THE PRECEDING WERE P, K, X, AND -U AT INITIAL TIME.
        MULT   PH,XO,     1,
        EQUAT  1,XO,ZO,CD,
HEAD 7BRING       7 T,T1,CF,CF, K, K, T,T2,
HEAD24BSR      7  7
        IF     T2,T1,HEAD 9
        SUBT   T2, T,    TU,
        ETPHI  CF TU,PH,
        MULT   PH,XO,    XO,
        EQUAT  T2, T,
        MULT   TI,XO,     X,
        MULT    K,XO,    -U,
        MULT    K,TR,    KR,
        JUXTR  KR,RZ,    *4,
        JUXTR  *4, X,    *5
        JUXTR  *5 RZ     *6
        JUXTR  *6 -U     *7
        RINT    T,    *7,INF
        ADD    T1,T2     T3
        MULT   .5,T3     HT
        SUBT   HT, T     TU
        ETPHI  CF,TU,PH,
        MULT   PH,XO     XO,
        EQUAT  HT, T,
        IF     T1,TL,HEAD12HEAD 7
HEAD 9SUBT     T1, T,    TU,
        ETPHI  CF,TU,PH,
        MULT   PH,XO,    XO,
        EQUAT  T1, T,
        BRING       7 T,T1,CF,CF, K, K, T,T2,
HEAD23BSR      7  7
        ADD    T1,T2,    T3,
        MULT   .5,T3,    HT
        SUBT   HT, T,    TU,
        ETPHI  CF,TU,PH,
        MULT   PH,XO,    XO,
        EQUAT  HT, T,
        IF     T1,TL,HEAD12HEAD 7
HEAD12BRING       7CF,CF, K, K, T,T2,
        SUBT   T2, T,    TU,
        ETPHI  CF,TU,PH,
```

Fig. 1

```
        MULT  PH,XO,   32,
        MULT  TI 32     X
        MULT   K,32    -U
        MULT   K,TR,   KR
        RINT  T2,   KR, K  X, X -U,-U
        WRITE    THE PRECEDING WERE K, X, AND -U AT FINAL TIME.
        END
 F          6      6
            -.6          -.76  .00296875                0          0
              0            1.            0          0          0
              0             0            0      102.4       -.4
              0             0            0          0          0
             1.             0            0          0          0
              0             0            0          0         1.
              0             0            0          0          0
              0
 G          6      1
   -2.375                   0            0          0          0
              0
 T1         1      1
          17.5
 T2         1      1
          12.5
 F1         6      6
            -.6          -.76  .00296875                0          0
              0            1.            0          0          0
              0             0            0      102.4       -.4
              0             0            0          0          0
             1.             0            0          0          0
              0             0            0       -.2           0
              0             0            0          0          0
            -.2
 G1         6      1
        -2.375               0            0          0          0
              0              0
 T3         1      1
          -15.
 20         1      1
            20.
 DT         1      1
   .25
 TS         1      1
        10.05
 Q          6      6
            99.              0            0          0          0
              0              0            0          0          0
              0              0            0          0      .0001
              0              0       -.0001          0          0
              0       .00005         -.00005          0          0
              0              0       -.00005     .00005          0
              0              0       -.0001           0          0
          .0001
 Q1         6      6
```

Fig. 1

```
                99.                     0               0               0               0
                  0                     0               0               0               0
                  0                     0               0               0               0
                  0                     0               0               0               0
                  0              .00005                -.00005          0               0
                  0                     0              -.00005          .00005          0
                  0                     0               0               0               0
                  0
TQ                1          1
      .05
DS                1          1
   .5
ZO                1          1
                  0
.5                1          1
                  .5
ON                1          1
                  1.
D2                1          3
                  0                     1.              0
D3                1          3
                  0                     0               1.
PC                1          4
                  0                     0               0               1.
RZ                1          6
                  0                     0               0               0               0
                  0
ZR                6          6
                  0                     0               0               0               0
                  0                     0               0               0               0
                  0                     0               0               0               0
                  0                     0               0               0               0
                  0                     0               0               0               0
                  0                     0               0               0               0
                  0                     0               0               0               0
                  0
TW                1          1
.0005
TL                1          1
                  20.
RI                1          1
1.
 S                6          6
                  0                     0               0               0               0
                  0                     0               5.              0               0
                  0        .17453293                    0               0               .005
                  0                     0              -.005            0               0
                  0              .0005                 -.0005           0               0
                  0                     0              -.0005           .0005           0
                  0        .17453293                   -.005            0               0
 .011092348
XX                6          6
                  1.                    0               0               0               0
```

Fig. 1

```
                0                0                1.               0                0
                0                0                0                0                1.
                0                0                0                0                0
                0                1.               0                0                0
                0                0                0                1.               0
                0                0                0                0                0
                1.
TR              6                6
                1.               0                0                0                0
                0                0                1.               0                0
                0                0                0                0                1.
                0                0                0                0                0
                0                1.               0                0                0
                0                0                0                1.               0
                0                0                0                0                0
                1.
XO              6                6
                0                0                0                0                0
                0          -.0625          -.0781          -.0938                    0
                0                0              -16.             -20.             -24.
                0                0                0             120.             100.
               80.               0                0                0             100.
              100.             100.               0                0                0
              -20.             -20.             -20.               0                0
                0
TI              6                6
                1.               0                0                0                0
                0                0                1.               0                0
                0                0                0                0                1.
                0                0                0                0                0
                0                1.               0                0                0
                0                0                0                1.               0
                0                0                0                0                0
                1.
R               1                1
              .25E4
2.              1                1
                2.
```

Fig. 1

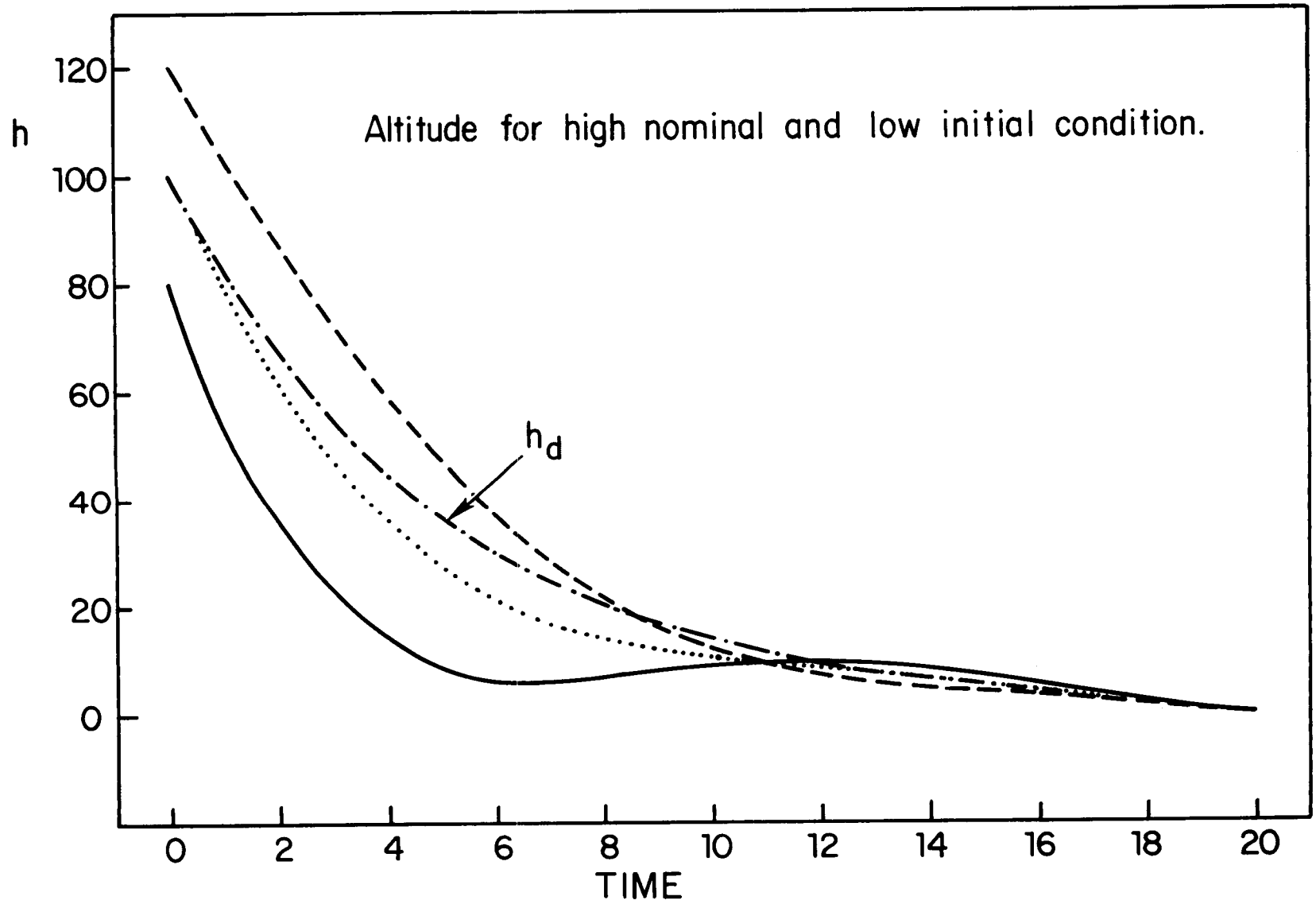
Altitude for high nominal and low initial condition.
h
120
100
80
60
40
20
0
h_d
0
2
4
6
8
10
12
14
16
18
20
TIME

Fig. 2

```
        BEGIN
        REW         5
        REW         6
        WEF         5
        LOAD    F, G,T1,T2,F1,G1,T3,20,DT,TS, Q,Q1,TQ,DS,ON
        REW         5
        WEF         6
        SAVE        5T1, T, F, F, G, G, Q, Q,T2, T,F1, F,G1, G,Q1, Q,T3, T,
        REW         6
        SUBT  20,TQ     20
        SAVE        620  T
        SUBT  20,TQ     20
        SAVE        620  T
        SUBT  20,TQ     20
        SAVE        620  T
        SUBT  20,TQ     20
        SAVE        620  T
        SUBT  20,TQ     20
        SAVE        620  T
HEAD21  SUBT  20,DT,    20
        SAVE        620, T,
        IF    20,TS,HEAD21
HEAD22  SUBT  20,DS,    20
        SAVE        620, T,
        IF    20,ON,HEAD22
        SUBT  20,TQ     20
        SAVE        620, T
        SUBT  20,TQ     20
        SAVE        620, T
        SUBT  20,TQ     20
        SAVE        620, T
        SUBT  20,TQ     20
        SAVE        620, T
        SUBT  20,TQ     20
        SAVE        620, T
        SUBT  20,TQ     20
        SAVE        620, T
        SUBT  20,TQ     20
        SAVE        620, T
        SUBT  20,TQ     20
        SAVE        620, T
        SUBT  20,TQ     20
        SAVE        620, T
        SUBT  20,TQ     20
        SAVE        620, T
        SUBT  20,TQ     20
        SAVE        620, T
        SUBT  20,TQ,    20,
        SAVE        620, T
        SUBT  20,TQ,    20,
        SAVE        620, T
        SUBT  20,TQ,    20,
        SAVE        620, T
        SUBT  20,TQ,    20,
        SAVE        620, T
```

Fig. 3

```
SUBT  20,TQ,   20,
SAVE       620, T
SUBT  20,TQ,   20,
SAVE       620, T
SUBT  20,TQ,   20,
SAVE       620, T
SUBT  20,TQ,   20,
SAVE       620, T
SUBT  20,TQ,   20,
SAVE       620, T
SUBT  20,TQ,   20,
SAVE       620, T
RINT  20,T
LOAD  ZO,,5,ON,D2,D3,PC,RZ,ZR,TW,TL,RI, S,XX,TR,XO,TI  R  2 TN OT
LOAD   H
WRITE        OPTIMAL FEEDBACK CONTROL
REW          5
REW          6
REW          7
WEF          7
EQUAT TL, T,TL,TF,
TRANP TI,   IT,
BRING       5 T,T1, F, F, G, G, Q, Q, T,T2,
MULT  IT, Q,   32,
MULT  32,TI,    Q,
MULT  OT  Q     Q
MULT  OT  Q     Q
MULT   2  S     S
MULT  OT  S     S
MULT  OT  S     S
MULT  IT  S    32
MULT  32 TI     S
BSR     1  5
MULT  TR, F,   32,
MULT  32,TI     F,
MULT  TR, G     G,
REW         7
TRANP  G,   GT
MULT  RI,GT,    C,
MULT   C, S,    K,
MULT   G, K,   GK,
SUBT   F,GK,   CF,
SAVE       7CF,CF, K, K, T, T,
BRING      6 T,PI,
SUBT  ZR, F,   MF,
TRANP  G,   GT,
MULT  RI,GT,    C,
MULT   G, C,   GC,
JUXTC MF,GC,   TP,
TRANP  F,   FT,
JUXTC  Q,FT,   BT,
JUXTR TP,BT,   PH,
ADD   T1 T2    T1,
```

Fig. 3

```
        MULT  .5,T1     T2,
        EQUAT PI, T,ON,CD,                A PRINT TIME MUST COME BETWEEN TL
        SUBT  TF, T,    TU,               AND THE LAST BREAK TIME.
        EQUAT  T,TF,
        ETPHI PH TU,P1,HT,
        MULT  TU D3     34
        MULT  HT,D2,    35,
        ADD   34 35     34
        RICAT  S P1, C,34,PC,XX, S, K,AL,
        RINT   S, P
        IF    ZO,ZO,HEAD 8
HEAD 5BRING       5 T,T1, F, F  G  G  Q  Q  T T2
        RINT  T1, T
        MULT  IT, Q,    32,
        MULT  32 TI      Q,
        MULT  OT  Q      Q
        MULT  OT  Q      Q
        BSR     1  5
        MULT  TR, F,    32
        MULT  32 TI      F
        MULT  TR  G      G
        MULT   G  K     GK
        SUBT   F GK     CF
        SAVE          7CF,CF, K, K, T, T,
HEAD 1SUBT  ZR, F,    MF,
        TRANP  G     GT,
        MULT  RI,GT,     C,
        MULT   G, C,    GC,
        JUXTC MF,GC,    TP
        TRANP  F,    FT,
        JUXTC  Q,FT,    BT,
        JUXTR TP,BT,    PH,
        ADD    T1,T2,   T1,
        MULT  .5,T1,    T2
HEAD 6IF      PI,T2,HEAD 2
        EQUAT T2, T,ZO,CD,
        IF    ZO,ZO,HEAD 3
HEAD 2EQUAT PI, T,ON,CD
HEAD 3SUBT  TF, T,    TU,
        EQUAT  T,TF,
        ETPHI PH,TU,P1,HT,
        MULT  TU,D3,    34
        MULT  HT D2     35
        ADD    34 35    34
        RICAT  S P1, C,34,PC,     S, K,AL,
HEAD 8MULT   G, K,     GK,
        SUBT   F,GK,    CF,
        SAVE          7CF,CF, K, K, T, T,
        IF    TW, T,HEAD 4
HEAD10IF     .5,CD,HEAD 5
        BRING       6 T,PI,
        IF    ZO,ZO,HEAD 6
HEAD 4MULT  ON  Q      Q
```

Fig. 3

```
        JUXTC  Q FT     BT
        JUXTR TP BT     PH
        LOAD   TW,
        IF      T,TW,HEAD10
        BSR      4  7
        BLOT   IT
        BRING       7 T,T1,CF,CF, K, K, T,T2,
        LOAD   LM TM
        MULT   LM, K,     K
        BSR      7  7
        ADD    T1,T2,    T3,
        MULT   .5 T3     HT,
        SUBT   HT T2     TU
        MULT   TR XO     XO
        EQUAT  HT  T
        MULT    G  K     GK
        SUBT    F GK     CF
        ETPHI  CF,TU,PH,
        MULT    K,XO,    -U,
        MULT    K,TR,    KR,
        MULT   TI,XO,     X,
        TRANP  TR,    ZK,
        MULT   ZK, S,    41,
        MULT   41 TR      S
        MULT    H  X     HX
        RINT   T2,     S, P KR, K  X,XO -U,-U HX AL
        WRITE       THE PRECEDING WERE P, K, X, AND -U AT INITIAL TIME.
        MULT   PH,XO,     1,
        EQUAT   1,XO,ZO,CD,
HEAD 7BRING       7 T,T1,CF,CF, K, K, T,T2,
        IF      T,TM,HEAD24
        LOAD   LM
        MULT   LM, K,     K
        MULT    G  K     GK
        SUBT    F GK     CF
HEAD24BSR      7  7
        IF     T2,T1,HEAD 9
        SUBT   T2, T,    TU,
        ETPHI  CF TU,PH,
        MULT   PH,XO,    XO,
        EQUAT  T2, T,
        MULT   TI,XO,     X,
        MULT    K,XO,    -U,
        MULT    K,TR,    KR,
        MULT    H  X     HX
        JUXTR KR,RZ,     *4,
        JUXTR *4, X,     *5
        JUXTR *5 RZ      Z5
        JUXTR Z5,HX      Z6
        JUXTR Z6 RZ      Z7
        JUXTR Z7 -U      *7
        RINT    T,    *7,INF
        ADD    T1,T2     T3
```

Fig. 3

```
        MULT  .5,T3     HT
        SUBT  HT, T     TU
        ETPHI CF,TU,PH,
        MULT  PH,X0     X0,
        EQUAT HT, T,
        IF    T1,TL,HEAD12HEAD 7
HEAD 9SUBT  T1, T,    TU,
        ETPHI CF,TU,PH,
        MULT  PH,X0,    X0,
        EQUAT T1, T,
        BRING       7 T,T1,CF,CF, K, K, T,T2,
        IF     T,TM,HEAD23
        LOAD  LM
        MULT  LM, K,     K
        MULT   G  K     GK
        SUBT   F GK     CF
HEAD23BSR     7  7
        ADD   T1,T2,    T3,
        MULT  .5,T3,    HT
        SUBT  HT, T,    TU,
        ETPHI CF,TU,PH,
        MULT  PH,X0,    X0,
        EQUAT HT, T,
        IF    T1,TL,HEAD12HEAD 7
HEAD12BRING       7CF,CF, K, K, T,T2,
        SUBT  T2, T,    TU,
        ETPHI CF,TU,PH,
        MULT  PH,X0,    32,
        MULT  TI 32      X
        MULT   K,32     -U
        MULT   K,TR,    KR
        MULT   H  X     HX
        RINT  T2,   KR, K  X, X -U,-U HX AL
        WRITE     THE PRECEDING WERE K, X, AND -U AT FINAL TIME.
        END
F           6     6
            -.6                -.76 .00296875                0          0
              0                1.              0             0          0
              0                0               0          102.4       -.4
              0                0               0             0          0
             1.                0               0             0          0
              0                0               0             0         1.
              0                0               0             0          0
              0
G           6     1
   -2.375                      0               0             0          0
              0
T1          1     1
          17.5
T2          1     1
          12.5
F1          6     6
            -.6                -.76   .00296875              0
```

Fig. 3

```
                0                1.              0               0               0
                0                0               0          102.4               -.4
                0                0               0               0               0
               1.                0               0               0               0
                0                0               0              -.2              0
                0                0               0               0               0
               -.2
G1              6       1
          -2.375                 0               0               0               0
                0                0
T3              1       1
          -15.
20              1       1
               20.
DT              1       1
   .25
TS              1       1
        10.05
 Q              6       6
99.
                0                0               0               0               0
                0                0               0               0           .0001
                0                0           -.0001              0               0
                0            .00005          -.00005             0               0
                0                0           -.00005         .00005              0
                0                0           -.0001              0               0
            .0001
Q1              6       6
           99.                   0               0               0               0
                0                0               0               0               0
                0                0               0               0               0
                0                0               0               0               0
                0            .00005          -.00005             0               0
                0                0           -.00005         .00005              0
                0                0               0               0               0
                0
TQ              1       1
          .05
DS              1       1
   .5
ON              1       1
1.025
ZO              1       1
                0
.5              1       1
               .5
ON              1       1
               1.
D2              1       3
                0                1.              0
D3              1       3
                0                0               1.
PC              1       4
```

Fig. 3

```
                0                   0                   0                   1.
RZ              1       6
                0                   0                   0                   0                   0
                0
ZR              6       6
                0                   0                   0                   0                   0
                0                   0                   0                   0                   0
                0                   0                   0                   0                   0
                0                   0                   0                   0                   0
                0                   0                   0                   0                   0
                0                   0                   0                   0                   0
                0                   0                   0                   0                   0
                0
TW              1       1
    2.005
TL              1       1
              20.
RI              1       1
  100.
 S              6       6
                0                   0                   0                   0                   0
                0                   0                   5.                  0                   0
                0          .17453293                    0                   0                .005
                0                   0          -.005                        0                   0
                0               .0005          -.0005                       0                   0
                0                   0          -.0005               .0005                       0
                0          .17453293               -.005                    0                   0
  .011092348
XX              6       6
               1.                   0                   0                   0                   0
                0                   0                  1.                   0                   0
                0                   0                   0                   0                  1.
                0                   0                   0                   0                   0
                0                  1.                   0                   0                   0
                0                   0                   0                  1.                   0
                0                   0                   0                   0                   0
               1.
TR              6       6
               1.                   0                   0                   0                   0
                0                   0                  1.                   0                   0
                0                   0                   0                   0                  1.
                0                   0                   0                   0                   0
                0                  1.                   0                   0                   0
                0                   0                   0                  1.                   0
                0                   0                   0                   0                   0
               1.
XO              6       6
                0                   0                   0                   0                   0
                0          -.0625               -.0781              -.0938                       0
                0                   0                -16.                -20.                -24.
                0                   0                   0                120.                100.
               80.                  0                   0                   0                100.
              100.                100.                  0                   0                   0
```

Fig. 3

```
                -20.          -20.          -20.          0             0
                  0
TI              6     6
                1.            0             0             0             0
                0             0             1.            0             0
                0             0             0             0             1.
                0             0             0             0             0
                0             1.            0             0             0
                0             0             0             1.            0
                0             0             0             0             0
                1.
R               1     1
              .25E4
2.              1     1
                2.
TN              1     1
10.
OT              1     1
.1
 H              1     6
0                     1.          -.00390625
0
TW              1     1
    1.005
TW              1     1
    .005
TW              1
    .005
LM              1     1
1.
TM              1     1
.8
LM              1     1
1.
LM              1     1
1.
LM              1     1
1.
LM              1     1
1.
LM              1     1
1.
LM              1     1
1.
LM              1     1
1.
LM              1     1
1.
LM              1     1
1.
LM              1     1
1.
LM              1     1
```

Fig. 3

1.		
LM	1	1
1.		
LM	1	1
1.		
LM	1	1
1.		
LM	1	1
1.		
LM	1	1
1.		

Fig. 3

```
          JUNE 28, 1965
          BEGIN
          REW          5
          REW          6
          WEF          5
          LOAD    F, G,T1,T2,F1,G1,T3,20,DT,TS, Q,Q1,TQ,DS,ON
          REW          5
          WEF          6
          SAVE         5T1, T, F, F, G, G, Q, Q,T2, T,F1, F,G1, G,Q1, Q,T3, T,
          REW          6
          SUBT   20,TQ      20
          SAVE         620   T
          SUBT   20,TQ      20
          SAVE         620   T
          SUBT   20,TQ      20
          SAVE         620   T
          SUBT   20,TQ      20
          SAVE         620   T
          SUBT   20,TQ      20
          SAVE         620   T
HEAD21    SUBT   20,DT,     20
          SAVE         620, T,
          IF     20,TS,HEAD21
HEAD22    SUBT   20,DS,     20
          SAVE         620, T,
          IF     20,ON,HEAD22
          SUBT   20,TQ      20
          SAVE         620, T
          SUBT   20,TQ      20
          SAVE         620, T
          SUBT   20,TQ      20
          SAVE         620, T
          SUBT   20,TQ      20
          SAVE         620, T
          SUBT   20,TQ      20
          SAVE         620, T
          SUBT   20,TQ      20
          SAVE         620, T
          SUBT   20,TQ      20
          SAVE         620, T
          SUBT   20,TQ      20
          SAVE         620, T
          SUBT   20,TQ      20
          SAVE         620, T
          SUBT   20,TQ      20
          SAVE         620, T
          SUBT   20,TQ,     20,
          SAVE         620, T
          SUBT   20,TQ,     20,
          SAVE         620, T
          SUBT   20,TQ,     20,
          SAVE         620, T
          SUBT   20,TQ,     20,
```

Fig. 4

```
SAVE        620, T
SUBT  20,TQ,    20,
SAVE        620, T
SUBT  20,TQ,    20,
SAVE        620, T
SUBT  20,TQ,    20,
SAVE        620, T
SUBT  20,TQ,    20,
SAVE        620, T
SUBT  20,TQ,    20,
SAVE        620, T
SUBT  20,TQ,    20,
SAVE        620, T
RINT  20,T
LOAD  ZO,,5,ON,D2,D3,PC,RZ,ZR,TW,TL,RI, S,XX,TR,XO,TI  R  2 TN OT
LOAD   H
WRITE        OPTIMAL FEEDBACK CONTROL
REW          5
REW          6
REW          7
WEF          7
EQUAT TL, T,TL,TF,
TRANP TI,    IT,
BRING        5 T,T1, F, F, G, G, Q, Q, T,T2,
MULT  IT, Q,    32,
MULT  32,TI,     Q,
MULT  OT  Q      Q
MULT  OT  Q      Q
MULT   2  S      S
MULT  IT  S     32
MULT  32 TI      S
BSR     1  5
MULT  TR, F,    32,
MULT  32,TI      F,
MULT  TR, G      G,
REW          7
TRANP  G,    GT
MULT  RI,GT,     C,
MULT   C, S,     K,
MULT   G, K,    GK,
SUBT   F,GK,    CF,
SAVE         7CF,CF, K, K, T, T,
BRING        6 T,PI,
SUBT  ZR, F,    MF,
TRANP  G,    GT,
MULT  RI,GT,     C,
MULT   G, C,    GC,
JUXTC MF,GC,    TP,
TRANP  F,    FT,
JUXTC  Q,FT,    BT,
JUXTR TP,BT,    PH,
ADD   T1 T2     T1,
MULT  .5,T1     T2,
```

Fig. 4

```
        EQUAT PI, T,ON,CD,              A PRINT TIME MUST COME BETWEEN TL
        SUBT  TF, T,   TU,              AND THE LAST BREAK TIME.
        EQUAT  T,TF,
        ETPHI PH TU,PI,HT,
        MULT  TU D3     34
        MULT  HT,D2,    35,
        ADD   34 35     34
        RICAT  S PI, C,34,PC,XX, S, K,AL,
        RINT   S, P
        IF    ZO,ZO,HEAD 8
HEAD 5BRING       5 T,T1, F, F  G  G  Q  Q  T T2
        RINT  T1, T
        MULT  IT, Q,    32,
        MULT  32 TI     Q,
        MULT  OT  Q     Q
        MULT  OT  Q     Q
        BSR     1  5
        MULT  TR, F,    32
        MULT  32 TI     F
        MULT  TR  G     G
        MULT   G  K     GK
        SUBT   F GK     CF
        SAVE          7CF,CF, K, K, T, T,
HEAD 1SUBT  ZR, F,    MF,
        TRANP  G      GT,
        MULT  RI,GT,    C,
        MULT   G, C,    GC,
        JUXTC MF,GC,    TP
        TRANP  F,     FT,
        JUXTC  Q,FT,    BT,
        JUXTR TP,BT,    PH,
        ADD   T1,T2,    T1,
        MULT  .5,T1,    T2
HEAD 6IF    PI,T2,HEAD 2
        EQUAT T2, T,ZO,CD,
        IF    ZO,ZO,HEAD 3
HEAD 2EQUAT PI, T,ON,CD
HEAD 3SUBT  TF, T,   TU,
        EQUAT  T,TF,
        ETPHI PH,TU,PI,HT,
        MULT  TU,D3,    34
        MULT  HT D2     35
        ADD   34 35     34
        RICAT  S PI, C,34,PC,    S, K,AL,
HEAD 8MULT   G, K,    GK,
        SUBT   F,GK,    CF,
        SAVE          7CF,CF, K, K, T, T,
        IF    TW, T,HEAD 4
HEAD10IF    .5,CD,HEAD 5
        BRING       6 T,PI,
        IF    ZO,ZO,HEAD 6
HEAD 4MULT  ON  Q     Q
        MULT  OT  Q     Q
```

Fig. 4

```
        MULT  OT  Q      Q
        MULT  OT  Q      Q
        JUXTC  Q FT      BT
        JUXTR TP BT      PH
        LOAD  TW,
        IF      T,TW,HEAD10
        BSR      4  7
        BLOT  IT
        BRING        7 T,T1,CF,CF, K, K, T,T2,
        LOAD  LM TM
        MULT  LM, K,     K
        BSR      7  7
        ADD   T1,T2,     T3,
        MULT  .5 T3      HT,
        SUBT  HT T2      TU
        MULT  TR XO      XO
        EQUAT HT  T
        MULT   G  K      GK
        SUBT   F GK      CF
        ETPHI CF,TU,PH,
        MULT   K,XO,     -U,
        MULT   K,TR,     KR,
        MULT  TI,XO,      X,
        TRANP TR,   ZK,
        MULT  ZK, S,     41,
        MULT  41 TR       S
        MULT   H  X      HX
        RINT  T2,    S, P KR, K  X,XO -U,-U HX AL
        WRITE       THE PRECEDING WERE P, K, X, AND -U AT INITIAL TIME,
        MULT  FH,XO,     1,
        EQUAT  1,XO,ZO,CD,
HEAD 7BRING        7 T,T1,CF,CF, K, K, T,T2,
        IF      T,TM,HEAD24
        LOAD  LM
        MULT  LM, K,     K
        MULT   G  K      GK
        SUBT   F GK      CF
HEAD24BSR       7  7
        IF     T2,T1,HEAD 9
        SUBT  T2, T,     TU,
        ETPHI  CF TU,PH,
        MULT  PH,XO,     XO,
        EQUAT T2, T,
        MULT  TI,XO,      X,
        MULT   K,XO,     -U,
        MULT   K,TR,     KR,
        MULT   H  X      HX
        JUXTR KR,RZ,     *4,
        JUXTR *4, X,     *5
        JUXTR *5 RZ      Z5
        JUXTR Z5,HX      Z6
        JUXTR Z6 RZ      Z7
        JUXTR Z7 -U      *7
```

Fig. 4

```
        RINT    T,    *7,INF
        ADD     T1,T2      T3
        MULT    .5,T3      HT
        SUBT    HT, T      TU
        ETPHI   CF,TU,PH,
        MULT    PH,XO      XO,
        EQUAT   HT, T,
        IF      T1,TL,HEAD12HEAD 7
HEAD 9SUBT      T1, T,     TU,
        ETPHI   CF,TU,PH,
        MULT    PH,XO,     XO,
        EQUAT   T1, T,
        BRING        7 T,T1,CF,CF, K, K, T,T2,
        IF       T,TM,HEAD23
        LOAD    LM
        MULT    LM, K,     K
        MULT     G  K      GK
        SUBT     F  GK     CF
HEAD23BSR        7  7
        ADD     T1,T2,     T3,
        MULT    .5,T3,     HT
        SUBT    HT, T,     TU,
        ETPHI   CF,TU,PH,
        MULT    PH,XO,     XO,
        EQUAT   HT, T,
        IF      T1,TL,HEAD12HEAD 7
HEAD12BRING         7CF,CF, K, K, T,T2,
        SUBT    T2, T,     TU,
        ETPHI   CF,TU,PH,
        MULT    PH,XO,     32,
        MULT    TI 32      X
        MULT     K,32      -U
        MULT     K,TR,     KR
        MULT     H  X      HX
        RINT    T2,    KR, K  X, X -U,-U HX AL
        WRITE     THE PRECEDING WERE K, X, AND -U AT FINAL TIME,
        END
  F             6      6
                -.6                -.76 .00296875                0          0
                  0                1.                 0          0          0
                  0                0                  0          102.4      -.4
                  0                0                  0          0          0
                 1.                0                  0          0          0
                  0                0                  0          0          1.
                  0                0                  0          0          0
                  0
   G            6      1
    -2.375                         0                  0          0          0
                  0
  T1            1      1
              17.5
  T2            1      1
              12.5
```

Fig. 4

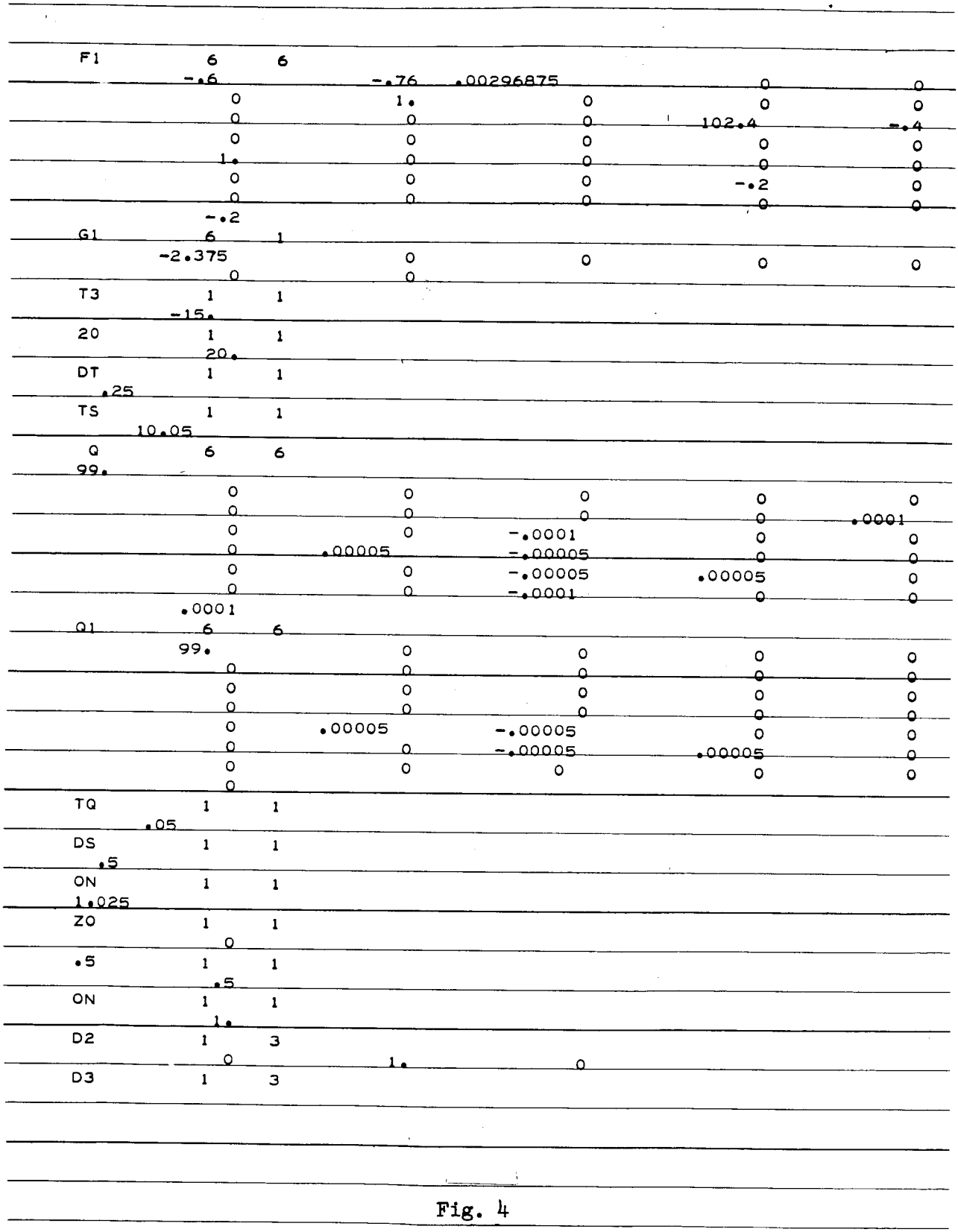
F1 6 6
-.6 -.76 .00296875 0 0
0 1. 0 0 0
0 0 0 102.4 -.4
0 0 0 0 0
1. 0 0 0 0
0 0 0 -.2 0
0 0 0 0 0
-.2
G1 6 1
-2.375 0 0 0 0
0 0
T3 1 1
-15.
20 1 1
20.
DT 1 1
.25
TS 1 1
10.05
Q 6 6
99. 0 0 0 0 0
0 0 0 0 .0001
0 0 -.0001 0 0
0 .00005 -.00005 0 0
0 0 -.00005 .00005 0
0 0 -.0001 0 0
.0001
Q1 6 6
99. 0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 .00005 -.00005 0 0
0 0 -.00005 .00005 0
0 0 0 0 0
0
TQ 1 1
.05
DS 1 1
.5
ON 1 1
1.025
ZO 1 1
0
.5 1 1
.5
ON 1 1
1.
D2 1 3
0 1. 0
D3 1 3

Fig. 4

```
                0                0                1.
PC          1       4
                0                0                0                1.
RZ          1       6
                0                0                0                0                0
                0
ZR          6       6
                0                0                0                0                0
                0                0                0                0                0
                0                0                0                0                0
                0                0                0                0                0
                0                0                0                0                0
                0                0                0                0                0
                0                0                0                0                0
                0
TW          1       1
    2.005
TL          1       1
           20.
RI          1       1
    100.
S           6       6
                0                0                0                0                0
                0                0                5.               0                0
                0   .17453293                     0                0           .005
                0                0          -.005                  0                0
                0        .0005               -.0005                0                0
                0                0          -.0005           .0005                  0
                0   .17453293                -.005                 0                0
 .011092348
XX          6       6
               1.                0                0                0                0
                0                0                1.               0                0
                0                0                0                0               1.
                0                0                0                0                0
                0               1.                0                0                0
                0                0                0               1.                0
                0                0                0                0                0
               1.
TR          6       6
               1.                0                0                0                0
                0                0               1.                0                0
                0                0                0                0               1.
                0                0                0                0                0
                0               1.                0                0                0
                0                0                0               1.                0
                0                0                0                0                0
               1.
XO          6       6
                0                0                0                0                0
                0          -.0625           -.0781           -.0938                0
                0                0              -16.             -20.             -24.
                0                0                0              120.             100.
```

Fig. 4

```
                80.          0          0          0        100.
               100.        100.          0          0          0
               -20.        -20.        -20.          0          0
                  0
TI              6   6
                1.  0          0          0          0          0
                 0  0          0         1.          0          0
                 0  0          0          0          0         1.
                 0  0          0          0          0          0
                 0 1.          0          0          0          0
                 0  0          0          0         1.          0
                 0  0          0          0          0          0
                1.
R               1   1
              .25E4
2.              1   1
                2.
TN              1   1
10.
OT              1   1
.1
 H              1   6
0                  1.      -.00390625
0
TW              1   1
    1.005
TW              1   1
    .005
TW              1   1
    .005
LM              1   1
1.
TM              1   1
.8
LM              1   1
1.
LM              1   1
1.
LM              1   1
1.
LM              1   1
1.
LM              1   1
1.
LM              1   1
1.
LM              1   1
1.
LM              1   1
1.
LM              1   1
1.
LM              1   1
```

Fig. 4

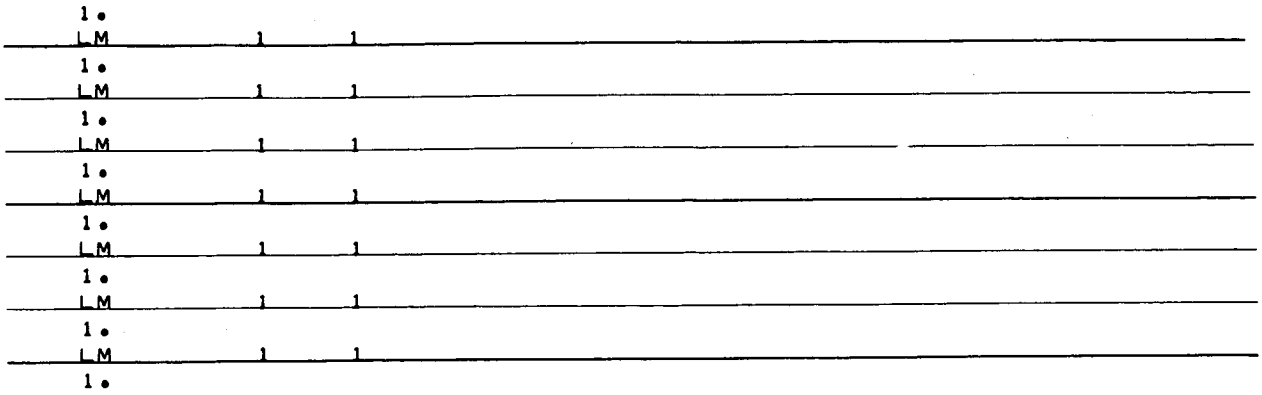

Fig. 4

```
          JUNE 28, 1965
          BEGIN
          REW         5
          REW         6
          WEF         5
          LOAD    F, G,T1,T2,F1,G1,T3,20,DT,TS, Q,Q1,TQ,DS,ON
          REW         5
          WEF         6
          SAVE        5T1, T, F, F, G, G, Q, Q,T2, T,F1, F,G1, G,Q1, Q,T3, T,
          REW         6
          SUBT   20,TQ     20
          SAVE        620   T
          SUBT   20,TQ     20
          SAVE        620   T
          SUBT   20,TQ     20
          SAVE        620   T
          SUBT   20,TQ     20
          SAVE        620   T
          SUBT   20,TQ     20
          SAVE        620   T
HEAD21    SUBT   20,DT,    20
          SAVE        620, T,
          IF     20,TS,HEAD21
HEAD22    SUBT   20,DS,    20
          SAVE        620, T,
          IF     20,ON,HEAD22
          SUBT   20,TQ     20
          SAVE        620, T
          SUBT   20,TQ     20
          SAVE        620, T
          SUBT   20,TQ     20
          SAVE        620, T
          SUBT   20,TQ     20
          SAVE        620, T
          SUBT   20,TQ     20
          SAVE        620, T
          SUBT   20,TQ     20
          SAVE        620, T
          SUBT   20,TQ     20
          SAVE        620, T
          SUBT   20,TQ     20
          SAVE        620, T
          SUBT   20,TQ     20
          SAVE        620, T
          SUBT   20,TQ     20
          SAVE        620, T
          SUBT   20,TQ,    20,
          SAVE        620, T
          SUBT   20,TQ,    20,
          SAVE        620, T
          SUBT   20,TQ,    20,
          SAVE        620, T
          SUBT   20,TQ,    20,
```

Fig. 5

```
SAVE        620, T
SUBT  20,TQ,   20,
SAVE        620, T
SUBT  20,TQ,   20,
SAVE        620, T
SUBT  20,TQ,   20,
SAVE        620, T
SUBT  20,TQ,   20,
SAVE        620, T
SUBT  20,TQ,   20,
SAVE        620, T
SUBT  20,TQ,   20,
SAVE        620, T
RINT  20,T
LOAD  ZO,,5,ON,D2,D3,PC,RZ,ZR,TW,TL,RI, S,XX,TR,XO,TI  R  2 TN OT
LOAD   H
WRITE       OPTIMAL FEEDBACK CONTROL
REW         5
REW         6
REW         7
WEF         7
EQUAT TL, T,TL,TF,
TRANP TI,   IT,
BRING       5 T,T1, F, F, G, G, Q, Q, T,T2,
MULT  IT, Q,   32,
MULT  32,TI,   Q,
MULT  OT  Q    Q
MULT  OT  Q    Q
MULT   2  S    S
MULT  IT  S    32
MULT  32 TI    S
BSR     1  5
MULT  TR, F,   32,
MULT  32,TI    F,
MULT  TR, G    G,
REW         7
TRANP  G,   GT
MULT  RI,GT,   C,
MULT   C, S,   K,
MULT   G, K,   GK,
SUBT   F,GK,   CF,
SAVE        7CF,CF, K, K, T, T,
BRING       6 T,PI,
SUBT  ZR, F,   MF,
TRANP  G,   GT,
MULT  RI,GT,   C,
MULT   G, C,   GC,
JUXTC MF,GC,   TP,
TRANP  F,   FT,
JUXTC  Q,FT,   BT,
JUXTR TP,BT,   PH,
ADD   T1 T2    T1,
MULT  .5,T1    T2,
```

Fig. 5

```
        EQUAT PI, T,ON,CD,              A PRINT TIME MUST COME BETWEEN TL
        SUBT  TF, T,   TU,              AND THE LAST BREAK TIME.
        EQUAT  T,TF,
        ETPHI PH TU,P1,HT,
        MULT  TU D3     34
        MULT  HT,D2,    35,
        ADD   34 35     34
        RICAT  S P1, C,34,PC,XX, S, K,AL,
        RINT   S, P
        IF    ZO,ZO,HEAD 8
HEAD 5BRING       5 T,T1, F, F  G  G  Q  Q  T T2
        RINT  T1, T
        MULT  IT, Q,    32,
        MULT  32 TI     Q,
        MULT  OT  Q     Q
        MULT  OT  Q     Q
        BSR     1  5
        MULT  TR, F,    32
        MULT  32 TI     F
        MULT  TR  G     G
        MULT   G  K     GK
        SUBT   F GK     CF
        SAVE        7CF,CF, K, K, T, T,
HEAD 1SUBT  ZR, F,  MF,
        TRANP  G    GT,
        MULT  RI,GT,    C,
        MULT   G, C,    GC,
        JUXTC MF,GC,    TP
        TRANP  F,   FT,
        JUXTC  Q,FT,    BT,
        JUXTR TP,BT,    PH,
        ADD   T1,T2,    T1,
        MULT  .5,T1,    T2
HEAD 6IF    PI,T2,HEAD 2
        EQUAT T2, T,ZO,CD,
        IF    ZO,ZO,HEAD 3
HEAD 2EQUAT PI, T,ON,CD
HEAD 3SUBT  TF, T,   TU,
        EQUAT  T,TF,
        ETPHI PH,TU,P1,HT,
        MULT  TU,D3,    34
        MULT  HT D2     35
        ADD   34 35     34
        RICAT  S P1, C,34,PC,    S, K,AL,
HEAD 8MULT   G, K,    GK,
        SUBT   F,GK,    CF,
        SAVE        7CF,CF, K, K, T, T,
        IF    TW, T,HEAD 4
HEAD10IF    .5,CD,HEAD 5
        BRING       6 T,PI,
        IF    ZO,ZO,HEAD 6
HEAD 4MULT  ON  Q     Q
        MULT  OT  Q     Q
```

Fig. 5

```
        MULT  OT  Q     Q
        MULT  OT  Q     Q
        JUXTC  Q FT     BT
        JUXTR TP BT     PH
        LOAD  TW,
        IF     T,TW,HEAD10
        BSR     4  7
        BLOT  IT
        BRING      7 T,T1,CF,CF, K, K, T,T2,
        LOAD  LM TM
        MULT  LM, K,    K
        BSR     7  7
        ADD   T1,T2,    T3,
        MULT  .5 T3     HT,
        SUBT  HT T2     TU
        MULT  TR XO     XO
        EQUAT HT  T
        MULT   G  K     GK
        SUBT   F GK     CF
        ETPHI CF,TU,PH,
        MULT   K,XO,    -U,
        MULT   K,TR,    KR,
        MULT  TI,XO,     X,
        TRANP TR,   ZK,
        MULT  ZK, S,    41,
        MULT  41 TR      S
        MULT   H  X     HX
        RINT  T2,    S, P KR, K  X,XO -U,-U HX AL
        WRITE      THE PRECEDING WERE P, K, X, AND -U AT INITIAL TIME.
        MULT  PH,XO,     1,
        EQUAT  1,XO,ZO,CD,
HEAD 7BRING       7 T,T1,CF,CF, K, K, T,T2,
        IF     T,TM,HEAD24
        LOAD  LM
        MULT  LM, K,    K
        MULT   G  K     GK
        SUBT   F GK     CF
HEAD24BSR      7  7
        IF    T2,T1,HEAD 9
        SUBT  T2, T,    TU,
        ETPHI CF TU,PH,
        MULT  PH,XO,    XO,
        EQUAT T2, T,
        MULT  TI,XO,     X,
        MULT   K,XO,    -U,
        MULT   K,TR,    KR,
        MULT   H  X     HX
        JUXTR KR,RZ,    *4,
        JUXTR *4, X,    *5
        JUXTR *5 RZ     Z5
        JUXTR Z5,HX     Z6
        JUXTR Z6 RZ     Z7
        JUXTR Z7 -U     *7
```

Fig. 5

```
        RINT    T,    *7,INF
        ADD     T1,T2      T3
        MULT    .5,T3      HT
        SUBT    HT, T      TU
        ETPHI   CF,TU,PH,
        MULT    PH,XO      XO,
        EQUAT   HT, T,
        IF      T1,TL,HEAD12HEAD 7
HEAD 9SUBT      T1, T,     TU,
        ETPHI   CF,TU,PH,
        MULT    PH,XO,     XO,
        EQUAT   T1, T,
        BRING        7 T,T1,CF,CF, K, K, T,T2,
        IF      T,TM,HEAD23
        LOAD    LM
        MULT    LM, K,     K
        MULT    G  K       GK
        SUBT    F  GK      CF
HEAD23BSR       7  7
        ADD     T1,T2,     T3,
        MULT    .5,T3,     HT
        SUBT    HT, T,     TU,
        ETPHI   CF,TU,PH,
        MULT    PH,XO,     XO,
        EQUAT   HT, T,
        IF      T1,TL,HEAD12HEAD 7
HEAD12BRING          7CF,CF, K, K, T,T2,
        SUBT    T2, T,     TU,
        ETPHI   CF,TU,PH,
        MULT    PH,XO,     32,
        MULT    TI 32      X
        MULT    K,32       -U
        MULT    K,TR,      KR
        MULT    H  X       HX
        RINT    T2,    KR, K  X, X -U,-U HX AL
        WRITE      THE PRECEDING WERE K, X, AND -U AT FINAL TIME.
        END
 F              6      6
                -.6                -.76 .00296875                0          0
                  0                1.                 0          0          0
                  0                0                  0          102.4      -.4
                  0                0                  0          0          0
                 1.                0                  0          0          0
                  0                0                  0          0          1.
                  0                0                  0          0          0
                  0
  G             6      1
   -2.375                          0                  0          0          0
                  0
 T1             1      1
          17.5
 T2             1      1
          12.5
```

Fig. 5

```
F1          6       6
          -.6                 -.76    .00296875                0           0
                0             1.            0                  0           0
                0             0             0                102.4         -.4
                0             0             0                  0           0
              1.              0             0                  0           0
                0             0             0                -.2           0
                0             0             0                  0           0
              -.2
G1          6       1
        -2.375                0             0                  0           0
                0             0
T3          1       1
          -15.
20          1       1
          20.
DT          1       1
    .25
TS          1       1
       10.05
 Q          6       6
99.
                0             0             0                  0           0
                0             0             0                  0         .0001
                0             0          -.0001                0           0
                0          .00005        -.00005               0           0
                0             0          -.00005            .00005         0
                0             0          -.0001                0           0
           .0001
Q1          6       6
          99.                 0             0                  0           0
                0             0             0                  0           0
                0             0             0                  0           0
                0             0             0                  0           0
                0          .00005        -.00005               0           0
                0             0          -.00005            .00005         0
                0             0             0                  0           0
                0
TQ          1       1
        .05
DS          1       1
    .5
ON          1       1
 1.025
ZO          1       1
                0
.5          1       1
              .5
ON          1       1
              1.
D2          1       3
                0             1.            0
D3          1       3
```

Fig. 5

```
                0                   0                   1.
PC      1       4
                0                   0                   0                   1.
RZ      1       6
                0                   0                   0                   0                   0
                0
ZR      6       6
                0                   0                   0                   0                   0
                0                   0                   0                   0                   0
                0                   0                   0                   0                   0
                0                   0                   0                   0                   0
                0                   0                   0                   0                   0
                0                   0                   0                   0                   0
                0                   0                   0                   0                   0
                0
TW      1       1
   2.005
TL      1       1
          20.
RI      1       1
   100.
S       6       6
                0                   0                   0                   0                   0
                0                   0                   5.                  0                   0
                0   .17453293                           0                   0                   .005
                0                   0       -.005                           0                   0
                0       .0005               -.0005                          0                   0
                0                   0       -.0005              .0005                           0
                0   .17453293               -.005               0                   0
  .011092348
XX      6       6
                1.                  0                   0                   0                   0
                0                   0                   1.                  0                   0
                0                   0                   0                   0                   1.
                0                   0                   0                   0                   0
                0                   1.                  0                   0                   0
                0                   0                   0                   1.                  0
                0                   0                   0                   0                   0
                1.
TR      6       6
                1.                  0                   0                   0                   0
                0                   0                   1.                  0                   0
                0                   0                   0                   0                   1.
                0                   0                   0                   0                   0
                0                   1.                  0                   0                   0
                0                   0                   0                   1.                  0
                0                   0                   0                   0                   0
                1.
XO      6       6
                0                   0                   0                   0                   0
                0       -.0625              -.0781              -.0938                          0
                0                   0           -16.                -20.            -24.
                0                   0                   0           120.            100.
```

Fig. 5

```
                80.          0          0          0        100.
                100.       100.         0          0          0
                -20.       -20.       -20.         0          0
                  0
TI              6     6
                1.           0          0          0          0
                 0           0         1.          0          0
                 0           0          0          0         1.
                 0           0          0          0          0
                 0          1.          0          0          0
                 0           0          0         1.          0
                 0           0          0          0          0
                1.
R               1     1
              .25E4
2.              1     1
                2.
TN              1     1
10.
OT              1     1
.1
H               1     6
0                    1.          -.00390625
0
TW              1     1
    1.005
TW              1     1
    .005
TW              1     1
    .005
LM              1     1
.37
TM              1     1
.8
LM              1     1
.37
LM              1     1
.37
LM              1     1
.42
LM              1     1
.45
LM              1     1
.48
LM              1     1
.51
LM              1     1
.55
LM              1     1
.6
LM              1     1
.65
LM              1     1
```

Fig. 5

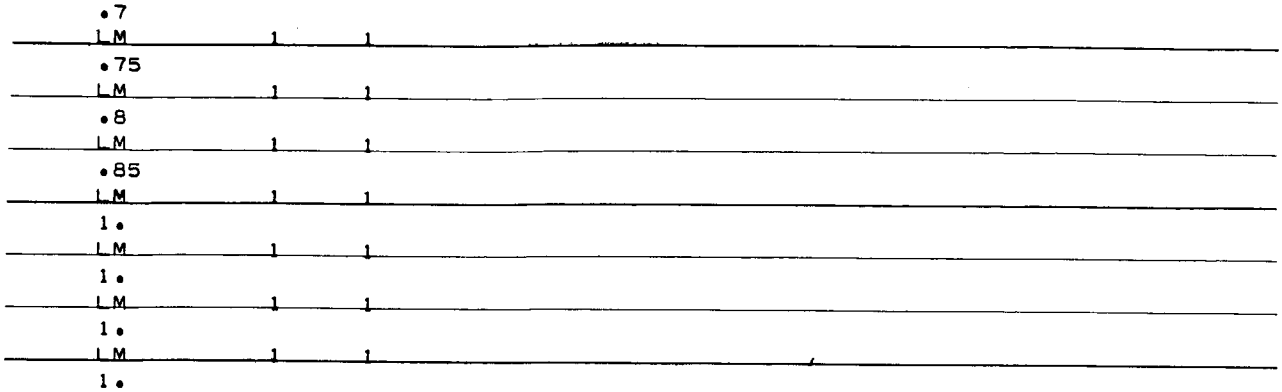

Fig. 5

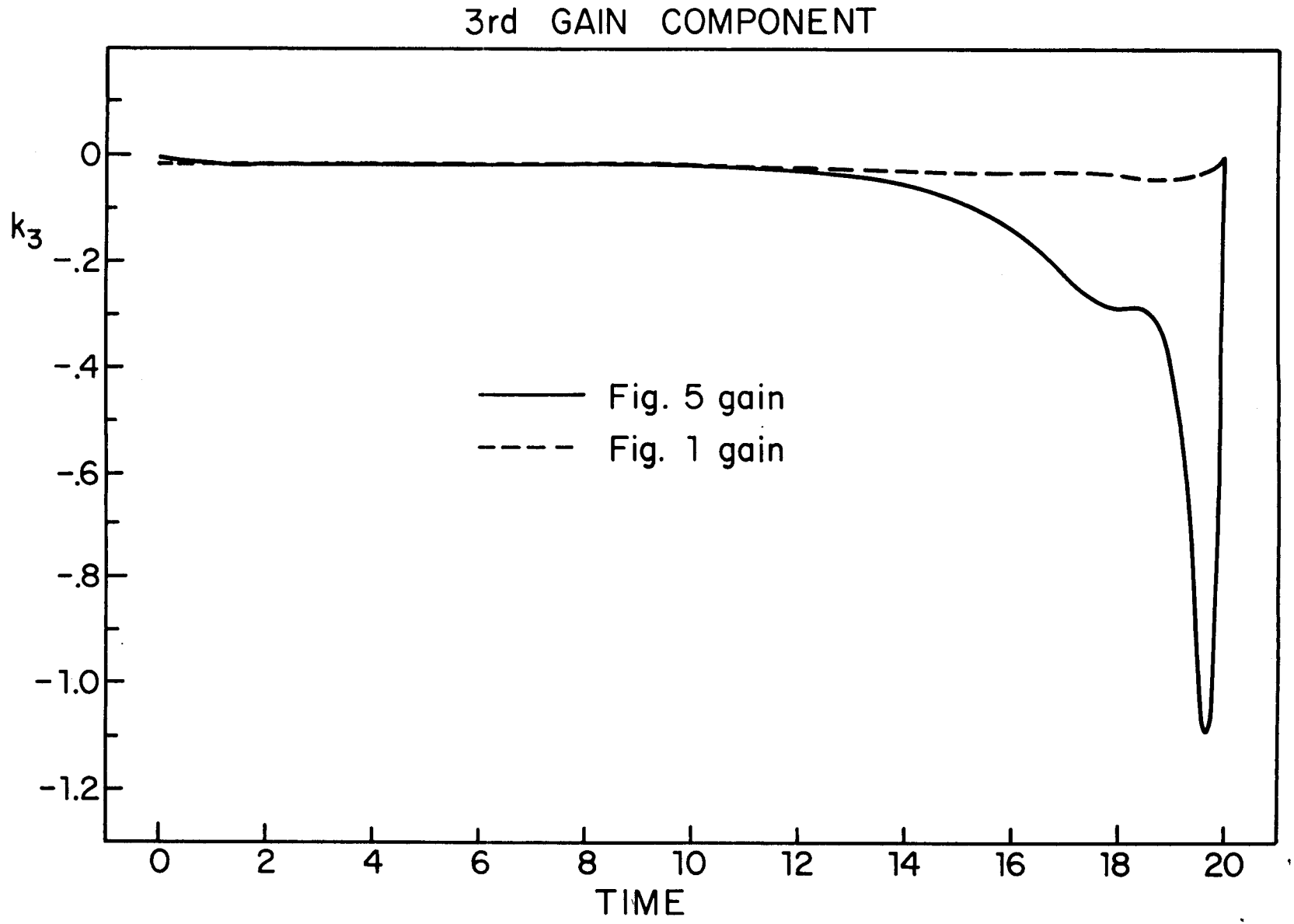
3rd GAIN COMPONENT
k_3
0
-.2
-.4
-.6
-.8
-1.0
-1.2
Fig. 5 gain
Fig. 1 gain
0
2
4
6
8
10
12
14
16
18
20
TIME

Fig. 7

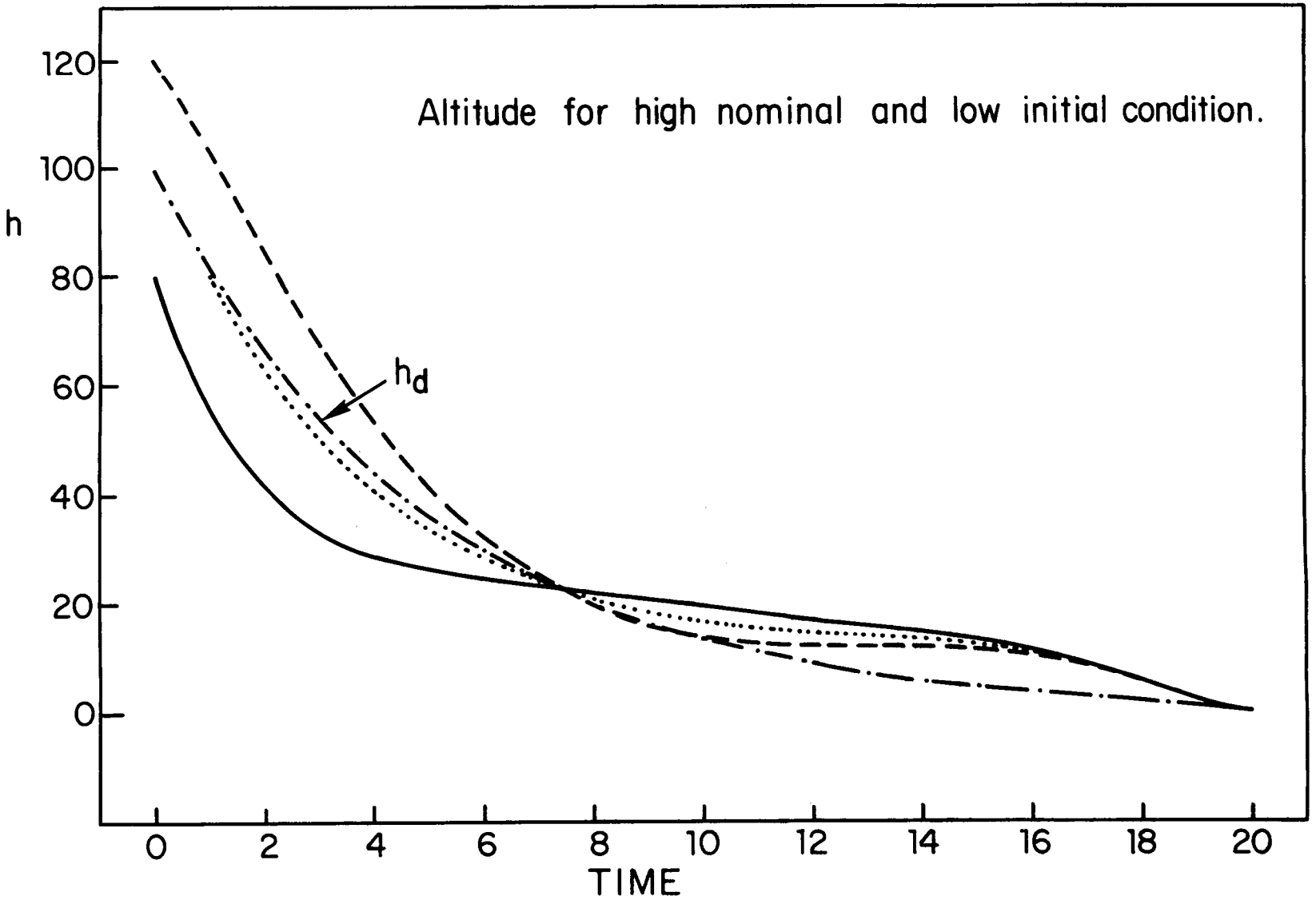
Altitude for high nominal and low initial condition.
h
120
100
80
60
40
20
0
h_d
0
2
4
6
8
10
12
14
16
18
20
TIME

Fig. 6

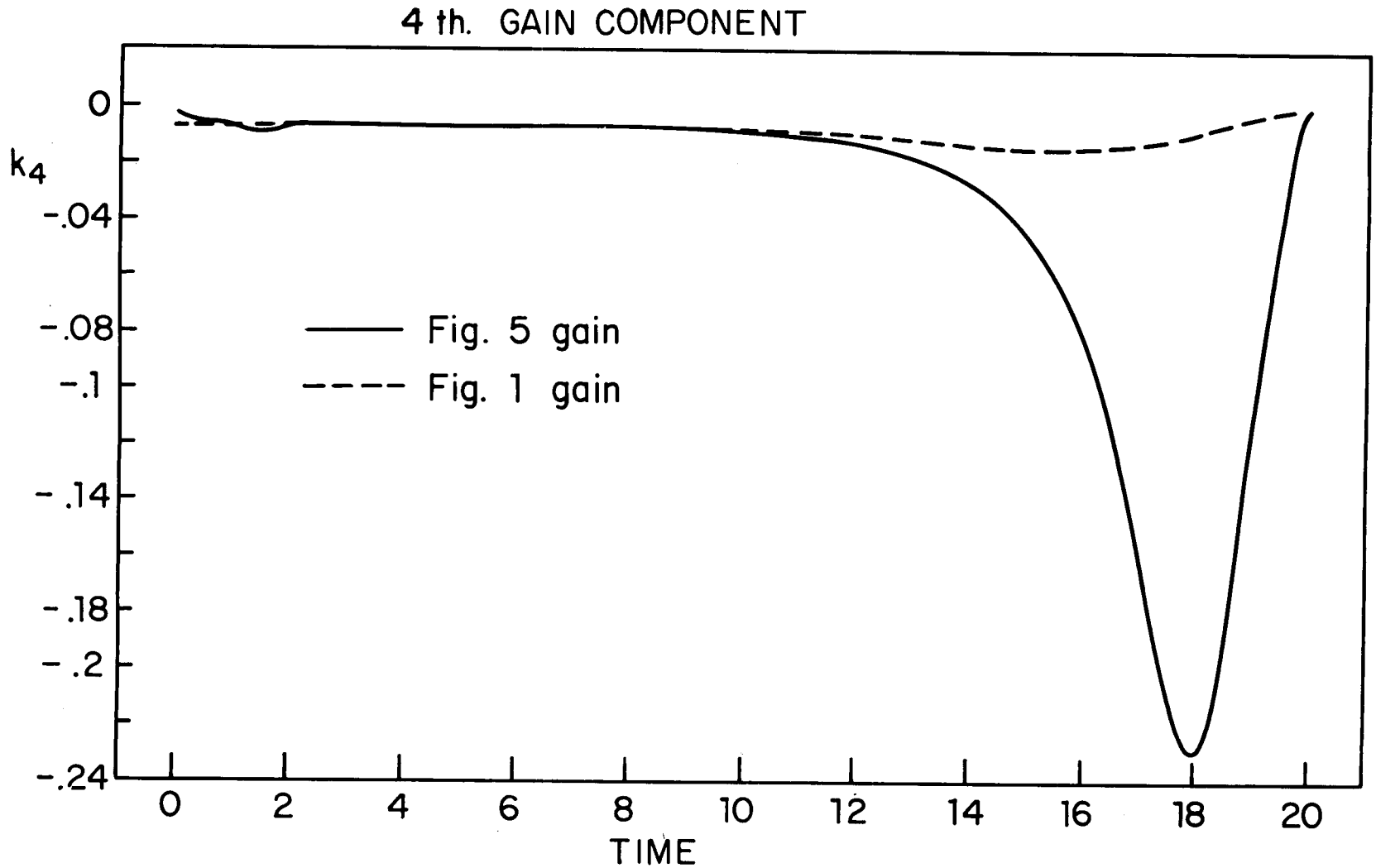
4 th. GAIN COMPONENT
k_4
0
-.04
-.08
-.1
-.14
-.18
-.2
-.24
Fig. 5 gain
Fig. 1 gain
0
2
4
6
8
10
12
14
16
18
20
TIME

Fig. 8

```
BEGIN
LOAD   F, G,Q1, T,ZR,ON, D,PC,I3, J, X, H,T2 H1 X1 X2 X3 H2
MULT   Q1,H1,    2Z,
TRANP  H1,    2Y
MULT   2Y 2Z      Q,
SUBT   ZR, F,     1
TRANP   F,    FT
TRANP   G,    GT
MULT   ON GT      C
MULT    G  C      5
NORM    Q,    NQ
NORM    5     N5
PSEUO NQ      PQ RK
MULT   PQ N5      M1
DECOM M1,      S,SJ,ER,PE, E,RK,
MULT    S  Q      Q
MULT   SJ  5      5
MULT   SJ  C      C
BLOT   NQ,
JUXTC   1  5,     2,
JUXTC   Q FT      3
JUXTR   2  3     PH,
ETPHI  PH, T,PH,
RICAT  Q,PH, C, D,PC,I3, P, K,AL,
RINT   P,PER K, K
MULT    G, K,    GK
SUBT    F GK      FC
ETPHI  FC   T P1
MULT   ZR   G     GM
TRNSI   J, K,ON, X,P1,GM, H,T2,
TRNSI   J, K,ON,X1,P1,GM,H2,T2,
TRNSI   J, K,ON,X2,P1,GM, H,T2,
TRNSI   J, K,ON,X3,P1,GM,H2,T2,
END
 F          8      8
0                  1.          0           0           0
0                  0           0           0           -2.93
-4.75               .78        0           0           0
0                  .086        0           -.11        -1.
0                  0           0           0           0
-.042              2.59        -.39        0           0
0                  0           0           0           0
0                  0           1.          0           0
0                  0           0           0           0
-1.                -73.14      3.18        0           0
0                  0           .086        0           -.11
-1.                0           0           0           0
.0086              .086        8.95        -.49
 G          8      2
0                  0           0           -3.91       .035
0                  -2.53       .31         0           0
0                  0           0           0           0
0
```

Fig. 9

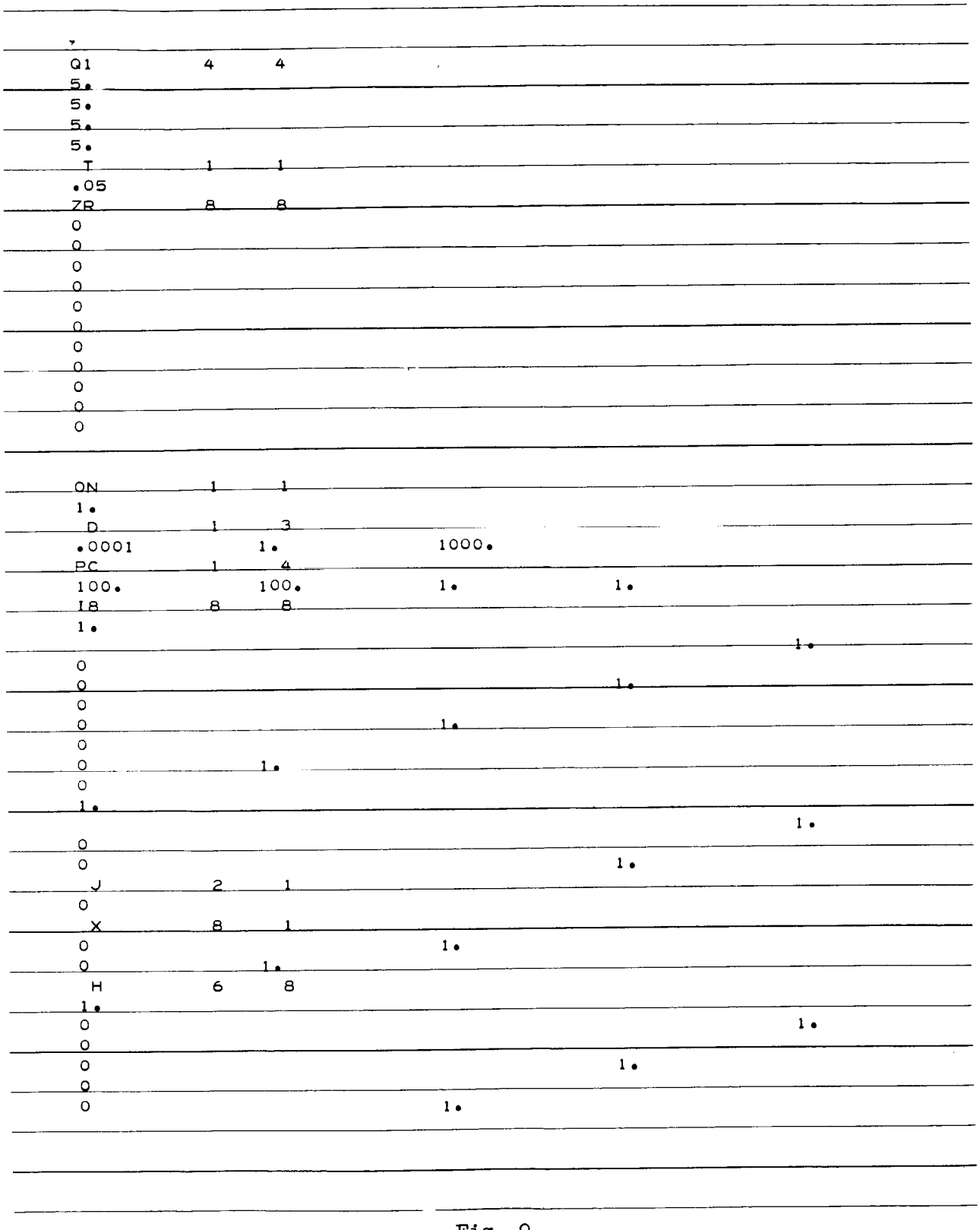
Q1 4 4
5.
5.
5.
5.
T 1 1
.05
ZR 8 8
0
0
0
0
0
0
0
0
0
0
ON 1 1
1.
D 1 3
.0001 1. 1000.
PC 1 4
100. 100. 1. 1.
I8 8 8
1.
1.
0
0 1.
0
0 1.
0
0 1.
0
1.
1.
0
0 1.
J 2 1
0
X 8 1
0 1.
0 1.
H 6 8
1.
0 1.
0
0 1.
0
0 1.

Fig. 9

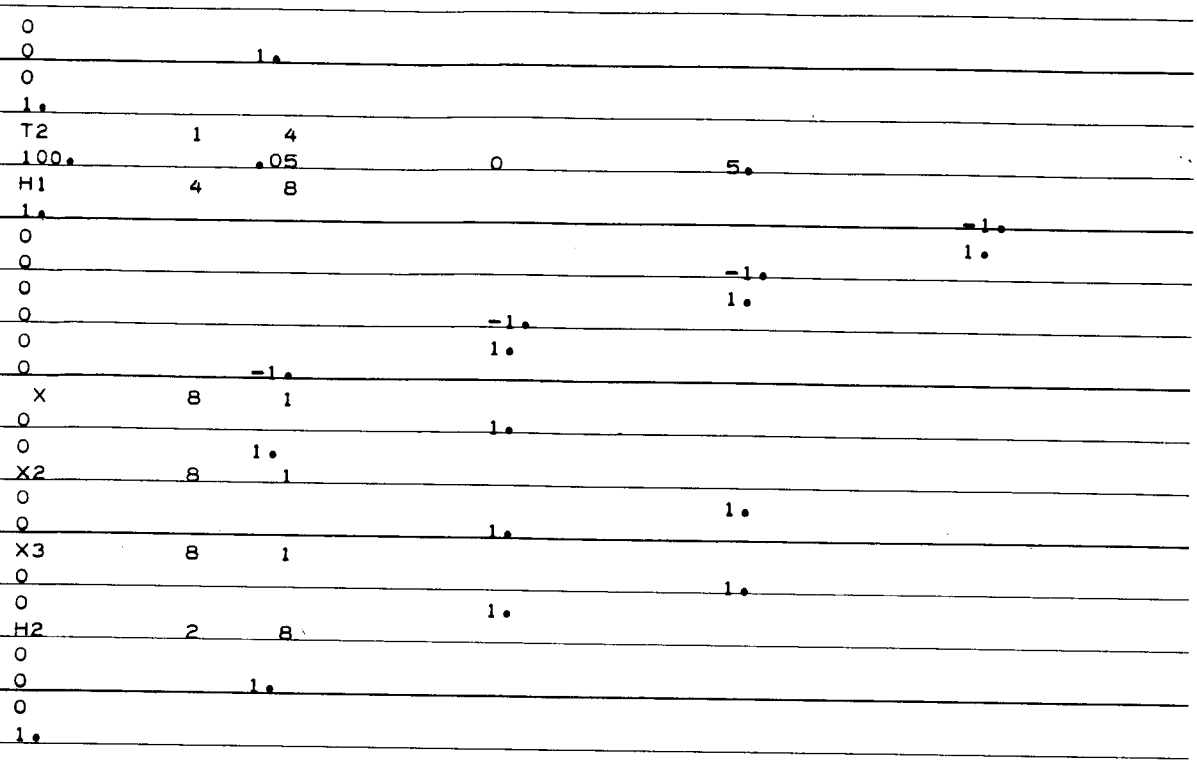
0
0
0
1.
T2 1 4
100. .05 0 5.
H1 4 8
1.
0
0
0
0
0
0
X 8 1
0
0
X2 8 1
0
0
X3 8 1
0
0
H2 2 8
0
0
0
1.
1.
-1.
1.
-1.
1.
-1.
1.
1.
1.
1.
1.
1.
1.
1.

Fig. 9