ON THE SIMULATION OF DYNAMIC SYSTEMS WITH LUMPED PARAMETERS AND TIME DELAYS

by

Nestor Leal-Cantu

I.M.E. Instituto Tecnologico y de Estudios

Superiores de Monterrey

(1966)

SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August,1967

Signature of Author . . . Nestor Leal Cantu . . . .
Department of Mechanical Engineering, Aug. 21, 1967

Certified by . Ronald C. Rozenberg., Asst. Prof. . . .
Thesis Supervisor

Accepted by . . . .
Chairman, Departmental Committee on Graduate Student

ON THE SIMULATION OF DYNAMIC SYSTEMS WITH

LUMPED PARAMETERS AND TIME DELAYS

by

Nestor Leal-Cantu

Submitted to the Department of Mechanical Engineering on August 21, 1967, in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

A method is developed for digital simulation of linear time-invariant dynamic systems with lumped parameters and time delays. Ordinarily, such systems can be described by a linear matrix differential-difference equation, which can be transformed to an infinite-dimensional difference equation whose solution is obtained in a recursive way.

As the present method depends on the accuracy of evaluation of the matrix exponential, a simple computational procedure based on the truncation of the infinite series for e^{AT} is described.

In addition, an algorithm is given that ensures that the transient state of an unforced linear time-invariant dynamic system with zero time delay is calculated to a specified accuracy.

Several sample problems are included.

Thesis Supervisor: Ronald Carl Rosenberg

Title: Assistant Professor of Mechanical Engineering

ACKNOWLEDGMENTS

The author is deeply indebted to Professor Ronald Carl Rosenberg, thesis advisor, for his encouragment at every stage of this work and the innumerable constructive criticisms for improvement toward making this report readable. The author also owes a debt of gratitude to the members of the ENPORT Project and to many other individuals who must remain anonymous due to lack of space. In a special category, the valuable suggestions of Professor Dean C. Karnopp and Mr. Yves Willems are greatly appreciated. Special thanks are due to Hojalata y Lamina S. A., Monterrey, Mexico which generously financed this work.

Work reported herein was supported in part by Project MAC, an M.I.T. research project sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Nonr-4102(01). Reproduction of this report or in part, is permitted for any purpose of the United States Government.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1-1 Description of the problem

This report presents a method for the simulation of linear time-invariant dynamic systems with lumped parameters and time delays.

In many industrial processes one often encounters a type of time delay called "transportation lag". This kind of delay is generated when process materials move from one point in a process to another point without any appreciable change taking place in the properties or characteristics of the process materials. Such delays may be caused by the flow of fluids through pipes, or by the motion of webs or filaments. Systems such as distillation columns and long heat exchangers are characterized by a multitude of small lags, which have an effect somewhat similar to that of time delays. The effects are not identical; however, some insight may be gained by using time delays models. The control of composition in a chemical reactor has been selected as a typical problem and this is depicted in section 5-2.

Models having delays often arise in the study of systems with a mixture of lumped and distributed elements. An interesting form of topological representation suitable for such systems has been invented by Prof. H. M. Paynter at M.I.T., and is called the bond graph. Rosenberg (17) and Auslander (1)* describe its use in modeling in some detail.

Many other physical systems, such as electrical, mechanical and

---

* Numbers in parenthesis refer to items in the bibliography.

hydraulic transmission lines, and certain types of structural problems, are good examples of distributed systems which can be modeled using the delay operator. These systems often are analyzed as two-port chains, and usually the equations are slightly more involved than the type treated in this report. It is suggested that the reader interested in these kind of problems consult Koepcke (9) and Vaughn (20), as well as any standard text treating transmission phenomena.

1-2 Formulation of the approach

As an extension to the use of ordinary differential equations which arise when the future behavior of the system depends only upon its present state and not upon its past history, many systems that include time delays can be described by a linear matrix differential-difference equation. That is, the system is described by

$$\dot{\underline{X}}(t) = \sum_{i=1}^{n} A_i \, \underline{X}(t - T_i) + \sum_{j=1}^{m} D_j \, \underline{U}(t - T_j) ,$$

where $\underline{X}$ and $\underline{U}$ are the state and input vectors, respectively and T_i and T_j are some fixed delay times. A_i are a set of n x n matrices, and D_j are a set of n x r matrices. Techniques such as the direct method of lyapunov or laplace transforms can be used in the analysis of the equation. However, the use of these techniques frequently requires extensive computation, and for that reason they are not practical for hand analysis. At this step, designers and analysts are forced to rely on the digital computer as a computing aid.

Because matrix manipulations are so convenient to implement on a digital computer, many existing dynamic systems programs are based on a matrix formulation of the problem. This convenience, together with the inherent elegance of the matrix approach, is helping to promote its acceptance among systems theorists.

This report analyzes systems governed by the following differential-difference equation, for which it is desired to have a time sampled version of the state response:

$$\dot{\underline{X}}(t) = A\,\underline{X}(t) + B\,\underline{X}(t - T) + D_1\underline{U}(t) + D_2\underline{U}(t - T)$$

where

T = time delay.

$\underline{X}(t)$ = $(n \times 1)$ vector. It is called the state vector.

$\underline{U}(t)$ = $(r \times 1)$ vector. It is the forcing signal or input vector, and it is assumed to be constant between samples.

A, B = $(n \times n)$ constant coefficient matrices.

D_1, D_2 = $(n \times r)$ constant driving matrices.

Koepcke (9) shows that the equivalent difference equation is (see section 3-1)

$$\underline{X}(t + \tau) = \sum_{i=0}^{\infty} [\Phi_i(\tau)\,\underline{X}(t - iN\tau) + \Delta_i(\tau)\,\underline{U}(t - iN\tau)]$$

where $N = \frac{T}{\tau}$, and Φ_i and Δ_i are called plant transition matrices and control transition matrices respectively.

The accuracy of evaluation of these sets of transition matrices depends upon the accuracy of evaluation of the matrix exponential. In section 2-3 a simple procedure based on the truncation of the infinite series of e^{AT} (11,6), which guarantees a specified accuracy in the matrix exponential, is described.

Also, a procedure is developed (21) to ensure that the calculated transient state of unforced linear time-invariant dynamic systems with zero time delay, is accurate to a specified tolerance.

Several sample problems are presented to demonstrate the computation techniques.

1-3 Application of results in dynamic simulation

The two sets of simulators deduced throughout the development of this work, were tested on the time-shared IBM 7094 operated by Project MAC, and the entire operation, input and output, was carried out at an IBM 1050 remote console typewriter. The algorithms will be part of the ENPORT Project which is being carried out at the mechanical engineering department under the direction of Professor Rosenberg.

ENPORT is a digital computer program that accepts a bond graph description of a dynamic system and produces its time response. Work is being done on the theory of bond graphs, and a systematic graphical method has been developed for generating the state differential equations. ENPORT is organized in such a way that a broad class of nonlinear, active and passive, mixed energy-type systems can be handled.

The wakelike nature of certain types of distributed systems make simulation by means of the digital computer, with its ability to exactly model the time delay operator, very natural. A simulation method based on delay-bond modeling has been developed by Auslander (1).

CHAPTER 2

DYNAMIC RESPONSE OF LINEAR TIME-INVARIANT SYSTEMS

The analysis of many systems problems encountered in scientific and engineering investigations can be performed by either one of two major approaches. The essentially block diagram approach, involves the determination of the transfer characteristics of the system components and the overall transfer characteristics. The second approach is based primarily upon the characterization of a system by a number of coupled first order differential equations which govern the behavior of the state variables. This technique is often implemented with the aid of a state variable diagram and is referred to as the state-variable approach.

2-1 System Characterization by State Variables

From the point of view of system analysis it is convenient to classify the variables which characterize or are associated with any system into (1) input, or forcing signals, U_i, which in essence represent the stimuli generated by systems other than the one under investigation and which influence the system behavior; (2) output, or response, variables Y_i, which describe those aspects of system behavior that are of interest to the investigator; and (3) state variables X_i, which characterize the dynamic behavior of the system under investigation.

One way of defining state variables is by making use of the state variable diagram. A state variable diagram is made up of integrators, coefficients and summing devices. It describes the relationships among the state variables and provide physical interpretations of them. The outputs of the integrators denote the state variables.

For continuous-time systems the state variable diagram is the same as the analog-computer simulation diagram. The state variable diagram may be derived from the overall transfer function of the system in three different ways (1) direct programming, (2) parallel programming, and (3) iterative programming. These methods are later ilustrated in the chapter corresponding to the solution to sample problems. Further information can be obtained from Tou (19), Schwarz and Friedland (18) and Ogata (15).

2-2 Digital Solution of the Matrix Differential Equation

A linear time-invariant system or process can be described by a set of first order linear differential equations with constant coefficients, which may be expressed in matrix form as

$$\dot{\underline{X}}(t) = A\,\underline{X}(t) + D\,\underline{U}(t) \tag{2.1}$$

where

A is the coefficient matrix

D is the driving matrix

$\underline{X}$ is the state variable vector

$\underline{U}$ is the state forcing signal vector

By analogy to the scalar case, the solution of eq. (2.1) is

$$\underline{X}(T) = e^{A(T - t_o)}\underline{X}(t_o) + \int_{t_o}^{T} e^{A(T - \tau)} D\,\underline{U}(\tau)\,d\tau \tag{2.2}$$

with the initial conditions given by $\underline{X}(t_o)$.

For simplicity let $t_o = 0$, and let us define

$$\Phi(T) = e^{AT} \tag{2.3}$$

as the transition matrix of the process. An equivalent name is the matrix exponential.

Therefore eq. (2.2) can be reduced to

$$\underline{X}(T) = \Phi(T)\ \underline{X}(0) + \Phi(T) \int_0^T e^{-A\tau}\ D\ \underline{U}(\tau)\ d\tau \tag{2.4}$$

If T is small compared to the shortest period of interest in $\underline{U}(t)$, $\underline{U}(t)$ may be approximated over the region by $\underline{U}(0)$.

Then eq. (2.4) becomes

$$\underline{X}(T) = \Phi(T)\ \underline{X}(0) + \Phi(T) \left(\int_0^T e^{-A\tau}\ d\tau \right) D\ \underline{U}(0) \tag{2.5}$$

By integration of the series of $e^{-A\tau}$

$$\int_0^T e^{-A\tau}\ d\tau = A^{-1}[1 - \Phi(-T)] \tag{2.6}$$

Thus

$$\underline{X}(T) = \Phi(T)\ \underline{X}(0) + \Phi(T)\ A^{-1}[1 - \Phi(-T)]\ D\ \underline{U}(0) \tag{2.7}$$

Let us define

$$\Delta(T) = [e^{AT}\ A^{-1} - e^{AT}\ A^{-1}\ e^{-AT}]\ D \tag{2.9}$$

as the control transition matrix.

From the series definition of e^{-AT}, it is observed that

$$A^{-1}\ e^{-AT} = e^{-AT}\ A^{-1}$$

Therefore, eq. (2.9) becomes

$$\Delta(T) = [e^{AT}\ A^{-1} - e^{AT}\ e^{-AT}\ A^{-1}]\ D$$

$$\Delta(T) = [e^{AT}\ A^{-1} - A^{-1}]\ D$$

or

$$\Delta(T) = [(e^{AT} - I)\ A^{-1}]\ D \tag{2.10}$$

Thus eq. (2.8) can finally be written as

$$\underline{X}(T) = e^{AT}\,\underline{X}(0) + [(e^{AT} - I)\,A^{-1}]\,D\,\underline{U}(0)\ , \qquad (2.11)$$

or

$$\underline{X}(T) = \Phi(T)\,\underline{X}(0) + \Delta(T)\,\underline{U}(0) \qquad (2.12)$$

In general eq. (2.12) can be expressed as

$$\underline{X}(\overline{K+1}T) = \Phi(T)\,\underline{X}(KT) + \Delta(T)\,\underline{U}(KT) \qquad (2.13)$$

which indicates that the state vector of the process after a particular interval depends upon the previous vector and also depends upon the forcing vector evaluated at the previous time.

There are several methods available for computing the closed form expression for e^{AT}, either as a special case of the study of the functions of a matrix or by a purely algebraic method based on the Laplace Transform. It is suggested, for those interested in these schemes, that they consult Ogata (15), Zadeh and Desoer (23), or Bellman (2).

2-3 Digital Evaluation of the Matrix Exponential

e^{AT} is given by

$$e^{AT} = e^{B} = I + B + \frac{B}{2}\left(\frac{B}{1!}\right) + \frac{B}{3}\left(\frac{B^2}{2!}\right) + ---- \qquad (2.14)$$

note that each term in parenthesis is equal to the previous term. This provides a convenient recursion scheme.

To ensure a reasonable truncation of the series, it is necessary to judge the convergence of the series. The norm of a matrix A is a real, non-negative number denoted by $\|A\|$, that gives a measure of the size of the matrix elements.

Let

$$\Phi(\tau) = e^{A\tau} = M + R$$

where M is the truncated matrix which is an approximation of $e^{A\tau}$ (see reference 11)

$$M = \sum_{i=0}^{K} \frac{(A\tau)^i}{i!} \tag{2.15}$$

and R is the remainder matrix

$$R = \sum_{i=K+1}^{\infty} \frac{(A\tau)^i}{i!} \tag{2.16}$$

If each element in the matrix $e^{A\tau}$ is required with an accuracy of "d" significant digits, then

$$|r_{ij}| \leq 10^{-d}|m_{ij}| \tag{2.17}$$

where r_{ij} and m_{ij} are elements of matrices R and M respectively.

Let us define the norm of matrix A as:

$$\|A\| = \min\{\max_i [\sum_j |a_{ij}|], \max_j [\sum_i |a_{ij}|]\} \tag{2.18}$$

For this norm, we have

$$\|A\ B\| \leq \|A\|\,\|B\| \tag{2.19}$$

$$|a_{ij}| \leq \|A\| \tag{2.20}$$

and

$$\|A\| + \|B\| \leq \|A\| + \|B\| \tag{2.21}$$

Then, it follows that

$$|r_{ij}| \leq \left\|\sum_{i=K+1}^{\infty} \frac{(A\tau)^i}{i!}\right\| \leq \sum_{i=K+1}^{\infty} \frac{\|A\|^i\ \tau^i}{i!} \tag{2.22}$$

if the same norm is applied to the remainder matrix R.

Upon expansion of eq. (2.22)

$$|r_{ij}| \leq \frac{\|A\|^{K+1}\tau^{K+1}}{(K+1)!} + \frac{\|A\|^{K+2}\tau^{K+2}}{(K+2)!} + \text{-----} \qquad (2.23)$$

and, calling ε the ratio of the second term to the first

$$\varepsilon = \frac{\dfrac{\|A\|^{K+2}\tau^{K+2}}{(K+2)!}}{\dfrac{\|A\|^{K+1}\tau^{K+1}}{(K+1)!}} = \frac{\|A\|\tau}{K+2} \qquad (2.24)$$

Therefore

$$\frac{\|A\|\tau}{K} \leq \varepsilon \qquad (2.25)$$

Making the substitution of eq. (2.19) into eq.(2.23), it follows that

$$|r_{ij}| \leq \left\|\frac{(A\tau)^K}{K!}\right\| \left\|\frac{A\tau}{K+1}\right\| + \left\|\frac{(A\tau)^K}{K!}\right\| \left\|\frac{(A\tau)^2}{(K+2)(K+1)}\right\| +$$

$$+ \left\|\frac{(A\tau)^K}{K!}\right\| \left\|\frac{(A\tau)^3}{(K+3)(K+2)(K+1)}\right\| + \text{---} , \qquad (2.26)$$

or

$$|r_{ij}| \leq \left\|\frac{(A\tau)^K}{K!}\right\| \left(\left\|\frac{A\tau}{K+1}\right\| + \left\|\frac{A\tau}{K+2}\right\| \left\|\frac{A\tau}{K+1}\right\| + \right.$$

$$\left. + \left\|\frac{A\tau}{K+3}\right\| \left\|\frac{A\tau}{K+2}\right\| \left\|\frac{A\tau}{K+1}\right\| + \text{------} \right) . \qquad (2.27)$$

Thus

$$|r_{ij}| \leq \left\|\frac{(A\tau)^K}{K!}\right\| \left(\left\|\frac{A\tau}{K+1}\right\| \left\{ 1 + \left\|\frac{A\tau}{K+2}\right\| + \right.\right.$$

$$\left.\left. + \left\|\frac{A\tau}{K+3}\right\| \left\|\frac{A\tau}{K+2}\right\| + \text{---} \right\} \right) \qquad (2.28)$$

Now, because any factor of the form $\left\|\frac{A\ \tau}{K+a}\right\|$ for a>2 is always less than ε, by eq. (2.24), then

$$|r_{ij}| \leq \frac{\|(A\tau)^K\|}{K!}\left(\frac{\|A\tau\|}{K+1}\{1+\varepsilon+\varepsilon^2+\varepsilon^3+\varepsilon^4+ ---\}\right) \qquad (2.29)$$

If $\varepsilon<1$, eq. (2.29) takes the form

$$|r_{ij}| \leq \frac{\|(A\tau)^K\|}{K!}\left(\frac{\|A\ \tau\|}{K+1}\cdot\frac{1}{1-\ \varepsilon}\right) \qquad (2.30)$$

This equation is suggested by Everling (6) as the upper bound in the remainder matrix R.

In order to initialize the procedure, a certain K has to be chosen, but this K cannot be arbitrary, because it may happen that $\varepsilon>1$, and relation (2.30) would not hold any more.

This situation can be solved using eq. (2.25); thus

$$K \geq \frac{\|A\ \tau\|}{\varepsilon}$$

In order to ensure that $\varepsilon \leq 1/2$, the initial condition for K should be

$$K \geq 2\ \|A\tau\| \qquad (2.31)$$

However, it is possible that $\|A\tau\|$ be less than 1/2; then K would be less than one. So, in order to avoid this possibility, an initial value of K can be obtained from

$$K = \max\ [\ 2\|A\tau\|,\ 2\ ] \qquad (2.32)$$

At this point, Everling (6) suggests that K be incremented by half of its initial value, in the course of iteration.

Although the matrix series approach for the evaluation of the transition matrix is suitable for digital computation, the disadventage stems from the convergence requirements for the series $e^{A\tau}$, so it would be desirable to speed the computation.

This can be done recalling the basic relationship

$$e^{A\tau} = \left(e^{A\tau/\alpha}\right)^{\alpha} \qquad (2.33)$$

where α is chosen using the following expression

$$\alpha = 2^{\beta} \geqq \max_{i,j}\left(\left|a_{ij}\right|\tau\right) \qquad (2.34)$$

where β is the smallest integer allowed.

The idea is to compute $e^{A\tau/\alpha}$, because the norm of $A\tau/\alpha$ is smaller than $A\tau$, and the series will converge faster. Once the addition of the corresponding elements in the matrix terms of the infinite series is done, all that is required is to raise the result to the power α. The last step involves very few matrix multiplications, because α is a power of 2; for example, if $\alpha = 32$ only 5 matrix multiplications are performed at the end of the computation.

The steps presented in this section are summarized in a flow diagram in chapter 4.

2-4 Error bounds in the transient response

Although the matrix e^{AT} can be obtained within prescribed accuracy, the truncation error of the matrix series, and the roundoff error do propagate in the state vector with increasing time.

It is desirable, therefore to derive recursion relations which bound the propagated error due to these sources. Whitney (21) suggests one method.

The homogeneous case of eq. (2.13) is

$$\underline{X}(\overline{K+1}\ T) = \Phi(T)\ \underline{X}(K\ T) \qquad (2.35)$$

If eq. (2.15) is used in place of $\Phi(T)$, the numerical calculation reads

$$\underline{X}_*(\overline{K+1}\ T) = M\ \underline{X}_*(K\ T) \tag{2.36}$$

where $\underline{X}_*(\overline{K+1}\ T)$ is the perturbed state vector obtained from numerical calculation.

The propagated error at time (K+1) T due to the approximate M is

$$\underline{E}(\overline{K+1}\ T) = \underline{X}(\overline{K+1}\ T) - \underline{X}_*(\overline{K+1}\ T) \tag{2.37}$$

Rewriting eq. (2.35) and substracting eq. (2.36) from it yields

$$\underline{X}(\overline{K+1}\ T) - \underline{X}_*(\overline{K+1}\ T) = [M+R][\underline{X}_*(K\ T) + \underline{E}(K\ T)] - M\ \underline{X}_*(K\ T) \quad , \tag{2.38}$$

or

$$\underline{E}(\overline{K+1}\ T) = [M + R]\underline{E}(K\ T) + R\ \underline{X}_*(K\ T) \tag{2.39}$$

From eq. (2.17)

$$|r_{ij}| \leqq 10^{-d} |m_{ij}| \tag{2.17}$$

We can define

$$R_* = |r_{ij}|\ I \tag{2.40}$$

where I is a matrix each of whose elements is unity. Replacing R with R_* in (2.39), we obtain the running error bound for $\underline{E}(\overline{K+1}\ T)$, that is

$$\underline{E}(\overline{K+1}\ T) = [M + R]\ \underline{E}(K\ T) + R_*\ \underline{X}_*(K\ T) \tag{2.41}$$

The computation may be initialized assuming $\underline{E}(0)$ is zero.

CHAPTER 3

DYNAMIC RESPONSE OF LINEAR TIME-INVARIANT SYSTEMS WITH LUMPED PARAMETERS AND TIME DELAYS

It has been found that many industrial processes in which transportation lags are common can be described by a system of differential-difference equations. The chemical process industry offers many examples.

This chapter analyzes the special case of a system subject to one delay, and a technique suitable for digital computation is described. The derivation follows a criterion developed by Koepcke (9).

3-1 Digital solution of the matrix differential-difference equation

Consider a dynamic system which is governed by the following differential-difference equation

$$\dot{\underline{X}}(t) = A\,\underline{X}(t) + B\,\underline{X}(t-T) + D_1\underline{U}(t) + D_2\underline{U}(t-T) \qquad (3.1)$$

where

$\underline{X}(t)$ = (n x 1) vector, referred to as the state vector;

$\underline{U}(t)$ = (r x 1) input vector, assumed constant between samples;

i.e. $U(t) = U(t_k)$ for $t_k \leq t \leq t_{K+1}$;

A, B = (n x n) constant coefficient matrices; and

D_1, D_2 = (n x r) constant driving matrices

Let us consider first the homogeneous part of eq. (3.1); that is

$$\dot{\underline{X}}(t) = A\,\underline{X}(t) + B\,\underline{X}(t-T) \qquad (3.2)$$

Taking the laplace transform of eq. (3.2),

$$S\underline{X}(S) - \underline{X}(0) = (A + B\ e^{-ST})\ \underline{X}(S) \tag{3.3}$$

or

$$\underline{X}(S) = [SI - (A + B\ e^{-ST}]^{-1}\ \underline{X}(0) \tag{3.4}$$

Defining $Z \equiv e^{-ST}$, then

$$\underline{X}(S) = [SI - (A + B\ Z)]^{-1}\ \underline{X}(0)\ , \tag{3.5}$$

or

$$\underline{X}(S) = \frac{1}{S}\ [I - (A + BZ)/S]^{-1}\ \underline{X}(0)\ . \tag{3.6}$$

Let $W = [I - R]^{-1}$, where $R = \frac{A + BZ}{S}$; then

$$W = I + R + R^2 + R^3 + R^4 + \text{------}\ . \tag{3.7}$$

Therefore, one should choose an "S" large enough to ensure that eq. (3.7) is valid.

Thus

$$\underline{X}(S) = \frac{1}{S}\ [I + \frac{A + BZ}{S} + \frac{(A + BZ)^2}{S^2} + \frac{(A + BZ)^3}{S^3} + \\ + \frac{(A + BZ)^4}{S^4} + \text{---}]\ \underline{X}(0) \tag{3.8}$$

Recall the facts that

$$(A + BZ)^2 = A^2 + A(BZ) + (BZ)A + (BZ)^2$$

$$(A + BZ)^3 = A^3 + A^2(BZ) + A(BZ)A + A(BZ)^2 + (BZ)A^2 + \\ + (BZ)A(BZ) + (BZ)^2A + (BZ)^3$$

$$(A + BZ)^4 = A^4 + A^3(BZ) + A^2(BZ)A + A^2(BZ)^2 + A(BZ)A^2 + \\ + A(BZ)A(BZ) + A(BZ)^2A + A(BZ)^3 + (BZ)A^3 + \\ + (BZ)A^2(BZ) + (BZ)A(BZ)A + (BZ)A(BZ)^2 + \\ + (BZ)^2A^2 + (BZ)^2A(BZ) + (BZ)^3A + (BZ)^4$$

etc.

Then, arranging by terms of equal delay,

$$\underline{X}(S) = [\frac{I}{S} + \frac{A}{S^2} + \frac{A^2}{S^3} + \frac{A^3}{S^4} + \frac{A^4}{S^5} + \text{-----}]\ \underline{X}(0) +$$

$$+ [\frac{BZ}{S^2} + \frac{A(BZ) + (BZ)A}{S^3} + \frac{A^2(BZ) + A(BZ)A + (BZ)A^2}{S^4} +$$

$$+ \frac{A^3(BZ) + A^2(BZ)A + A(BZ)\ A^2 + (BZ)A^3}{S^5} + \text{--}\ ]\ \underline{X}(0) +$$

$$+ [\frac{(BZ)}{S^3} + \frac{A(BZ)^2 + (BZ)A(BZ) + (BZ)^2A}{S^4} + \frac{A^2(BZ)^2 + A(BZ)A(BZ)}{S^5}$$

$$+ \frac{A(BZ)^2A + (BZ)A^2(BZ) + (BZ)A(BZ)A + (BZ)^2A^2}{S^5} +$$

$$+ \text{--}\ ]\ \underline{X}(0) +$$

$$+ [\frac{(BZ)^3}{S^4} + \frac{A(BZ)^3 + (BZ)A(BZ)^2 + (BZ)^2A(BZ) + (BZ)^3A}{S^5} +$$

$$+ \text{--}\ ]\ \underline{X}(0) +$$

$$+ [\frac{(BZ)^4}{S^5} + \text{-----------------------------------}\ ]\ \underline{X}(0) +$$

$$+ \text{------} \tag{3.9}$$

Now, because

$$Z\underline{X}(t) = \underline{X}(t - T) \qquad (Z \equiv e^{-ST}) \ , \tag{3.10}$$

We have

$$Z\underline{X}(0) = \underline{X}(-T) \ ,$$

$$Z^2\underline{X}(0) = \underline{X}(-2T) \ ,$$

$$Z^3\underline{X}(0) = \underline{X}(-3T) \ , \text{ and so forth.}$$

Therefore, $\underline{X}(S)$ can be arranged in the following way.

$$\underline{X}(S) = \Phi_0(S)\underline{X}(0) + \Phi_1(S)\underline{X}(-T) + \Phi_2(S)\underline{X}(-2T) + \Phi_3(S)\underline{X}(-3T) + $$
$$+ \Phi_4(S)\underline{X}(-4T) + \text{---} \tag{3.11}$$

where

$$\Phi_0(S) = \frac{I}{S} + \frac{A}{S^2} + \frac{A^2}{S^3} + \frac{A^3}{S^4} + \frac{A^4}{S^5} + \text{-----------} \tag{3.12}$$

$$\Phi_1(S) = \frac{B}{S^2} + \frac{AB + BA}{S^3} + \frac{A^2B + ABA + BA^2}{S^4} + $$

$$+ \frac{A^3B + A^2BA + ABA^2 + BA^3}{S^5} + \text{----} \tag{3.13}$$

$$\Phi_2(S) = \frac{B^2}{S^3} + \frac{AB^2 + BAB + B^2A}{S^4} + \frac{A^2B^2 + ABAB + AB^2A + BA^2B}{S^5} + $$

$$+ \frac{BABA + B^2A^2}{S^5} + \text{---------------} \tag{3.14}$$

$$\Phi_3(S) = \frac{B^3}{S^4} + \frac{AB^3 + BAB^2 + B^2AB + B^3A}{S^5} + \text{----} \tag{3.15}$$

$$\Phi_4(S) = \frac{B^4}{S^5} + \text{------------------------------} \tag{3.16}$$

Rearranging terms, it follows that

$$\Phi_0(S) = \frac{I}{S} + \frac{A}{S^2} + \frac{A^2}{S^3} + \frac{A^3}{S^4} + \frac{A^4}{S^5} + \text{-----------} \tag{3.17}$$

$$\Phi_1(S) = \frac{B}{S^2} + \frac{AB + BA}{S^3} + \frac{A(AB + BA) + BA^2}{S^4} + $$

$$+ \frac{A[A(AB + BA) + BA^2] + BA^3}{S^5} + \text{-----} \tag{3.18}$$

$$\Phi_2(S) = \frac{B^2}{S^3} + \frac{AB^2 + B(AB + BA)}{S^4} + \frac{A[AB^2 + B(AB + BA)]}{S^5} + $$

$$+ \frac{B[A(AB + BA) + BA^2]}{S^5} + \text{-----------} \tag{3.19}$$

$$\Phi_3(S) = \frac{B^3}{S^4} + \frac{AB^3 + B[AB^2 + B(AB + BA)]}{S^5} + \text{--} \tag{3.20}$$

$$\Phi_4(S) = \frac{B^4}{S^5} + \text{------------------------------} \tag{3.21}$$

Let us try to find a relationship among the coefficients. With this idea in mind, we shall form the array shown in figure 3.1:

s^{-1}	s^{-2}	s^{-3}	s^{-4}	s^{-5}
I	A	A^2	A^3	A^4
0	B	AB + BA	$A(AB + BA) + BA^2$	$A[A(AB + BA) + BA^2] + BA^3$
0	0	B^2	$A(B^2) + B(AB + BA)$	$A[A(B^2) + B(AB + BA)] + B[A(AB + BA) + BA^2]$
0	0	0	B^3	$A[B^2] + B[A(B^2) + B(AB + BA)]$
0	0	0	0	B^4
0	0	0	0	0
↓	↓	↓	↓	↓

Figure 3.1 Array of the elements of the laplace-transformed transition matrices

It is seen that the correlation among the elements (call any element by $C_{i,j}$) is

s^{-1}	s^{-2}	s^{-3}	s^{-4}	s^{-5}
C_{oo}	C_{o1}	C_{o2}	C_{o3}	C_{o4}
0	C_{11}	C_{12}	C_{13}	C_{14}
0	0	C_{22}	C_{23}	C_{24}
0	0	0	C_{33}	C_{34}
0	0	0	0	C_{44}

where the arrows indicate the inmediate dependance; i.e., C_{12} depends on C_{o1} and C_{11}, etc.

From a careful study of the array in fig. 3.1, it is found that

$$\frac{C_{i,j}}{s^j} = \frac{A\ C_{i,j-1}}{s^{j+1}} + \frac{B\ C_{i-1,j-1}}{s^{j+1}} \tag{3.22}$$

where "i" is the subindex denoting row and "j" is the subindex denoting column.

The following conditions should be added, in order to initialize a computational procedure

$$C_{-1,j} = 0 \qquad j \geq 0 \tag{3.23}$$

$$C_{o,o} = I \tag{3.24}$$

$$C_{i,o} = 0 \qquad i > 0 \tag{3.25}$$

The inverse laplace transform of eq. (3.22) yields (note: $L[t^n/n!] = 1/s^{n+1}$)

$$[C_{i,j}]\,\frac{\tau^{j-1}}{(j-1)!} = [AC_{i,j-1}]\,\frac{\tau^j}{j!} + [BC_{i-1,j-1}]\,\frac{\tau^j}{j!} \qquad (3.26)$$

Therefore

$$C_{i,j} = [AC_{i,j-1}]\,\frac{\tau^j}{j!}\,\frac{(j-1)!}{\tau^{j-1}} + [BC_{i-1,j-1}]\,\frac{\tau^j}{j!}\,\frac{(j-1)!}{\tau^{j-1}}\ , \qquad (3.27)$$

or

$$C_{i,j} = \frac{[AC_{i,j-1}]\tau + [BC_{i-1,j-1}]\tau}{j}\ . \qquad (3.28)$$

Changing j for j+1, eq. (3.28) takes the final form

$$C_{i,j+1} = \frac{[A\tau]C_{i,j} + [B\tau]C_{i-1,j}}{J+1} \qquad (3.29)$$

Actually eq. (3.29) gives all coefficients without any need to multiply them by $\frac{\tau^j}{j!}$.

This is because τ has been associated with matrix A and B, and in order to compute any $C_{i,j+1}$, the initial conditions given by eqs. (3.23), (3.24) and (3.25) have to be considered.

The computation of the $C_{i,j+1}$ is done in a recursive way, as given by eq. (3.29). Once they are computed, they may be substituted in the inverse laplace transformation of eqs. (3.17), (3.18), etc., so that $\Phi_0(\tau)$, $\Phi_1(\tau)$, $\Phi_2(\tau)$, ... can be generated. The last set of matrices are called "plant transition matrices".

Returning to eq. (3.11), if e^{tS} is multiplied into both sides, then

$$e^{tS}\underline{X}(S) = \Phi_0(S)e^{tS}\underline{X}(0) + \Phi_1(S)e^{tS}\underline{X}(-T) + \Phi_2(S)e^{tS}\underline{X}(-2T) + $$
$$+ \Phi_3(S)e^{tS}\underline{X}(-3T) + --- \ , \qquad (3.30)$$

or

$$e^{tS}\underline{X}(S) = \Phi_0(S)\underline{X}(t) + \Phi_1(S)\underline{X}(t - T) + \Phi_2(S)\underline{X}(t - 2T) + + \Phi_3(S)\underline{X}(t - 3T) + \text{-----} . \tag{3.31}$$

Taking the inverse laplace transform of eq. (3.31), it turns out to be

$$\underline{X}(t + \tau) = \Phi_0(\tau)\underline{X}(t) + \Phi_1(\tau)\underline{X}(t - T) + \Phi_2(\tau)\underline{X}(t - 2T) + + \Phi_3(\tau)\underline{X}(t - 3T) + \text{-----}, \tag{3.32}$$

or

$$\underline{X}(t + \tau) = \sum_{i=0}^{\infty} \Phi_i(\tau)\underline{X}(t - iT) \tag{3.33}$$

This is the sampled version of the homogeneous part of the differential-difference equation.

Now, let us consider the addition of an input vector or forcing signal.

In chapter 2, section 2-2, it was found that the digital version of the time-invariant matrix differential equation adopted the form

$$\underline{X}(\overline{K+1}\ T) = \Phi(T)\ \underline{X}(KT) + \Delta(T)\ \underline{U}(KT) \tag{3.34}$$

where

$$\Phi(T) = e^{AT} \tag{3.35}$$

and

$$\Delta(T) = (e^{AT} - I)\ A^{-1}\ D \tag{3.36}$$

Although it was not demonstrated, it can be shown that

$$\Delta(T) = \sum_{j=0}^{\infty} \frac{(AT)^j}{(j+1)!}\ T\ D \quad , \tag{3.37}$$

or

$$\Delta(T) = \sum_{j=0}^{\infty} \frac{(AT)^j}{j!} \frac{1}{j+1}\ T\ D\ . \tag{3.38}$$

If the terms $\frac{1}{j+1}$ and TD were absent, the series would be the well known matrix exponential, whose terms can be computed in a recursive way by

$$C_{o,j} = \frac{(AT)}{j+1} \tag{3.39}$$

Therefore, eq. (3.38) is

$$\Delta(T) = \sum_{j=0}^{\infty} C_{o,j} \frac{T}{j+1} D \tag{3.40}$$

By following the same line of reasoning, the control transition matrices in the case of the complete differential-difference equation can be written as

$$\Delta_i(\tau) = \sum_{j=i}^{\infty} C_{i,j} \frac{\tau}{j+1} D_1 + \sum_{j=i}^{\infty} C_{i-1,j} \frac{\tau}{j+1} D_2 \tag{3.41}$$

and the complete difference equation is

$$\underline{X}(t + \tau) = \sum_{i=0}^{\infty} [\Phi_i(\tau) \underline{X}(t - iT) + \Delta_i(\tau) \underline{U}(t - iT)] \tag{3.42}$$

In resume, the digital version of

$$\underline{X}(t) = A \underline{X}(t) + B \underline{X}(t - T) + D_1\underline{U}(t) + D_2\underline{U}(t - T)$$

is

$$\underline{X}(t + \tau) = \sum_{i=0}^{\infty} [\Phi_i(\tau) \underline{X}(t - iN\tau) + \Delta_i(\tau) \underline{U}(t - iN\tau)]$$

where

$$N = \frac{T}{\tau}$$

$$\Phi_i(\tau) = \sum_{j=0}^{\infty} C_{i,j}$$

$$C_{i,j+1} = \frac{[A\tau] C_{i,j} + [B\tau] C_{i-1,j}}{j+1}$$

$$C_{o,o} = I$$

CHAPTER 4

ALGORITHMS FOR DIGITAL COMPUTATION

This chapter presents flowcharts for the algorithms of chapters 2 and 3, from which the computer programs were derived. They accept as input the coefficient matrices, the driving matrices, the initial state vector, and deterministic forcing vectors. As output, the computer will produce the state vector at the current sampling time and the set of transition matrices, if desired.

Because these routines will eventually become part of Project ENPORT, they were designed to be used on the time-sharing system. However, they may be operated in the BATCH procedure without any difficulty, by modifying the input/output statements.

The programs were written in the MAD language, and are listed in Appendix A.

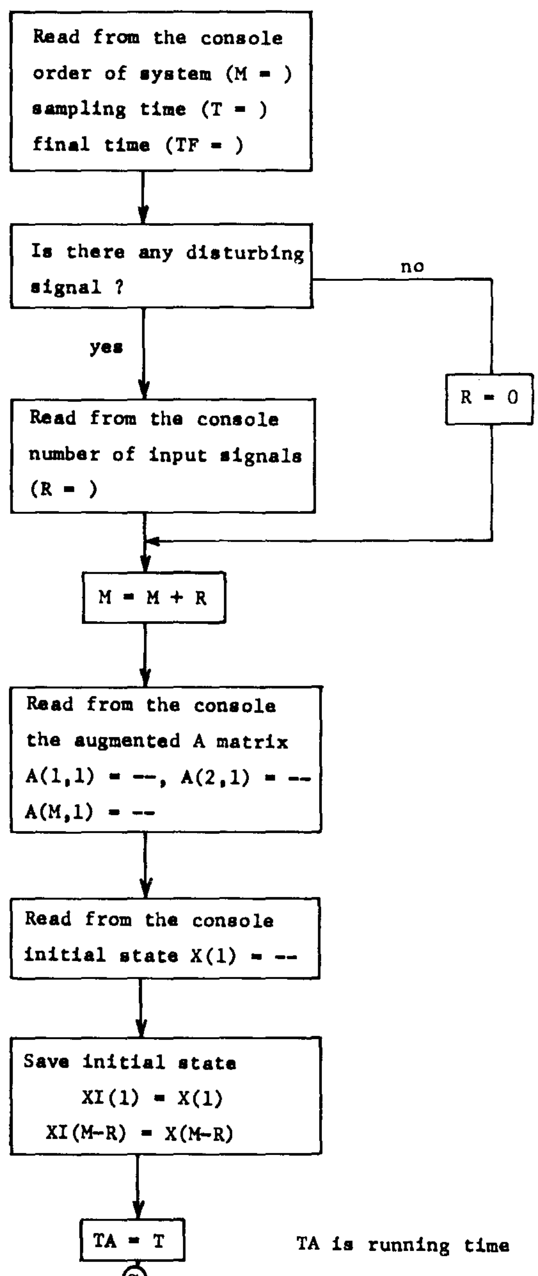
4-1-1 TRANS

Purpose: to compute the time response of linear time-invariant systems.

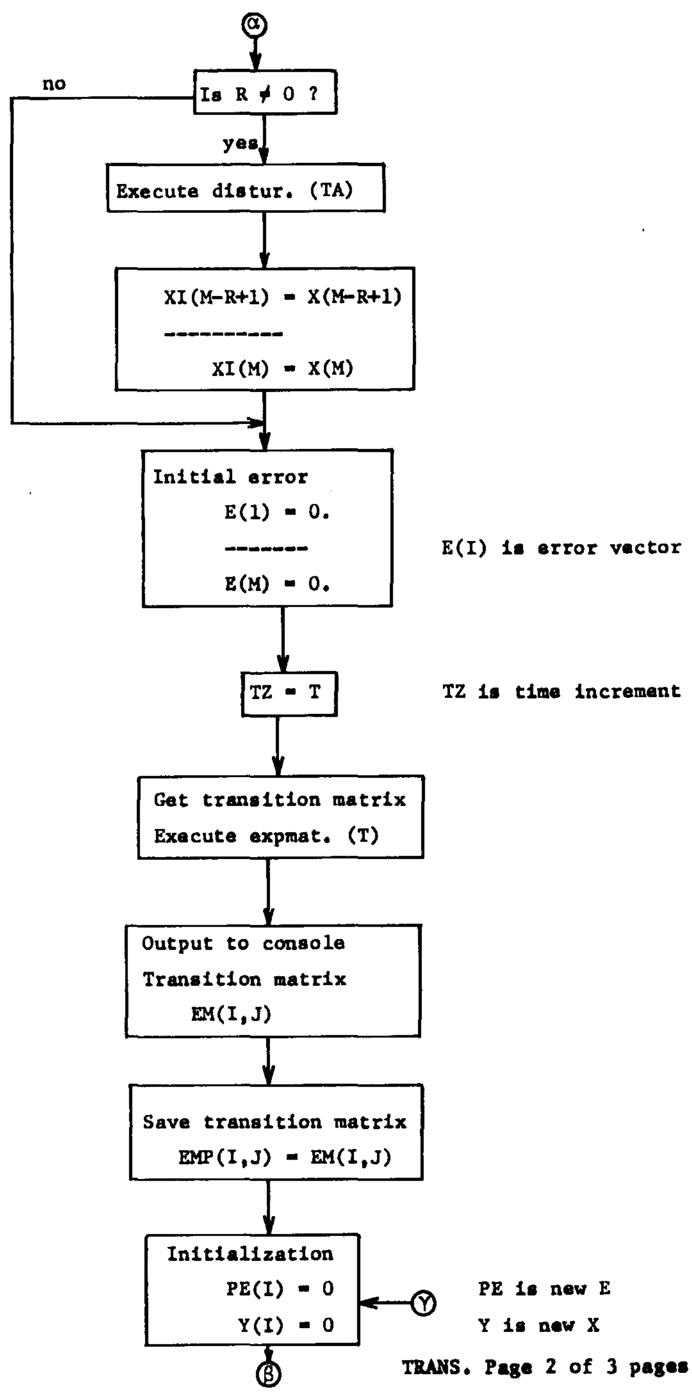
Inputs: order of system (M =); sampling time (T =); final time (TF =); number of input signals (R =); the augmented A matrix and the initial state (x(1) =).

Outputs: the transition matrix; the current time; and the state of the system.

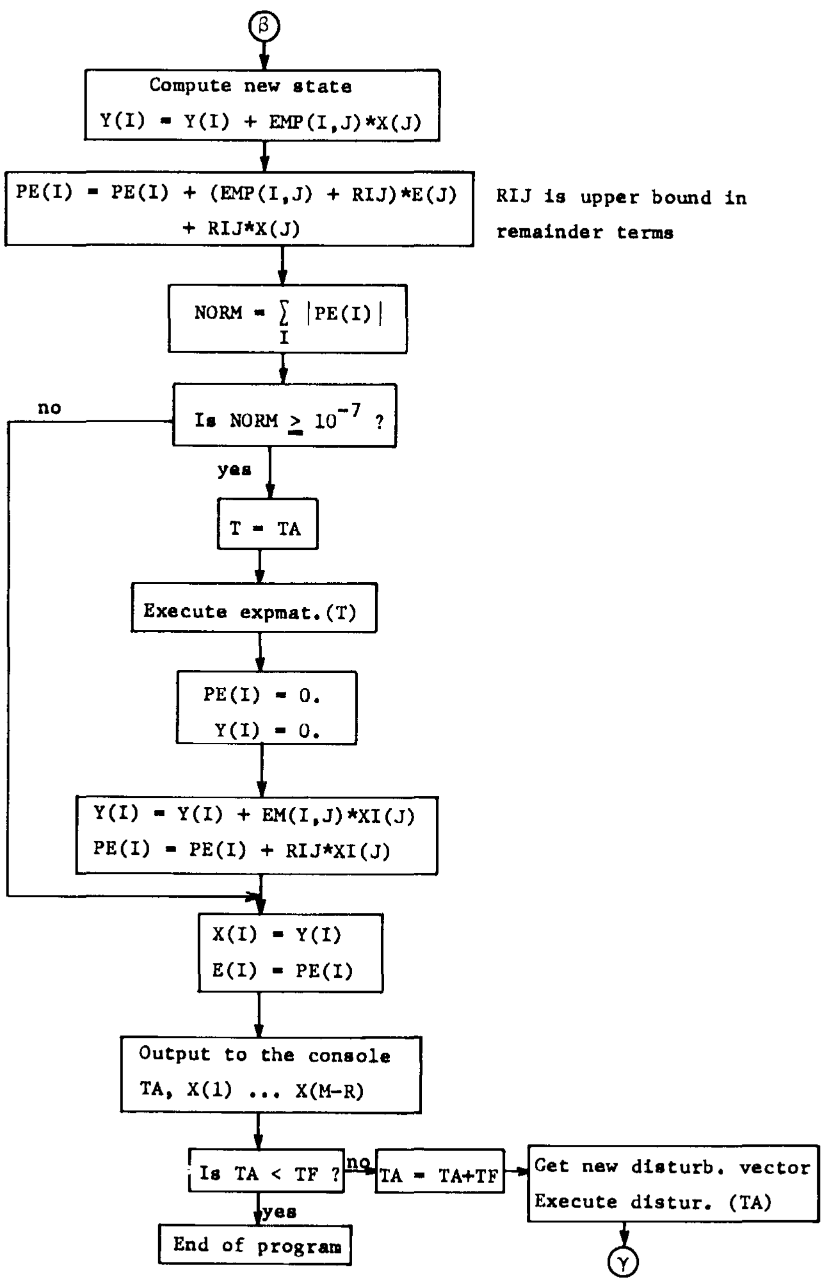
Remarks: main program. Subroutines called by TRANS: EXPMAT, and DISTUR.



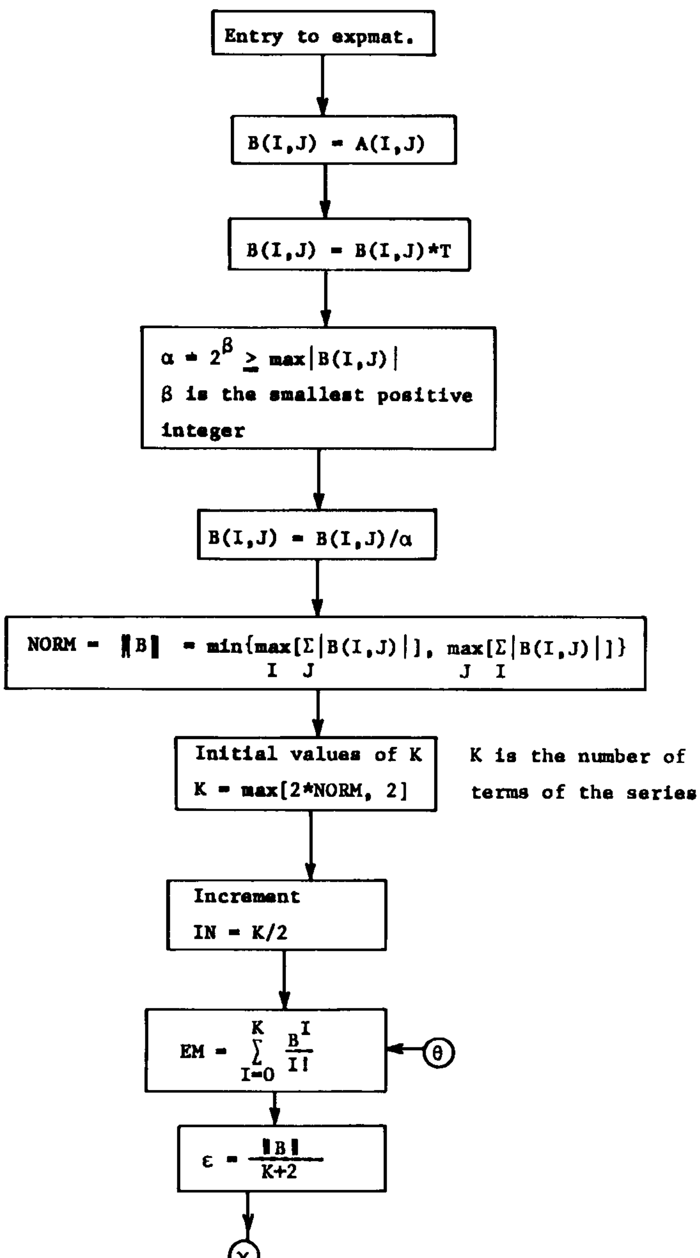
TRANS. Page 1 of 3 pages.



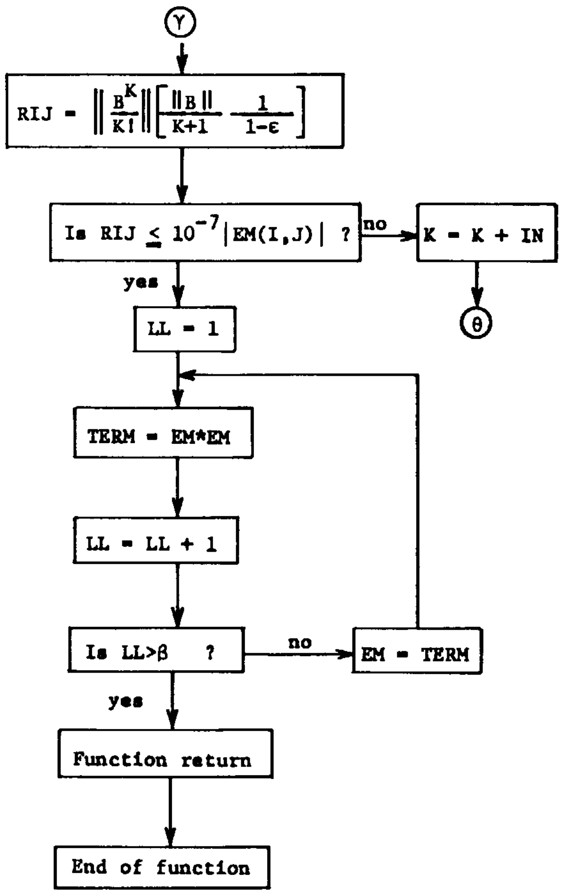
TRANS. Page 2 of 3 pages



TRANS. Page 3 of 3 pages.



EXPMAT. Page 1 of 2 pages.


Y
RIJ = $\left\| \frac{B^K}{K!} \right\| \left[\frac{\|B\|}{K+1} \cdot \frac{1}{1-\varepsilon} \right]$
Is RIJ $\leq 10^{-7}$ |EM(I,J)| ?
no
K = K + IN
θ
yes
LL = 1
TERM = EM*EM
LL = LL + 1
Is LL>β ?
no
EM = TERM
yes
Function return
End of function

4-1-3 DISTUR

Purpose: To compute the forcing signal vector at the current time

Remarks: Subroutine called by TRANS.

4-1-4 EXPMAT

Purpose: To compute the matrix exponential.

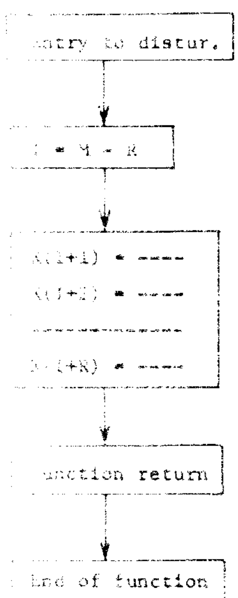
Remarks: Subroutine called by TRANS.

$$C_{i,0} = [illegible]$$

$$[illegible]_{i,j} = [illegible]$$

and

$$[illegible] \frac{[illegible]}{j+1} [illegible] + \sum_{j=1} C_{i-1,j} \frac{[illegible]}{j+1} [illegible]$$


ntry to distur.
= M - R
(I+1) = ----
(I+2) = ----
(I+R) = ----
unction return
of function

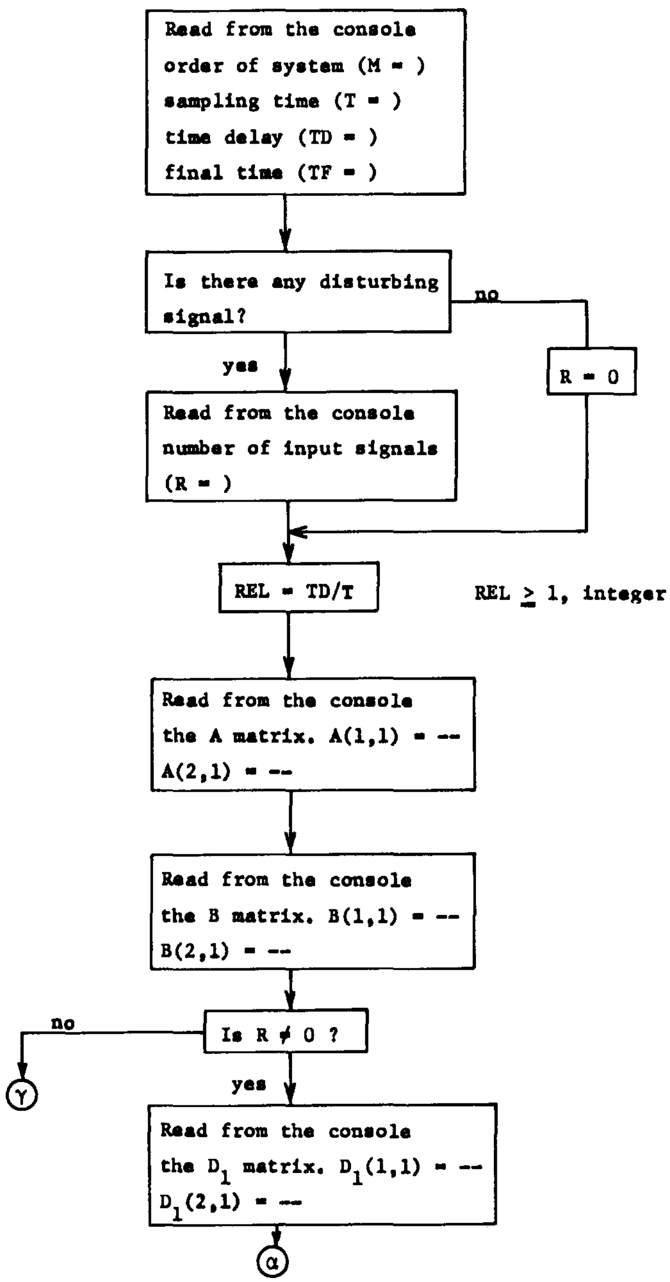
4-2-1 TIMDEL

Purpose: to compute the time response of linear systems with lumped parameters and time delays.

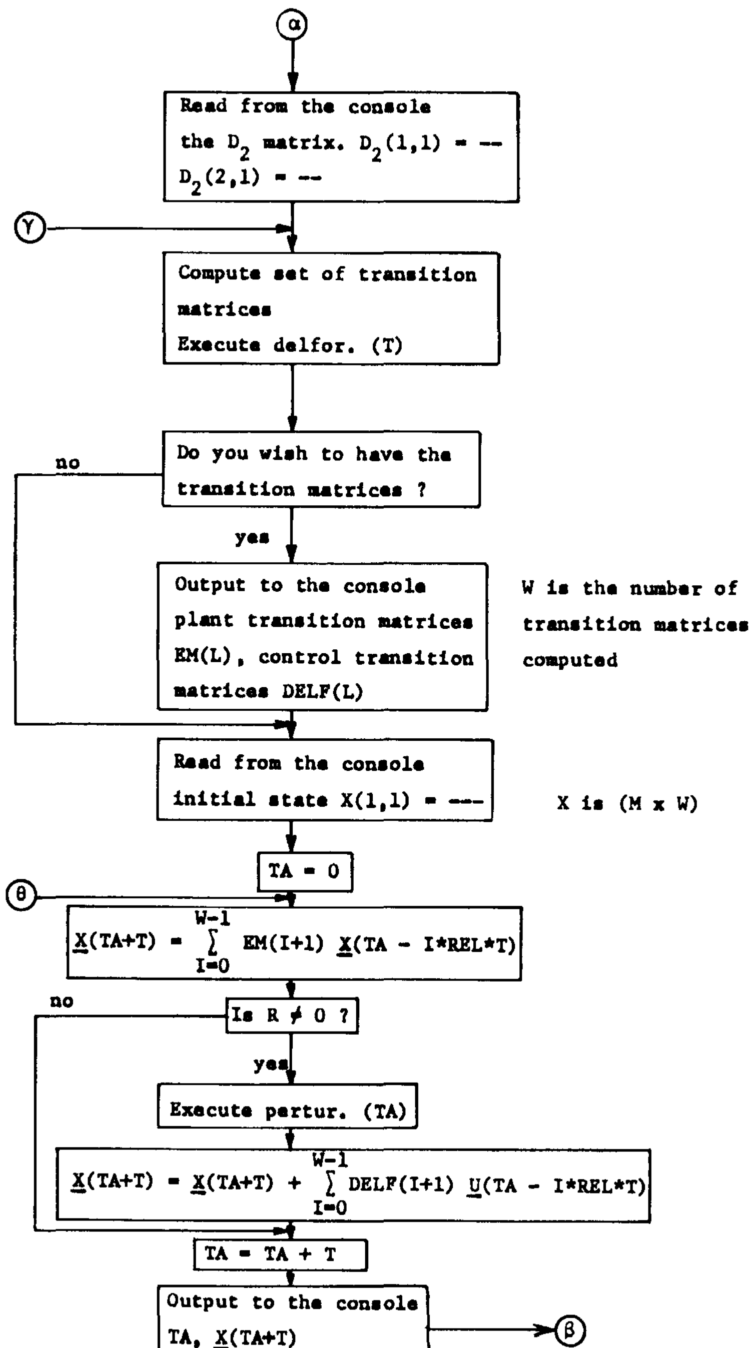
Inputs: order of system (M =); sampling time (T =); time delay (TD =); final time (TF =); number of input signals (R =); the A matrix; the B matrix; the D_1 matrix; the D_2 matrix; the initial state (X(1,1) =).

Outputs: the plant transition matrices, the control transition matrices if desired; the current time; and the state of the system.

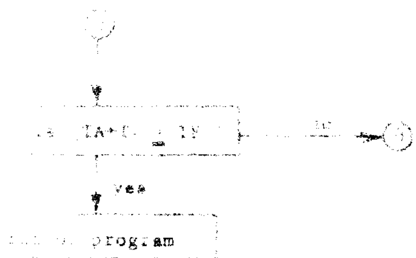
Remarks: main program. Subroutines called by TIMDEL: DELFOR, and PERTUR.



TIMDEL. Page 1 of 3 pages.



TIMDEL. Page 2 of 3 pages.

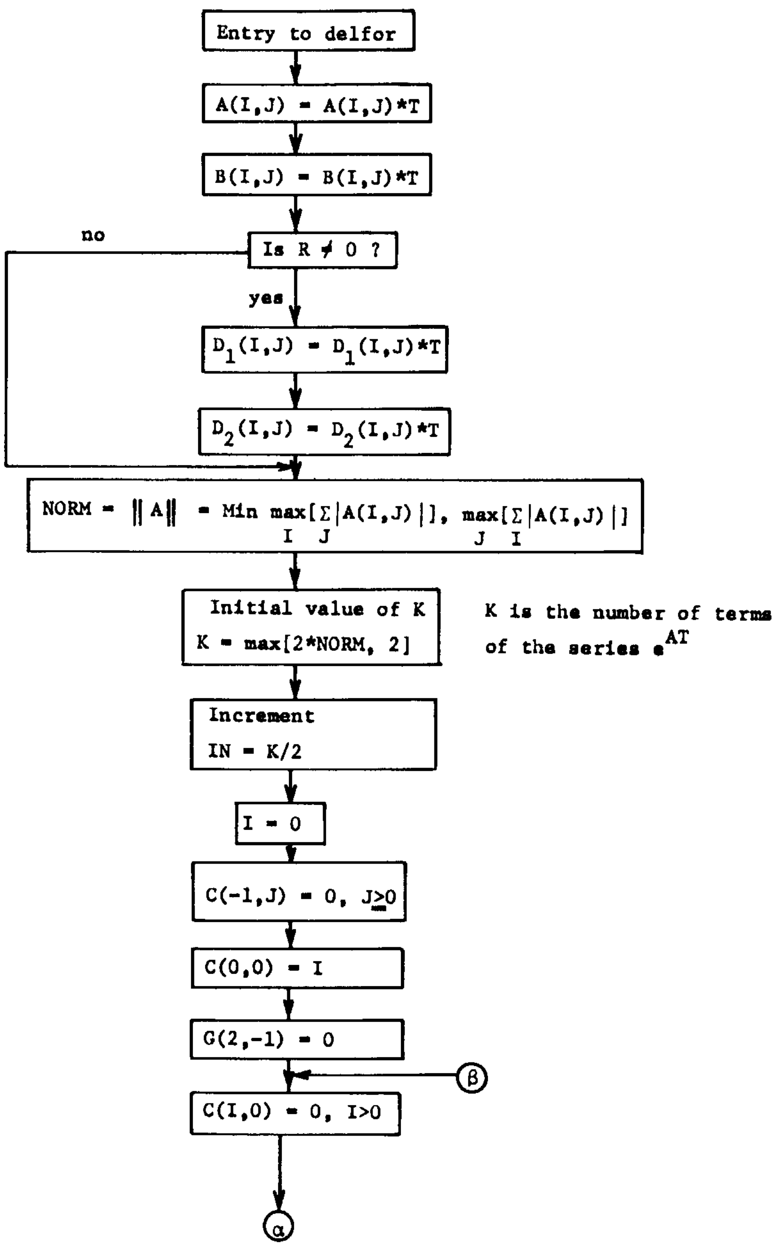

yes
program

DELFOR

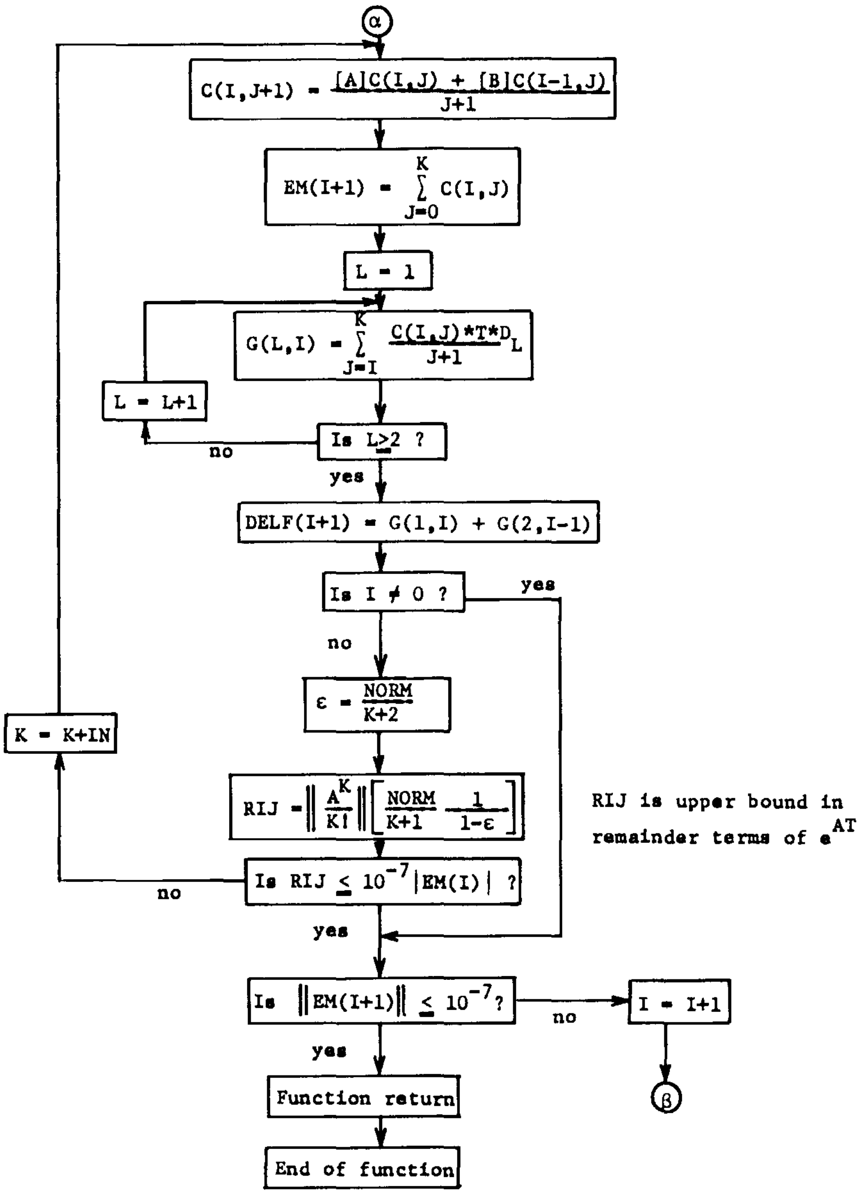
ompute the plant transition matrices and the

rol transition matrices.

outine called by TIMDEL.



DELFOR. Page 1 of 2 pages.

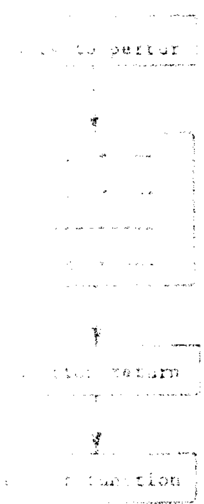


DELFOR. Page 2 of 2 pages.

4-2-3 PERTUR

Purpose: To compute the forcing signal vector at the current time. The program has to keep track of the past.

Remarks: subroutine called by TIMDEL.



page.

CHAPTER 5

SOLUTION TO SAMPLE PROBLEMS

This chapter describes a set of sample problems which were selected because they represent typical applications of the two simulators. They are intended to show the use of the state variable diagram, and also to show the accuracy of the methods.

5-1 Test problem for the simulation of dynamic systems without delay

Example 5-1. Although this example may represent a great number of physical processes, it was selected purely from the mathematical point of view. The same problem was run by Liou (11).

Given

$$\dot{\underline{X}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -.75 & -2.75 & -3 \end{bmatrix} \underline{X}(t) \tag{5.1}$$

and

$$\underline{X}(0) = \begin{bmatrix} 2 \\ -2.5 \\ 3.75 \end{bmatrix} \tag{5.2}$$

Obtain $\underline{X}(nT)$ using $T = 0.1$ Min.

The reported solution by Liou and the one obtained by the simulator are

[redacted]

```
loadgo trans expmat distur
W 1010.4
EXECUTION.
GIVE ORDER OF SYSTEM (M = )
SAMPLING TIME (T = ), FINAL TIME (TF = )
m=3,t=.1,tf=2.*

IS THERE ANY DISTURBING SIGNAL
no

GIVE THE A MATRIX (A(1,1)=--,A(2,1)=--)
a(1,1)=0.,1.,0.*
a(2,1)=0.,0.,1.*
a(3,1)=-.75,-2.75,-3.*

GIVE INITIAL STATE (X(1)=--)
x(1)=2.,-2.5,3.75*
```

$$\dot{\underline{V}}(t) = A\,\underline{V}(t)$$

$$\underline{V}(t) = \begin{bmatrix} X(1) \\ X(2) \\ X(3) \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -.75 & -2.75 & -3 \end{bmatrix}$$

$$\underline{V}(0) = \begin{bmatrix} 2 \\ -2.5 \\ 3.75 \end{bmatrix}$$

```
 TERMS OF THE MATRIX EXPONENTIAL
          EM(   1,    1) =    .999884E 00
          EM(   1,    2) =    .995717E-01
          EM(   1,    3) =    .452513E-02
          EM(   2,    1) =   -.339385E-02
          EM(   2,    2) =    .987440E 00
          EM(   2,    3) =    .859963E-01
          EM(   3,    1) =   -.644972E-01
          EM(   3,    2) =   -.239884E 00
          EM(   3,    3) =    .729451E 00
```

TIME	X(1)	X(2)	X(3)
.10	.176781E 01	-.215290E 01	.320616E 01
.20	.156774E 01	-.185614E 01	.274116E 01
.30	.139515E 01	-.160242E 01	.234368E 01
.40	.124603E 01	-.138548E 01	.200401E 01
.50	.111700E 01	-.119997E 01	.171382E 01
.60	.100515E 01	-.104131E 01	.146596E 01
.70	.907978E 00	-.905571E 00	.125431E 01
.80	.823379E 00	-.789413E 00	.107362E 01
.90	.749538E 00	-.689964E 00	.919418E 00
1.00	.684911E 00	-.604775E 00	.787838E 00
1.10	.628178E 00	-.531753E 00	.675590E 00
1.20	.578215E 00	-.469107E 00	.579853E 00
1.30	.534062E 00	-.415312E 00	.498212E 00
1.40	.494901E 00	-.369064E 00	.428602E 00
1.50	.460034E 00	-.329250E 00	.369257E 00
1.60	.428868E 00	-.294921E 00	.318666E 00
1.70	.400894E 00	-.265268E 00	.275537E 00
1.80	.375681E 00	-.239602E 00	.238768E 00
1.90	.352861E 00	-.217334E 00	.207415E 00
2.00	.332118E 00	-.197965E 00	.180676E 00
2.10	.313185E 00	-.181068E 00	.157862E 00

```
END OF EXECUTION
TO CONTINUE, GO TO THE TOP OF A NEW PAGE
AND PRINT AN ASTERISK
```

Figure 5.1 Console transaction for example 5.1

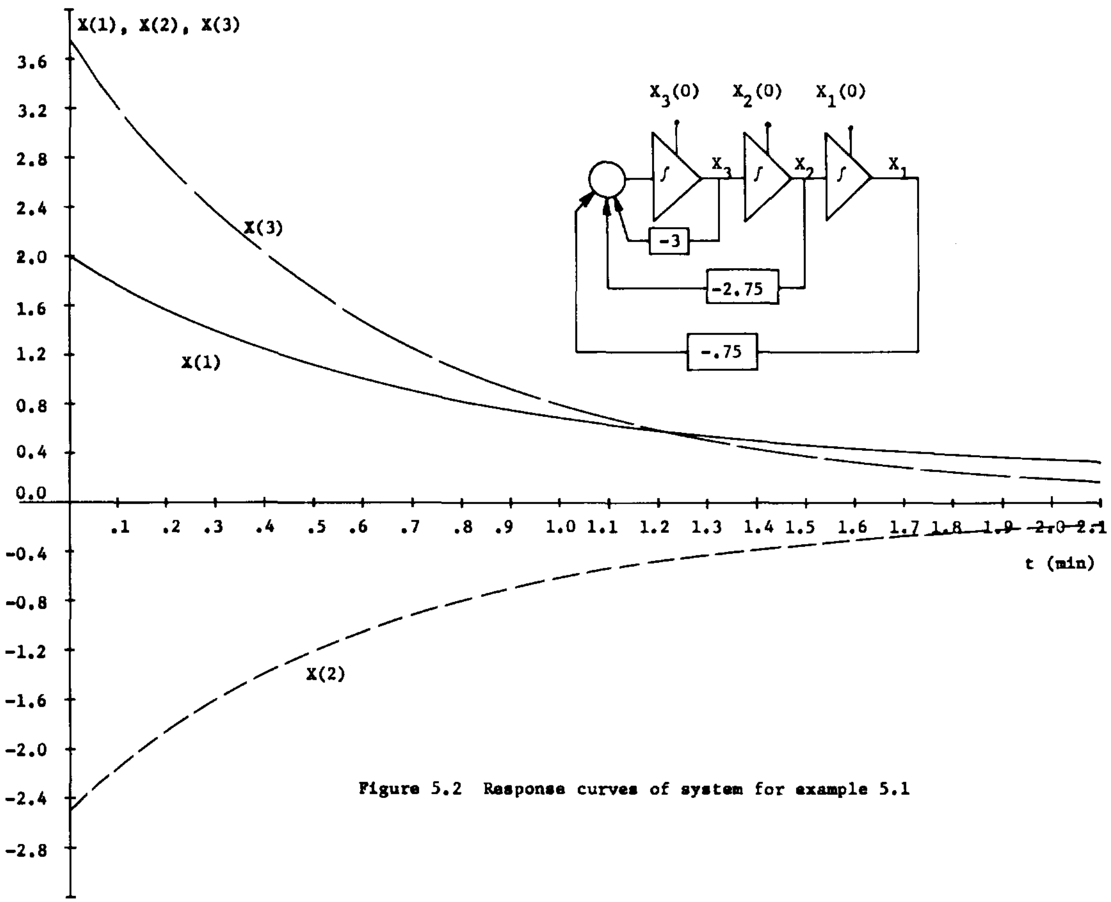


Figure 5.2 Response curves of system for example 5.1

A liquid stream enters tank 1 (figure 5.3) at a volumetric flow rate F cfm and contains reactant A at a concentration of C_o moles A/ft^3. Reactant A decomposes in the tanks according to the irreversible chemical reaction.

$$A \longrightarrow B$$

The reaction is first order and proceeds at a rate

$$r = k\ c$$

where

r = moles A decomposing/(ft^3)(time)

c = concentration of A, moles A/ft^3

k = velocity constant, a function of temperature

The reaction is to be carried out in a series of two stirred tanks. The tanks are maintained at different temperatures. The temperature in tank 2 is to be greater than the temperature in tank 1, with the result that k_2, the velocity constant in tank 2, is greater than in tank 1, k_1. Changes in physical properties due to chemical reaction are neglected.

The purpose of the control system is to maintain c_2, the concentration of A leaving tank 2, at some desired value in spite of variation in inlet concentration c_o. This will be accomplished by adding a stream of pure A to tank 1 through a control valve.

Further assumptions are that the control valve and the measuring element have no dynamics, and that the controller exert proportional action on the process.

A portion of the liquid leaving tank 2 is continuously withdrawn through a sample line. The measuring element is remotely located from the process, because rigid ambient conditions must be maintained for accurate concentration measurements. The sample line can be represented by a transportation lag.

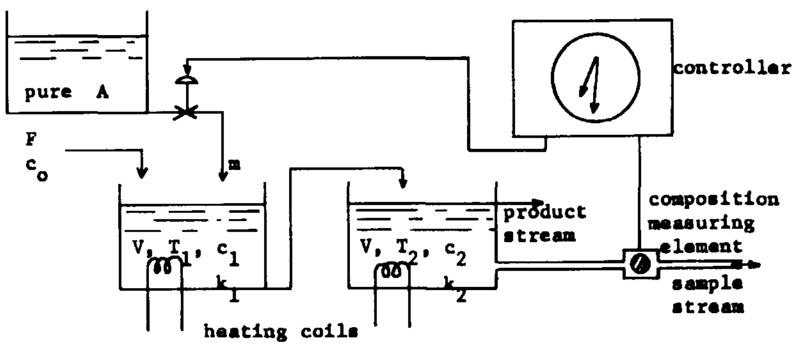


Figure 5.3

Control of a stirred-tank chemical reactor

The following data is assumed to apply to the system

Molecular weight of A = 100 lb/lb mole

$$\rho_A = 0.8 \text{ lb mole/ft}^3$$

$$C_{os} = 0.1 \text{ lb mole A/ft}^3$$

$$F = 100 \text{ cfm}$$

$$m_s = 1.0 \text{ lb mole/min}$$

$$k_1 = 1/6 \text{ min}^{-1}$$

$$k_2 = 2/3 \text{ min}^{-1}$$

$$V = 300 \text{ ft}^3$$

Valve sensitivity k_v = 1/6 cfm/psi

Measuring device sensitivity

$$k_m = 100 \text{ in. pen travel/(lb mole/ft}^3)$$

Time delay in sample line = T

The overall block diagram which the authors propose is

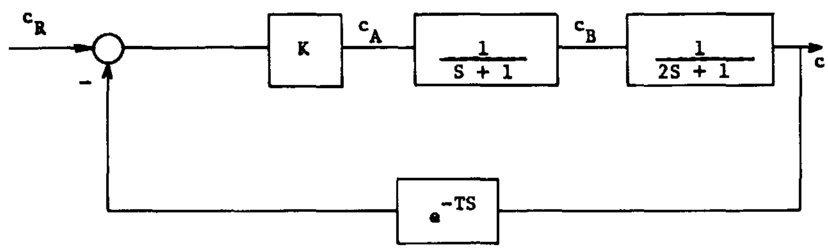


Figure 5.4

Block diagram for a chemical reactor control system

It is assumed that the inlet concentration c_o does not change with time.

As was discussed in chapter 2, the state variable diagram can be obtained in three ways. Direct programming will be used in this case. With this purpose, let it be called c_A the input to the lag term $\frac{1}{S+1}$ and c_B its output in figure 5.4, then

$$\frac{c}{c_B} = \frac{1}{2S+1} , \tag{5.3}$$

or

$$\frac{c}{c_B} = .5 \frac{S^{-1}}{1+.5\ S^{-1}} \ . \tag{5.4}$$

Eq. (5.4) can be written as

$$c = .5\ S^{-1}\ E_b \tag{5.5}$$

where

$$E_b = \frac{c_B}{1+.5S^{-1}} \tag{5.6}$$

Transposing

$$E_b = c_B - .5\ S^{-1}\ E_b \tag{5.7}$$

By a similar procedure

$$\frac{c_B}{c_A} = \frac{1}{S+1} \quad , \tag{5.8}$$

or

$$\frac{c_B}{c_A} = \frac{S^{-1}}{1+S^{-1}} \quad . \tag{5.9}$$

$$c_B = S^{-1} E_a \tag{5.10}$$

where

$$E_a = \frac{c_A}{1+S^{-1}} \tag{5.11}$$

Transposing

$$E_a = c_A - S^{-1} c_A \tag{5.12}$$

The state variable diagram follows from eqs. (5.5), (5.7) and eqs. (5.10), (5.12), and is shown in figure 5.5.

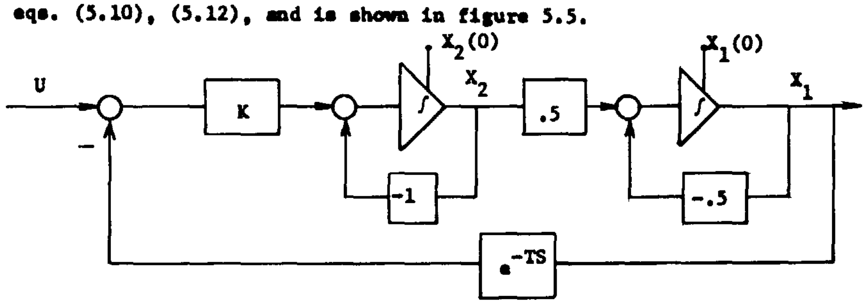


Figure 5.5

State variable diagram for a chemical reactor control system

The notation in figure 5.5 has been changed slightly. This is in order to follow the same symbolism given in the previous chapters.

In figure 5.5 the state variables are X_1 and X_2. The differential-difference equations for the state variables are readly obtained by inspection of the diagram. That is,

$$\dot{X}_1 = -.5\ X_1 + .5\ X_2 \tag{5.13}$$

$$\dot{X}_2 = -\ X_2 + K\ U - K\ X_1(t - T) \tag{5.14}$$

Therefore the matrix differential-difference equation is

$$\underline{\dot{X}}(t) = \begin{bmatrix} -.5 & .5 \\ 0 & -1 \end{bmatrix} \underline{X}(t) + \begin{bmatrix} 0 & 0 \\ -K & 0 \end{bmatrix} \underline{X}(t - T) + \begin{bmatrix} 0 \\ K \end{bmatrix} U(t) \tag{5.15}$$

where

$$\underline{X}(t) = \begin{bmatrix} X_1(t) \\ X_2(t) \end{bmatrix}$$

From this equation, it is seen that the coefficient matrices and driving matrices are

$$A = \begin{bmatrix} -.5 & .5 \\ 0 & -1 \end{bmatrix} \tag{5.16}$$

$$B = \begin{bmatrix} 0 & 0 \\ -K & 0 \end{bmatrix} \tag{5.17}$$

$$D_1 = \begin{bmatrix} 0 \\ K \end{bmatrix} \tag{5.18}$$

$$D_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5.19}$$

Five numerical examples were run using this system. These are summarized as follows.

Example 5.2.1. Overall forward gain K= 5.24. We assume a time delay equal to zero. A unit step is the input and all initial conditions are zero. The matrix differential equation is

$$\underline{\dot{V}}(t) = \begin{bmatrix} -.5 & .5 & 0 \\ -5.24 & -1 & 5.24 \\ 0 & 0 & 0 \end{bmatrix} \underline{V}(t) \tag{5.20}$$

where

$$\underline{V}(t) = \begin{bmatrix} X_1(t) \\ X_2(t) \\ U(t) \end{bmatrix} \tag{5.21}$$

Example 5.2.2. Overall forward gain K = 5.24, and time delay = .5 Min.. Same conditions of the state were taken. The matrix differential-difference equation is

$$\underline{\dot{X}}(t) = \begin{bmatrix} -.5 & .5 \\ 0 & -1 \end{bmatrix} \underline{X}(t) + \begin{bmatrix} 0 & 0 \\ -5.24 & 0 \end{bmatrix} \underline{X}(t - .5) + \begin{bmatrix} 0 \\ 5.24 \end{bmatrix} U(t) + \begin{bmatrix} 0 \\ 0 \end{bmatrix} U(t - .5) \tag{5.22}$$

Example 5.2.3. Overall forward gain K = 1.85. Time delay is zero. The remaining conditions are the same. The state equation is

$$\underline{\dot{V}}(t) = \begin{bmatrix} -.5 & .5 & 0 \\ -1.85 & -1 & 1.85 \\ 0 & 0 & 0 \end{bmatrix} \underline{V}(t) \tag{5.23}$$

Example 5.2.4. Overall forward gain K = 1.85. Time delay = .5 min. Unit step and zero initial conditions are assumed. The state equation is

$$\dot{\underline{X}}(t) = \begin{bmatrix} -.5 & .5 \\ 0 & -1 \end{bmatrix} \underline{X}(t) + \begin{bmatrix} 0 & 0 \\ -1.85 & 0 \end{bmatrix} \underline{X}(t - .5) + \begin{bmatrix} 0 \\ 1.85 \end{bmatrix} U(t) + \begin{bmatrix} 0 \\ 0 \end{bmatrix} U(t - .5) \qquad (5.24)$$

Example 5.2.5. This is the same as example 5.2.4, with the exception of the time delay, which is taken equal to 1 min.. The state equation is

$$\dot{\underline{X}}(t) = \begin{bmatrix} -.5 & .5 \\ 0 & -1 \end{bmatrix} \underline{X}(t) + \begin{bmatrix} 0 & 0 \\ -1.85 & 0 \end{bmatrix} \underline{X}(t - 1) + \begin{bmatrix} 0 \\ 1.85 \end{bmatrix} U(t) + \begin{bmatrix} 0 \\ 0 \end{bmatrix} U(t - 1) \qquad (5.25)$$

All five examples with the input/output information and the response curves, are shown in figures 5.6 to 5.15.

The interested reader should compare the responses of the three cases with delay with those given by Coughanowr and Koppel on page 467 of reference (4).

5-3 Test problem for the simulation of dynamic systems with delays

The eighth example was run in order to check the accuracy of evaluation of the set of transition matrices. This example is discussed by Koepcke (9).

The problem is described as an unstable process which is governed by

```
*
GIVE ORDER OF SYSTEM (M = )
SAMPLING TIME (T = ), FINAL TIME (TF = )
m=2,t=.5,tf=11.*

IS THERE ANY DISTURBING SIGNAL
yes

GIVE NUMBER OF INPUT SIGNALS (R = )
r=1*

GIVE THE A MATRIX (A(1,1)=--,A(2,1)=--)
a(1,1)=-.5,.5,0.*
a(2,1)=-5.24,-1.,5.24*
a(3,1)=0.,0.,0.*

GIVE INITIAL STATE (X(1)=--)
x(1)=0.,0.*

 TERMS OF THE MATRIX EXPONENTIAL
          EM(    1,     1) =    .556076E 00
          EM(    1,     2) =    .154089E 00
          EM(    1,     3) =    .243387E 00
          EM(    2,     1) =  -.161485E 01
          EM(    2,     2) =    .401987E 00
          EM(    2,     3) =    .185824E 01
          EM(    3,     1) =    .000000E 00
          EM(    3,     2) =    .000000E 00
          EM(    3,     3) =  1.000000E 00

      TIME        X(1)             X(2)

       .50      .243387E 00      .185824E 01
      1.00      .665063E 00      .221219E 01
      1.50      .954087E 00      .167353E 01
      2.00      .103181E 01      .990269E 00
      2.50      .969739E 00      .590102E 00
      3.00      .873563E 00      .529468E 00
      3.50      .810740E 00      .660403E 00
      4.00      .795981E 00      .814488E 00
      4.50      .811516E 00      .900262E 00
      5.00      .833372E 00      .909654E 00
      5.50      .846973E 00      .878136E 00
      6.00      .849679E 00      .843503E 00
      6.50      .845848E 00      .825210E 00
      7.00      .840898E 00      .824044E 00
      7.50      .837966E 00      .831568E 00
      8.00      .837495E 00      .839327E 00
      8.50      .838429E 00      .843206E 00
      9.00      .839546E 00      .843258E 00
      9.50      .840175E 00      .841475E 00
     10.00      .840250E 00      .839743E 00
     10.50      .840025E 00      .838925E 00
     11.00      .839774E 00      .838960E 00
END OF EXECUTION
TO CONTINUE, GO TO THE TOP OF A NEW PAGE
AND PRINT AN ASTERISK
```

$$\underline{V}(t) = A\,\underline{V}(t)$$

$$\underline{V}(t) = \begin{bmatrix} X(1) \\ X(2) \\ U(1) \end{bmatrix}$$

$$A = \begin{bmatrix} -.5 & .5 & 0 \\ -5.24 & -1 & 5.24 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\underline{X}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Figure 5.6 Console transaction for example 5.2.1

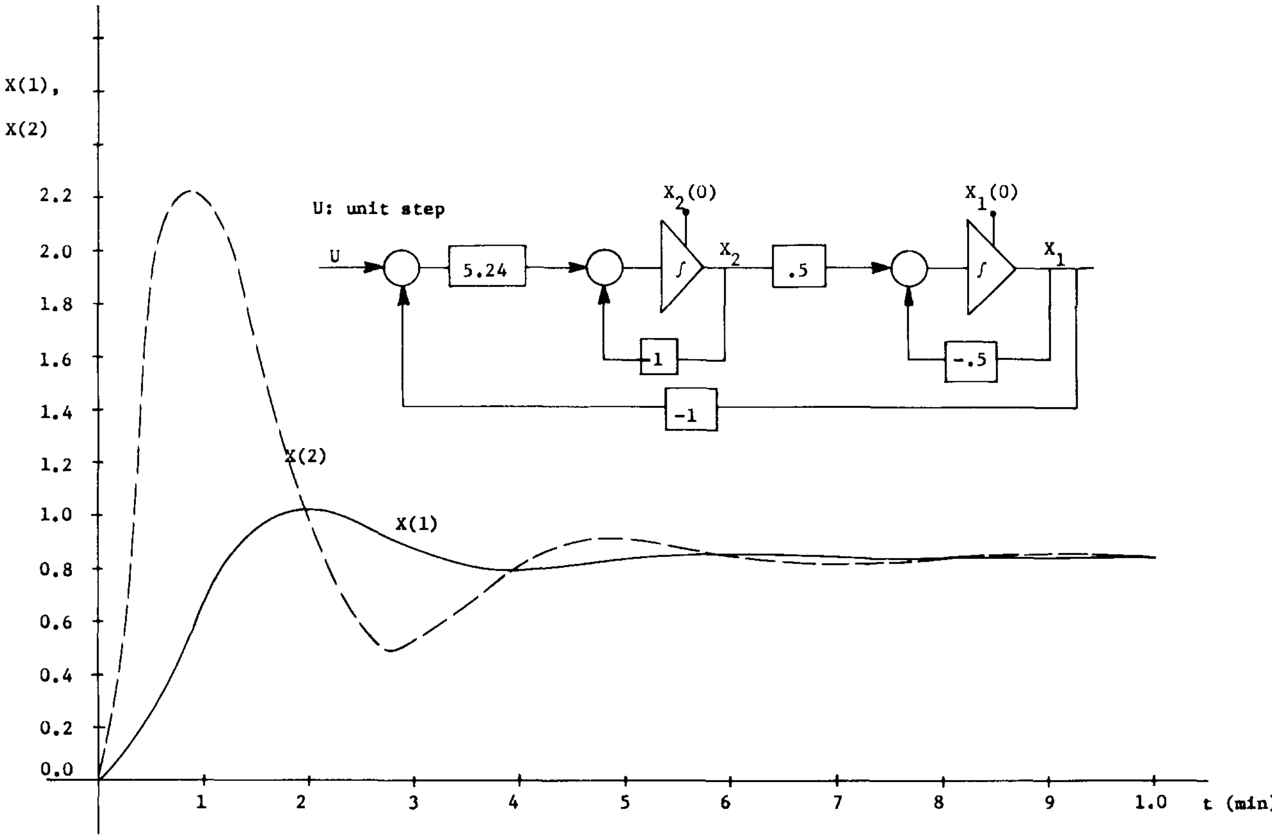


Figure 5.7 Response curves of system for example 5.2.1

```
loadgo timdel delfor pertur
W 1039.4
EXECUTION.
GIVE ORDER OF SYSTEM (M = )
DESIRED SAMPLING TIME (T = )
TIME DELAY (TD = ), FINAL TIME (TF = )
m=2,t=.5,td=.5,tf=15.*

IS THERE ANY DISTURBING SIGNAL
yes

NUMBER OF INPUT SIGNALS (R = )
r=1*

GIVE THE A MATRIX (A(1,1)=--,A(2,1)=--)
a(1,1)=-.5,.5*
a(2,1)=0.,-1.*

GIVE THE B MATRIX (B(1,1)=--,B(2,1)=--)
b(1,1)=0.,0.*
b(2,1)=-5.24,0.*

GIVE THE D1 MATRIX (D1(1,1)=--,D1(2,1)=--)
d1(1,1)=0.*
d1(2,1)=5.24*

GIVE THE D2 MATRIX (D2(1,1)=--,D2(2,1)=--)
d2(1,1)=0.*
d2(2,1)=0.*
DO YOU WISH TO HAVE THE TRANSITION MATRICES
yes
```

$$\dot{\underline{X}}(t) = A\,\underline{X}(t) + B\,X(t-.5) + D_1U(t) + D_2U(t-.5)$$

$$A = \begin{bmatrix} -.5 & .5 \\ 0 & -1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ -5.24 & 0 \end{bmatrix}$$

$$D_1 = \begin{bmatrix} 0 \\ 5.24 \end{bmatrix}$$

$$D_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

```
         TRANSFER MATRIX  PHI(    0)        TRANSFER MATRIX DELTA(    0)
         EM(   1,    1) =   .778801E 00     DEL(    1,    1) =   .256388E 00
         EM(   1,    2) =   .172270E 00
         EM(   2,    1) =   .000000E 00     DEL(    2,    1) =   .206178E 01
         EM(   2,    2) =   .606531E 00
         TRANSFER MATRIX  PHI(    1)        TRANSFER MATRIX DELTA(    1)
         EM(   1,    1) =  -.235067E 00     DEL(    1,    1) =  -.132818E-01
         EM(   1,    2) =  -.187866E-01
         EM(   2,    1) =  -.180539E 01     DEL(    2,    1) =  -.210165E 00
         EM(   2,    2) =  -.216281E 00
         TRANSFER MATRIX  PHI(    2)        TRANSFER MATRIX DELTA(    2)
         EM(   1,    1) =   .126127E-01     DEL(    1,    1) =   .283552E-03
         EM(   1,    2) =   .614986E-03
         EM(   2,    1) =   .196883E 00     DEL(    2,    1) =   .672861E-02
         EM(   2,    2) =   .119977E-01
         TRANSFER MATRIX  PHI(    3)        TRANSFER MATRIX DELTA(    3)
         EM(   1,    1) =  -.273338E-03     DEL(    1,    1) =  -.327556E-05
         EM(   1,    2) =  -.958848E-05
         EM(   2,    1) =  -.644506E-02     DEL(    2,    1) =  -.103763E-03
         EM(   2,    2) =  -.263750E-03
         TRANSFER MATRIX  PHI(    4)        TRANSFER MATRIX DELTA(    4)
         EM(   1,    1) =   .318384E-05     DEL(    1,    1) =   .236515E-07
         EM(   1,    2) =   .872148E-07
         EM(   2,    1) =   .100487E-03     DEL(    2,    1) =   .937662E-06
         EM(   2,    2) =   .309662E-05
         TRANSFER MATRIX  PHI(    5)        TRANSFER MATRIX DELTA(    5)
         EM(   1,    1) =  -.231099E-07     DEL(    1,    1) =  -.116723E-09
         EM(   1,    2) =  -.519268E-09
         EM(   2,    1) =  -.914011E-06     DEL(    2,    1) =  -.555865E-08
         EM(   2,    2) =  -.225906E-07
```

```
          TRANSFER MATRIX  PHI(   6)         TRANSFER MATRIX DELTA(   6)
          EM(   1,    1) =   .114463E-09     DEL(   1,    1) =   .418403E-12
          EM(   1,    2) =   .218008E-11
          EM(   2,    1) =   .544192E-08     DEL(   2,    1) =   .232657E-10
          EM(   2,    2) =   .112283E-09

GIVE THE INITIAL STATE (X(1,1)=---)
x(1,1)=0.,0.*
      TIME        X(1)          X(2)

       .50     .2564E 00     .2062E 01
      1.00     .7980E 00     .3102E 01
      1.50     .1300E 01     .2831E 01
      2.00     .1502E 01     .1539E 01
      2.50     .1332E 01     .2406E-01
      3.00     .9204E 00    -.8884E 00
      3.50     .5132E 00    -.7847E 00
      4.00     .3246E 00     .1653E 00
      4.50     .4295E 00     .1364E 01
      5.00     .7391E 00     .2150E 01
      5.50     .1067E 01     .2155E 01
      6.00     .1237E 01     .1465E 01
      6.50     .1179E 01     .5227E 00
      7.00     .9475E 00    -.1450E 00
      7.50     .6857E 00    -.2169E 00
      8.00     .5349E 00     .2774E 00
      8.50     .5622E 00     .1013E 01
      9.00     .7330E 00     .1573E 01
      9.50     .9408E 00     .1683E 01
     10.00     .1072E 01     .1335E 01
     10.50     .1065E 01     .7646E 00
     11.00     .9404E 00     .2988E 00
     11.50     .7766E 00     .1715E 00
     12.00     .6646E 00     .4115E 00
     12.50     .6579E 00     .8506E 00
     13.00     .7480E 00     .1234E 01
     13.50     .8763E 00     .1366E 01
     14.00     .9708E 00     .1205E 01
     14.50     .9854E 00     .8692E 00
     15.00     .9214E 00     .5560E 00
END OF EXECUTION
TO CONTINUE, GO TO THE TOP OF A NEW PAGE
AND PRINT AN ASTERISK
```

Figure 5.8 Console transaction for example 5.2.2

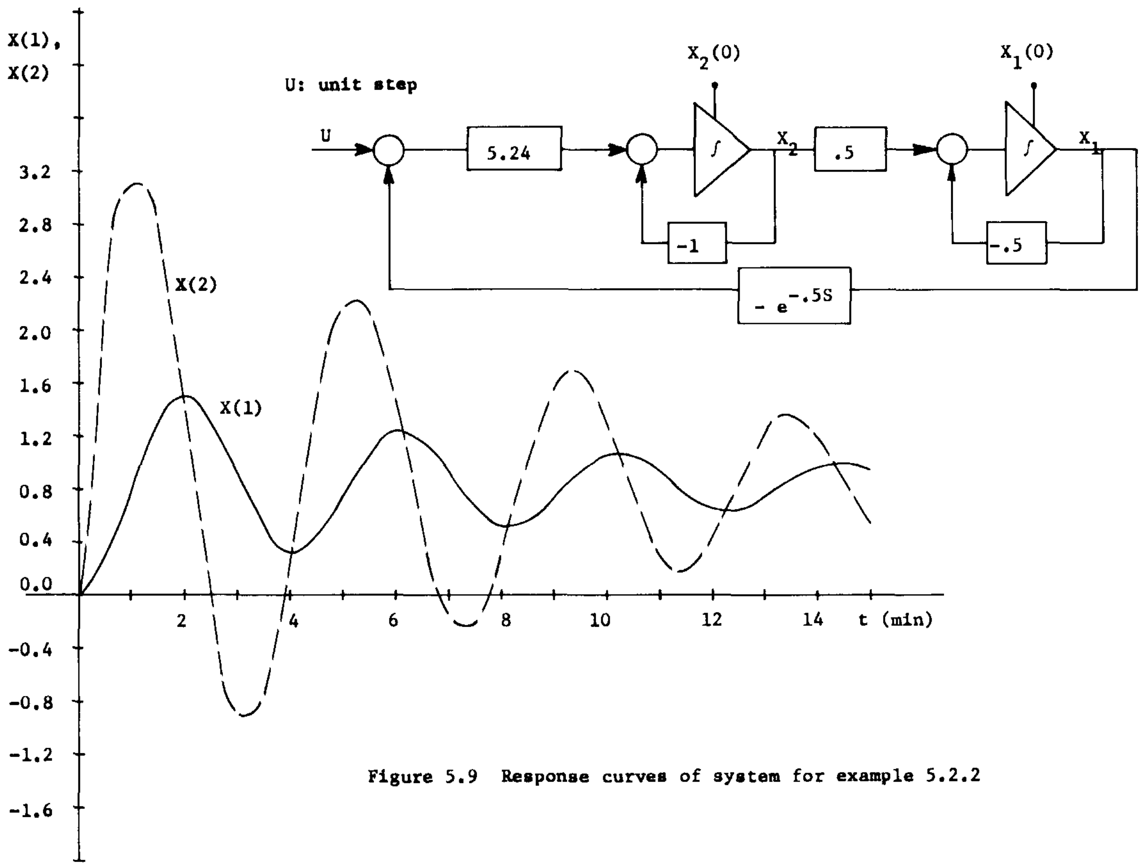


Figure 5.9 Response curves of system for example 5.2.2

```
*
GIVE ORDER OF SYSTEM (M = )
SAMPLING TIME (T = ), FINAL TIME (TF = )
m=2,t=.5,tf=11.*

IS THERE ANY DISTURBING SIGNAL
yes

GIVE NUMBER OF INPUT SIGNALS (R = )
r=1*

GIVE THE A MATRIX (A(1,1)=--,A(2,1)=--)
a(1,1)=-.5,.5,0.*
a(2,1)=-1.85,-1.,1.85*
a(3,1)=0.,0.,0.*

GIVE INITIAL STATE (X(1)=--)
x(1)=0.,0.*
```

$$\dot{\underline{V}}(t) = A\,\underline{V}(t)$$

$$\underline{V}(t) = \begin{bmatrix} X(1) \\ X(2) \\ U(1) \end{bmatrix}$$

$$A = \begin{bmatrix} -.5 & .5 & 0 \\ -1.85 & -1 & 1.85 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\underline{X}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

```
 TERMS OF THE MATRIX EXPONENTIAL
         EM(   1,   1) =   .697370E 00
         EM(   1,   2) =   .165714E 00
         EM(   1,   3) =   .888757E-01
         EM(   2,   1) =  -.613141E 00
         EM(   2,   2) =   .531656E 00
         EM(   2,   3) =   .702016E 00
         EM(   3,   1) =   .000000E 00
         EM(   3,   2) =   .000000E 00
         EM(   3,   3) =  1.000000E 00

      TIME        X(1)            X(2)

       .50      .888757E-01      .702016E 00
      1.00      .267189E 00      .102075E 01
      1.50      .444358E 00      .108088E 01
      2.00      .577874E 00      .100422E 01
      2.50      .658281E 00      .881598E 00
      3.00      .694033E 00      .767104E 00
      3.50      .699993E 00      .684312E 00
      4.00      .690429E 00      .636641E 00
      4.50      .675860E 00      .617160E 00
      5.00      .662472E 00      .615736E 00
      5.50      .652899E 00      .623187E 00
      6.00      .647458E 00      .633019E 00
      6.50      .645293E 00      .641581E 00
      7.00      .645202E 00      .647461E 00
      7.50      .646113E 00      .650643E 00
      8.00      .647276E 00      .651776E 00
      8.50      .648274E 00      .651666E 00
      9.00      .648953E 00      .650995E 00
      9.50      .649314E 00      .650222E 00
     10.00      .649438E 00      .649590E 00
     10.50      .649420E 00      .649178E 00
     11.00      .649339E 00      .648970E 00
END OF EXECUTION
TO CONTINUE, GO TO THE TOP OF A NEW PAGE
AND PRINT AN ASTERISK
```

Figure 5.10 Console transaction for example 5.2.3

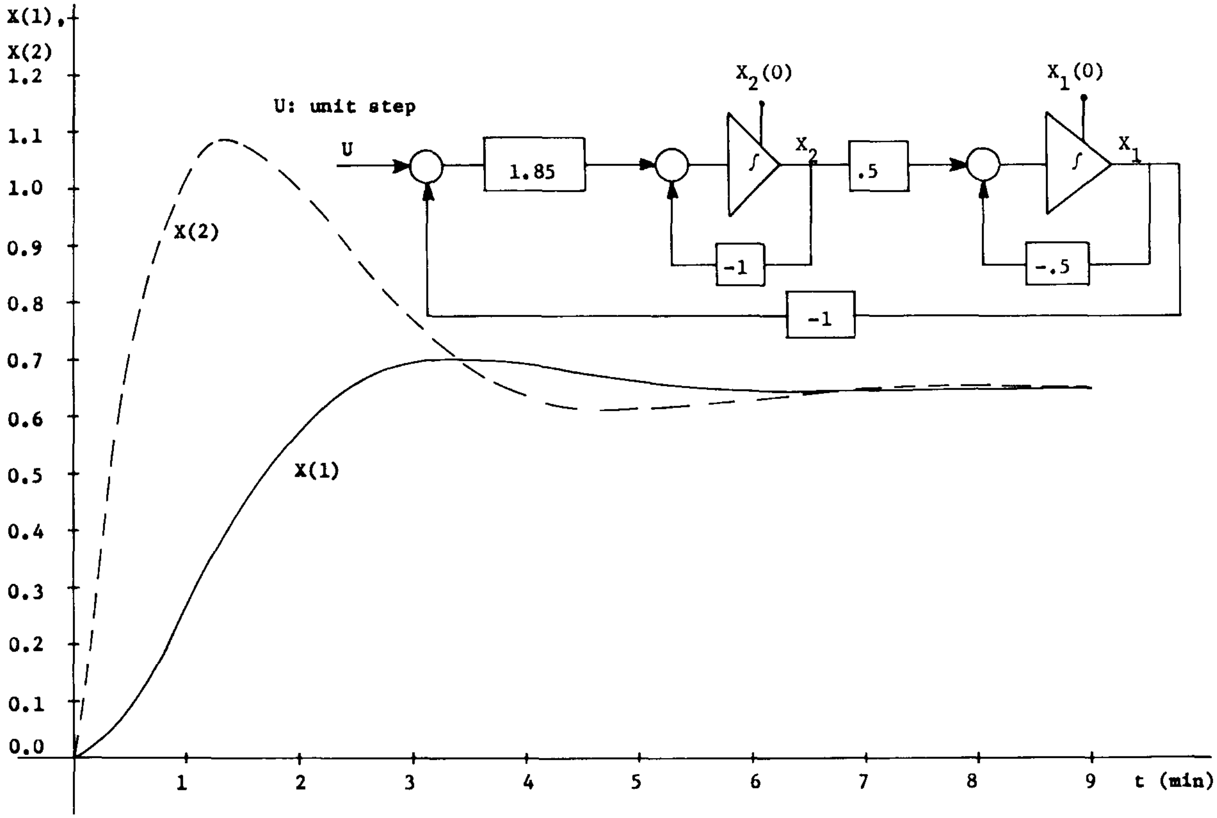


Figure 5.11 Response curves of system for example 5.2.3

```
*
GIVE ORDER OF SYSTEM (M = )
DESIRED SAMPLING TIME (T = )
TIME DELAY (TD = ), FINAL TIME (TF = )
m=2,t=.5,td=.5,tf=15.*

IS THERE ANY DISTURBING SIGNAL
yes

NUMBER OF INPUT SIGNALS (R = )
r=1*

GIVE THE A MATRIX (A(1,1)=--,A(2,1)=--)
a(1,1)=-.5,.5*
a(2,1)=0.,-1.*

GIVE THE B MATRIX (B(1,1)=--,B(2,1)=--)
b(1,1)=0.,0.*
b(2,1)=-1.85,0.*

GIVE THE D1 MATRIX (D1(1,1)=--,D1(2,1)=--)
d1(1,1)=0.*
d1(2,1)=1.85*

GIVE THE D2 MATRIX (D2(1,1)=--,D2(2,1)=--)
d2(1,1)=0.*
d2(2,1)=0.*
DO YOU WISH TO HAVE THE TRANSITION MATRICES
no

GIVE THE INITIAL STATE (X(1,1)=---)
x(1,1)=0.,0.*
```

TIME	X(1)	X(2)
.50	.9052E-01	.7279E 00
1.00	.2848E 00	.1143E 01
1.50	.4952E 00	.1282E 01
2.00	.6644E 00	.1214E 01
2.50	.7664E 00	.1034E 01
3.00	.8015E 00	.8266E 00
3.50	.7862E 00	.6539E 00
4.00	.7431E 00	.5448E 00
4.50	.6932E 00	.5021E 00
5.00	.6512E 00	.5114E 00
5.50	.6245E 00	.5508E 00
6.00	.6138E 00	.5995E 00
6.50	.6158E 00	.6421E 00
7.00	.6251E 00	.6704E 00
7.50	.6369E 00	.6829E 00
8.00	.6472E 00	.6825E 00
8.50	.6541E 00	.6740E 00
9.00	.6572E 00	.6627E 00
9.50	.6572E 00	.6523E 00
10.00	.6552E 00	.6450E 00
10.50	.6525E 00	.6414E 00
11.00	.6499E 00	.6411E 00
11.50	.6482E 00	.6429E 00
12.00	.6473E 00	.6455E 00
12.50	.6472E 00	.6480E 00
13.00	.6476E 00	.6499E 00
13.50	.6482E 00	.6508E 00
14.00	.6488E 00	.6510E 00
14.50	.6493E 00	.6507E 00
15.00	.6495E 00	.6501E 00

$$\dot{\underline{X}}(t) = A\,\underline{X}(t) + B\,\underline{X}(t-.5) + D_1U(t) + D_2U(t-.5)$$

$$A = \begin{bmatrix} -.5 & .5 \\ 0 & -1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ -1.85 & 0 \end{bmatrix}$$

$$D_1 = \begin{bmatrix} 0 \\ 1.85 \end{bmatrix}$$

$$D_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Figure 5.12

Console transaction for example 5.2.4

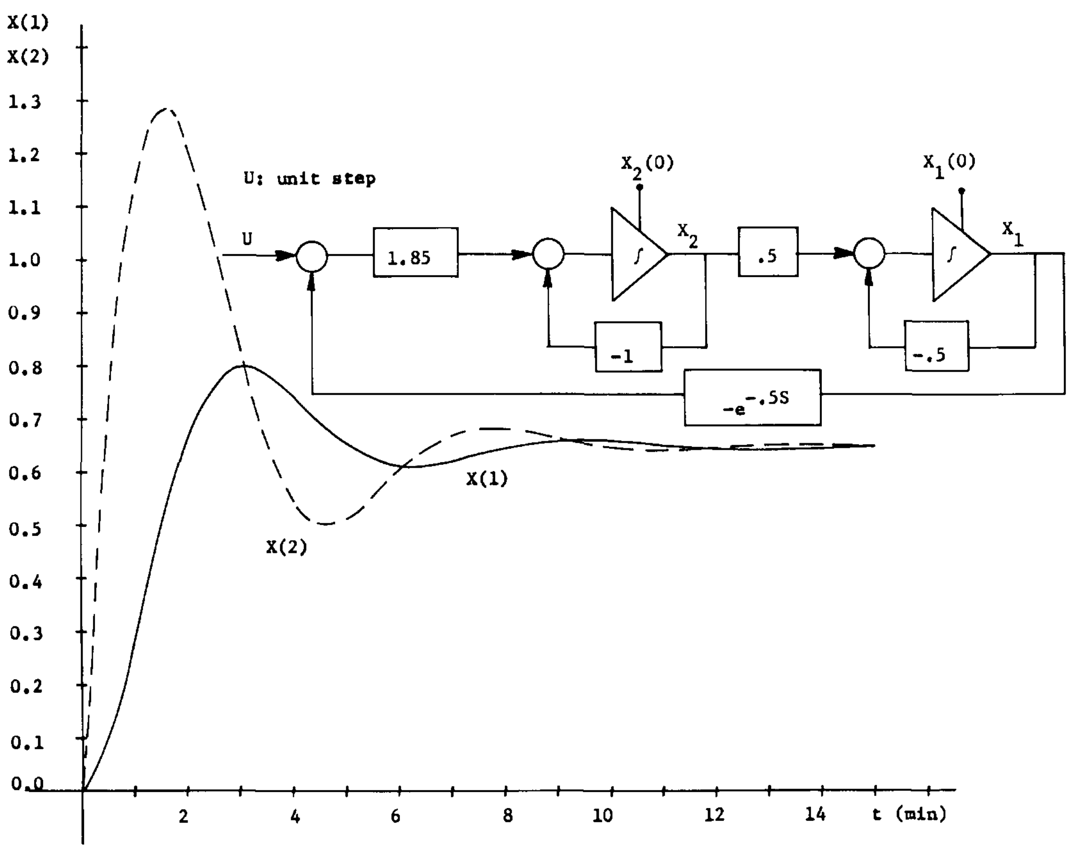


Figure 5.13 Response curves of system for example 5.2.4

```
loadgo timdel delfor pertur
W 1250.1
EXECUTION.
GIVE ORDER OF SYSTEM (M = )
DESIRED SAMPLING TIME (T = )
TIME DELAY (TD = ), FINAL TIME (TF = )
m=2,t=.5,td=1.,tf=15.*

IS THERE ANY DISTURBING SIGNAL
yes

NUMBER OF INPUT SIGNALS (R = )
r=1*

GIVE THE A MATRIX (A(1,1)=--,A(2,1)=--)
a(1,1)=-.5,.5*
a(2,1)=0.,-1.*

GIVE THE B MATRIX (B(1,1)=--,B(2,1)=--)
b(1,1)=0.,0.*
b(2,1)=-1.85,0.*

GIVE THE D1 MATRIX (D1(1,1)=--,D1(2,1)=--)
d1(1,1)=0.*
d1(2,1)=1.85*

GIVE THE D2 MATRIX (D2(1,1)=--,D2(2,1)=--)
d2(1,1)=0.*
d2(2,1)=0.*
DO YOU WISH TO HAVE THE TRANSITION MATRICES
yes
```

$$\dot{\underline{X}}(t) = A\,\underline{X}(t) + B\,X(t-1) + D_1U(t) + D_2U(t-1)$$

$$A = \begin{bmatrix} -.5 & .5 \\ 0 & -1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ -1.85 & 0 \end{bmatrix}$$

$$D_1 = \begin{bmatrix} 0 \\ 1.85 \end{bmatrix}$$

$$D_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

```
TRANSFER MATRIX  PHI(   0)        TRANSFER MATRIX DELTA(   0)
EM(  1,   1) =   .778801E 00      DEL(   1,   1) =   .905188E-01
EM(  1,   2) =   .172270E 00
EM(  2,   1) =   .000000E 00      DEL(   2,   1) =   .727918E 00
EM(  2,   2) =   .606531E 00
TRANSFER MATRIX  PHI(   1)        TRANSFER MATRIX DELTA(   1)
EM(  1,   1) =  -.829913E-01      DEL(   1,   1) =  -.165553E-02
EM(  1,   2) =  -.663267E-02
EM(  2,   1) =  -.637399E 00      DEL(   2,   1) =  -.261964E-01
EM(  2,   2) =  -.763586E-01
TRANSFER MATRIX  PHI(   2)        TRANSFER MATRIX DELTA(   2)
EM(  1,   1) =   .157213E-02      DEL(   1,   1) =   .124783E-04
EM(  1,   2) =   .766560E-04
EM(  2,   1) =   .245409E-01      DEL(   2,   1) =   .296106E-03
EM(  2,   2) =   .149548E-02
TRANSFER MATRIX  PHI(   3)        TRANSFER MATRIX DELTA(   3)
EM(  1,   1) =  -.120288E-04      DEL(   1,   1) =  -.508918E-07
EM(  1,   2) =  -.421960E-06
EM(  2,   1) =  -.283627E-03      DEL(   2,   1) =  -.161214E-05
EM(  2,   2) =  -.116068E-04
TRANSFER MATRIX  PHI(   4)        TRANSFER MATRIX DELTA(   4)
EM(  1,   1) =   .494666E-07      DEL(   1,   1) =   .129736E-09
EM(  1,   2) =   .135504E-08
EM(  2,   1) =   .156125E-05      DEL(   2,   1) =   .514338E-08
EM(  2,   2) =   .481116E-07
TRANSFER MATRIX  PHI(   5)        TRANSFER MATRIX DELTA(   5)
EM(  1,   1) =  -.126765E-09      DEL(   1,   1) =  -.226048E-12
EM(  1,   2) =  -.284835E-11
EM(  2,   1) =  -.501365E-08      DEL(   2,   1) =  -.107649E-10
EM(  2,   2) =  -.123917E-09
```

```
GIVE THE INITIAL STATE (X(1,1)=---)
X(1,1)=0.,0.*
      TIME       X(1)          X(2)

       .50     .9052E-01     .7279E 00
      1.00     .2864E 00     .1169E 01
      1.50     .5134E 00     .1411E 01
      2.00     .7194E 00     .1444E 01
      2.50     .8664E 00     .1306E 01
      3.00     .9369E 00     .1063E 01
      3.50     .9328E 00     .7864E 00
      4.00     .8712E 00     .5417E 00
      4.50     .7771E 00     .3720E 00
      5.00     .6770E 00     .2960E 00
      5.50     .5928E 00     .3093E 00
      6.00     .5384E 00     .3897E 00
      6.50     .5185E 00     .5062E 00
      7.00     .5299E 00     .6269E 00
      7.50     .5634E 00     .7262E 00
      8.00     .6073E 00     .7881E 00
      8.50     .6503E 00     .8080E 00
      9.00     .6838E 00     .7909E 00
      9.50     .7029E 00     .7482E 00
     10.00     .7068E 00     .6944E 00
     10.50     .6980E 00     .6429E 00
     11.00     .6810E 00     .6038E 00
     11.50     .6611E 00     .5825E 00
     12.00     .6430E 00     .5795E 00
     12.50     .6301E 00     .5915E 00
     13.00     .6239E 00     .6128E 00
     13.50     .6243E 00     .6370E 00
     14.00     .6296E 00     .6584E 00
     14.50     .6379E 00     .6733E 00
     15.00     .6466E 00     .6800E 00
END OF EXECUTION
TO CONTINUE, GO TO THE TOP OF A NEW PAGE
AND PRINT AN ASTERISK
```

Figure 5.14 Console transaction for example 5.2.5

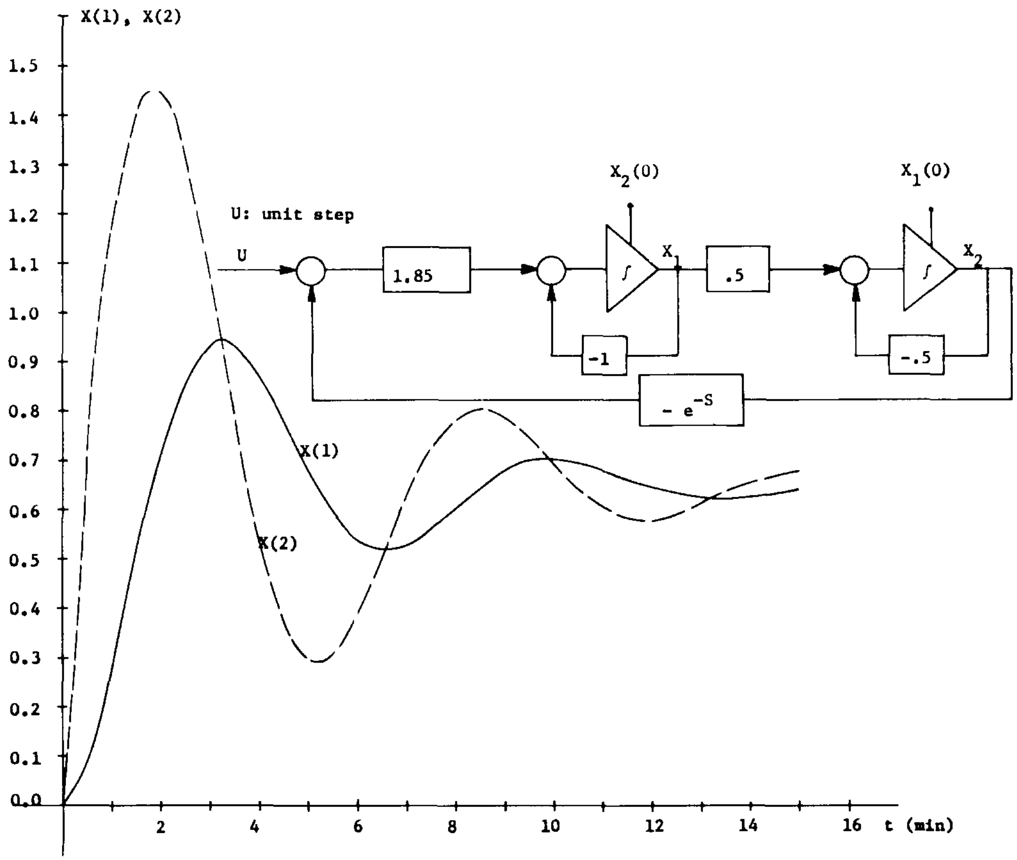


Figure 5.15 Response curves of system for example 5.2.5

$$\dot{\underline{X}}(t) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \underline{X}(t) + \begin{bmatrix} .2 & 0 \\ 0 & -.1 \end{bmatrix} \underline{X}(t - T) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} U(t - T) \quad (5.26)$$

It is assumed the sampling time equal to the time delay. That is,

$$\tau = T = \frac{\pi}{4} \text{ min} \qquad (5.27)$$

Koepcke reported the following results of the plant transition matrices and the control transition matrices:

$$\Phi_0 = \begin{bmatrix} .7071068 & .7071068 \\ -.7071068 & .7071068 \end{bmatrix} \qquad \Delta_0 = \begin{bmatrix} .00000 \\ .00000 \end{bmatrix}$$

$$\Phi_1 = \begin{bmatrix} .1338340 & .0277680 \\ -.0277680 & -.0782980 \end{bmatrix} \qquad \Delta_1 = \begin{bmatrix} .2928932 \\ .7071068 \end{bmatrix}$$

$$\Phi_2 = \begin{bmatrix} .0109582 & .0022524 \\ -.0022524 & .0026278 \end{bmatrix} \qquad \Delta_2 = \begin{bmatrix} .0075873 \\ -.0308106 \end{bmatrix}$$

$$\Phi_3 = \begin{bmatrix} .0005903 & .0000742 \\ -.0000742 & -.0000854 \end{bmatrix} \qquad \Delta_3 = \begin{bmatrix} .0004532 \\ .0007349 \end{bmatrix}$$

$$\Phi_4 = \begin{bmatrix} .0000236 & .0000026 \\ -.0000026 & .0000013 \end{bmatrix} \qquad \Delta_4 = \begin{bmatrix} .0000119 \\ -.0000165 \end{bmatrix}$$

$$\Phi_5 = \begin{bmatrix} .0000008 & .0000001 \\ -.0000001 & -.0000000 \end{bmatrix} \qquad \Delta_5 = \begin{bmatrix} .0000003 \\ .0000002 \end{bmatrix}$$

The time response of the system was obtained assuming a step input and zero initial conditions for the integrators.

The solution is depicted in figures 5.18 and 5.19.

In a similar way, this same example was tested assuming no lags in the system, that is

$$\underline{\dot{V}}(t) = \begin{bmatrix} .2 & 1 & 0 \\ -1 & -.1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \underline{V}(t) \tag{5.28}$$

where

$$\underline{V}(t) = \begin{bmatrix} X_1(t) \\ X_2(t) \\ U(t) \end{bmatrix} \tag{5.29}$$

The evaluation of the state is shown in figures 5.16 and 5.17.

It is interesting to compare the transient response in both cases. As it can be seen in the plots (figures 5.17 and 5.19), the case with delay is something less unstable than the linear one with delay equal to zero.

```
*
GIVE ORDER OF SYSTEM (M = )
SAMPLING TIME (T = ), FINAL TIME (TF = )
m=2,t=.5,tf=15.*

IS THERE ANY DISTURBING SIGNAL
yes

GIVE NUMBER OF INPUT SIGNALS (R = )
r=1*

GIVE THE A MATRIX (A(1,1)=--,A(2,1)=--)
a(1,1)=.2,1.,0.*
a(2,1)=-1.,-.1,1.*
a(3,1)=0.,0.,0.*

GIVE INITIAL STATE (X(1)=--)
x(1)=0.,0.*
```

$$\dot{\underline{V}}(t) = A\,\underline{V}(t)$$

$$\underline{V}(t) = \begin{bmatrix} X(1) \\ X(2) \\ U(1) \end{bmatrix}$$

$$A = \begin{bmatrix} .2 & 1 & 0 \\ -1 & -.1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\underline{X}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

```
 TERMS OF THE MATRIX EXPONENTIAL
         EM(   1,    1) =   .976370E 00
         EM(   1,    2) =   .492031E 00
         EM(   1,    3) =   .124527E 00
         EM(   2,    1) =  -.492031E 00
         EM(   2,    2) =   .828760E 00
         EM(   2,    3) =   .467126E 00
         EM(   3,    1) =   .000000E 00
         EM(   3,    2) =   .000000E 00
         EM(   3,    3) =  1.000000E 00
```

TIME	X(1)	X(2)
.50	.124527E 00	.467126E 00
1.00	.475952E 00	.792990E 00
1.50	.979407E 00	.890141E 00
2.00	.151877E 01	.722940E 00
2.50	.196311E 01	.318989E 00
3.00	.219820E 01	-.234422E 00
3.50	.215544E 01	-.808739E 00
4.00	.183111E 01	-.126367E 01
4.50	.129060E 01	-.148112E 01
5.00	.655877E 00	-.139538E 01
5.50	.783336E-01	-.101203E 01
6.00	-.296938E 00	-.410143E 00
6.50	-.367197E 00	.273318E 00
7.00	-.995124E-01	.874313E 00
7.50	.457555E 00	.124068E 01
8.00	.118173E 01	.127022E 01
8.50	.190332E 01	.938392E 00
9.00	.244459E 01	.308336E 00
9.50	.266306E 01	-.480150E 00
10.00	.248841E 01	-.124111E 01
10.50	.194347E 01	-.178583E 01
11.00	.114338E 01	-.196915E 01
11.50	.272011E 00	-.172740E 01
12.00	-.459826E 00	-.109832E 01

```
     12.50      [illegible]E 00      -.216867E 00
     13.00      [illegible]581E 00    .712322E 00
     13.50      -.[illegible]741E 00   .143467E 01
     14.00       .521283E 00          .184421E 01
     14.50       .[illegible]54091E 01  .173905E 01
     15.00       .[illegible]E 01     .115020E 01
END OF EXECUTION
TO CONTINUE, GO TO THE TOP OF A NEW PAGE
AND PRINT AN ASTERISK
```

Figure 5.16 Console transaction for example 5.3 when time delay = 0

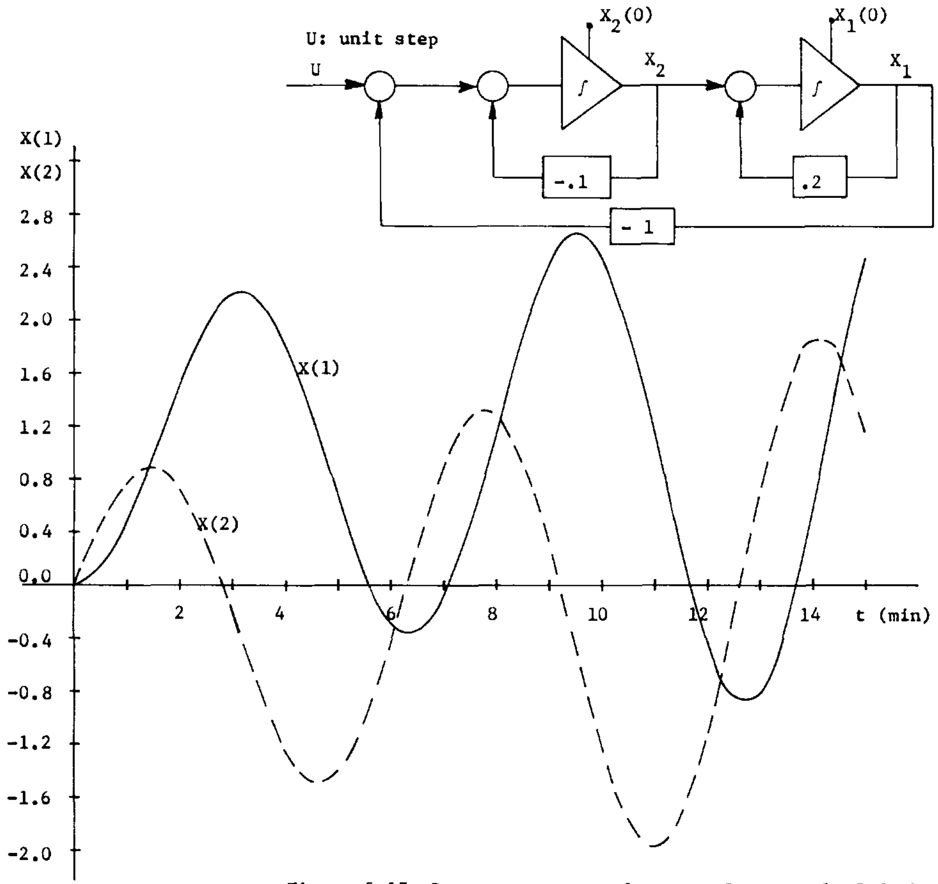


Figure 5.17 Response curves of system for example 5.3 when time delay = 0

```
*
GIVE ORDER OF SYSTEM (M = )
DESIRED SAMPLING TIME (T = )
TIME DELAY (TD = ), FINAL TIME (TF = )
m=2,t=.7853982,td=.7853982,tf=20.*

IS THERE ANY DISTURBING SIGNAL
yes

NUMBER OF INPUT SIGNALS (R = )
r=1*

GIVE THE A MATRIX (A(1,1)=--,A(2,1)=--)
a(1,1)=0.,1.*
a(2,1)=-1.,0.*

GIVE THE B MATRIX (B(1,1)=--,B(2,1)=--)
b(1,1)=.2,0.*
b(2,1)=0.,-.1*

GIVE THE D1 MATRIX (D1(1,1)=--,D1(2,1)=--)
d1(1,1)=0.*
d1(2,1)=0.*

GIVE THE D2 MATRIX (D2(1,1)=--,D2(2,1)=--)
d2(1,1)=0.*
d2(2,1)=1.*
DO YOU WISH TO HAVE THE TRANSITION MATRICES
yes
```

$$\dot{\underline{X}}(t) = A\,\underline{X}(t) + B\,\underline{X}(t - \tfrac{\pi}{4}) + D_1 U(t) + D_2 U(t - \tfrac{\pi}{4})$$

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

$$B = \begin{bmatrix} .2 & 0 \\ 0 & -.1 \end{bmatrix}$$

$$D_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$D_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

```
TRANSFER MATRIX  PHI(   0)
EM(   1,    1) =   .707107E 00
EM(   1,    2) =   .707107E 00
EM(   2,    1) =  -.707107E 00
EM(   2,    2) =   .707107E 00
TRANSFER MATRIX  PHI(   1)
EM(   1,    1) =   .133834E 00
EM(   1,    2) =   .277680E-01
EM(   2,    1) =  -.277680E-01
EM(   2,    2) =  -.782980E-01
TRANSFER MATRIX  PHI(   2)
EM(   1,    1) =   .109582E-01
EM(   1,    2) =   .225237E-02
EM(   2,    1) =  -.225237E-02
EM(   2,    2) =   .262783E-02
TRANSFER MATRIX  PHI(   3)
EM(   1,    1) =   .590343E-03
EM(   1,    2) =   .741765E-04
EM(   2,    1) =  -.741765E-04
EM(   2,    2) =  -.853680E-04
TRANSFER MATRIX  PHI(   4)
EM(   1,    1) =   .235559E-04
EM(   1,    2) =   .259254E-05
EM(   2,    1) =  -.259254E-05
EM(   2,    2) =   .130299E-05
TRANSFER MATRIX  PHI(   5)
EM(   1,    1) =   .748663E-06
EM(   1,    2) =   .651465E-07
EM(   2,    1) =  -.651465E-07
EM(   2,    2) =  -.290989E-07
```

```
TRANSFER MATRIX DELTA(   0)
DEL(   1,    1) =   .000000E 00
DEL(   2,    1) =   .000000E 00
TRANSFER MATRIX DELTA(   1)
DEL(   1,    1) =   .292893E 00
DEL(   2,    1) =   .707107E 00
TRANSFER MATRIX DELTA(   2)
DEL(   1,    1) =   .758732E-02
DEL(   2,    1) =  -.308106E-01
TRANSFER MATRIX DELTA(   3)
DEL(   1,    1) =   .453237E-03
DEL(   2,    1) =   .734907E-03
TRANSFER MATRIX DELTA(   4)
DEL(   1,    1) =   .118773E-04
DEL(   2,    1) =  -.164710E-04
TRANSFER MATRIX DELTA(   5)
DEL(   1,    1) =   .344110E-06
DEL(   2,    1) =   .217079E-06
```

```
        TRANSFER MATRIX  PHI(   6)          TRANSFER MATRIX DELTA(   6)
        EM(   1,    1) =   .197624E-07      DEL(   1,    1) =   .739099E-08
        EM(   1,    2) =   .150533E-08
        EM(   2,    1) =  -.150533E-08      DEL(   2,    1) =  -.367553E-08
        EM(   2,    2) =   .218458E-09

GIVE THE INITIAL STATE (X(1,1)=---)
x(1,1)=0.,0.*
      TIME       X(1)          X(2)

       .79     .0000E 00     .0000E 00
      1.57     .2929E 00     .7071E 00
      2.36     .1008E 01     .9692E 00
      3.14     .1758E 01     .5864E 00
      3.93     .2125E 01    -.2538E 00
      4.71     .1889E 01    -.1100E 01
      5.50     .1158E 01    -.1478E 01
      6.28     .3207E 00    -.1159E 01
      7.07    -.1582E 00    -.2928E 00
      7.85     .3256E-02     .6571E 00
      8.64     .7401E 00     .1163E 01
      9.42     .1663E 01     .9241E 00
     10.21     .2263E 01     .4467E-01
     11.00     .2192E 01    -.1009E 01
     11.78     .1462E 01    -.1654E 01
     12.57     .4567E 00    -.1514E 01
     13.35    -.2735E 00    -.6351E 00
     14.14    -.3088E 00     .5194E 00
     14.92     .3980E 00     .1315E 01
     15.71     .1481E 01     .1292E 01
     16.49     .2349E 01     .4317E 00
     17.28     .2509E 01    -.8188E 00
     18.06     .1842E 01    -.1775E 01
     18.85     .6889E 00    -.1890E 01
     19.63    -.3244E 00    -.1067E 01
     20.42    -.6256E 00     .2718E 00
END OF EXECUTION
TO CONTINUE, GO TO THE TOP OF A NEW PAGE
AND PRINT AN ASTERISK
```

Figure 5.18 Console transaction for example 5.3

when time delay = $\frac{\pi}{4}$ min

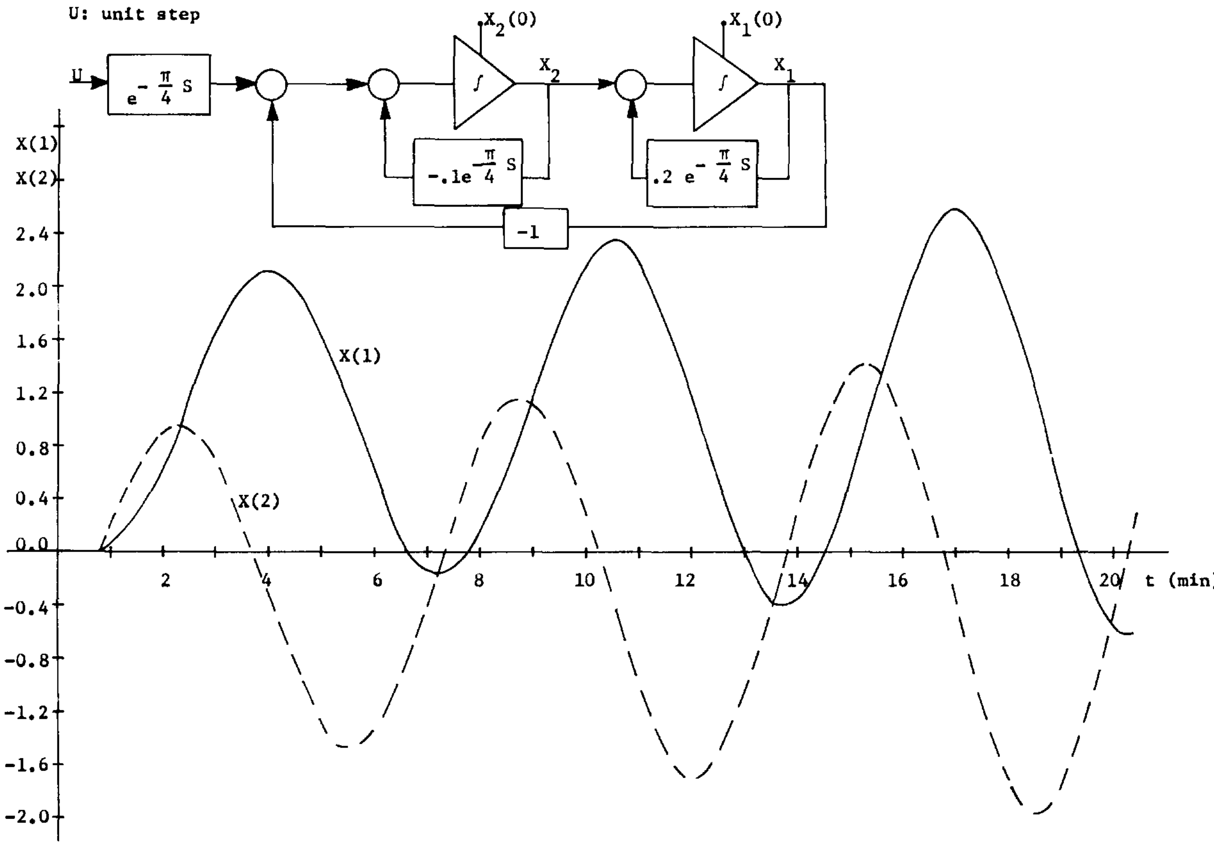


Figure 5.19 Response curves of system for example 5.3 when time delay = $\frac{\pi}{4}$

CHAPTER 6

COMMENTS AND SUGGESTIONS FOR FUTURE RESEARCH

In obtaining e^{AT} by the use of a digital computer the virtues of the series expansion technique are its simplicity and ease in programming. It is not necessary to find the eigenvalues of A. There is, however, some computational disadvantage to the series expansion method. This comes from the convergence requirements for the series. In general, it is reasonable to compute e^{AT} by the power series when T is small. The running time for the matrix exponential simulation will be among the longest of various schemes. Use of the Jordan Canonical form, for example, requires considerably more programming, but will run in a fraction of time needed for the series solution.

Some suggestions concerning the bound on the error in the evaluation of the matrix exponential when the matrix A is known with some error are given by Levis (10).

The simulation technique for linear time-invariant dynamic systems has been tested, and it was found that the use of the augmented A matrix ($\dot{\underline{X}}(t) = A\,\underline{X}(t) + D\,\underline{U}(t)$ can be expressed as $\dot{\underline{V}}(t) = \begin{pmatrix} A & D \\ 0 & 0 \end{pmatrix} \underline{V}(t)$, where $\underline{V}(t) = \begin{pmatrix} X \\ U \end{pmatrix}$) greatly improved the procedure. The reason is that the actual reduction of the elements of the augmented matrix times T to values less than one can be performed successfully. However, this method cannot be used for calculating the digital version of the control transition matrix.

Another scheme that can be used to check the error bound in the state is to divide the time region of interest in two or three parts. Preferably these times should be powers of two times the sampling time. Next, compute the matrix exponential at the desired sampling time.

Recursively multiply it until the matrix exponential is found for the other selected times. The state at those times can be found and saved. Now, using the recursive process of state evaluation at the sampling time, compare the state with the selected ones. If the error is unacceptable, the state with less error can be used as a new initial condition, and the procedure may be continued.

It was found in chapter 3 that the elements $C_{i,j}$ form an array of infinite order. The first row is of main importance because its elements are the terms of $e^{A\tau}$. Therefore, the truncation technique already discussed can be used.

In a similar fashion, the elements $C_{i,i}$ are actually the terms of the infinite series $e^{B\tau}$. It is reasonable to expect smaller values of these norms as "i" grows. Therefore, intuitively the number of terms used to truncate the first row can be used to truncate $\Phi_1(\tau)$, $\Phi_2(\tau)$, etc. It would be interesting to make a study about how the truncation terms should be taken in each row in order to save computation time while maintaining accuracy.

BIBLIOGRAPHY

1. Auslander, D. M., Analysis of Networks of Wavelike Transmission Elements, Sc. D., Thesis. Dept. of Mech. Engr., Mass. Inst. of Tech., August, 1966.

2. Bellman, R., Introduction to Matrix Analysis. Mc Graw-Hill Book Company, Inc. 1960.

3. Buckley, P. S., "Automatic Control of Processes with Dead Time" from "Theory of Continuous Linear Control Systems". Reprint from Automatic and Remote Control. Proceeding of the First Congress of the International Federation of Automatic Control, page 31-37. 1963

4. Coughanowr and Koppel, Process Systems Analysis and Control, Mc Graw-Hill Book Company. 1965.

5. Crisman, P. A., Editor, The Compatible Time-Sharing System; A Programmer's Guide, The M.I.T. Press, Cambridge, Mass., 1965.

6. Everling, W., "On the Evaluation of e^{AT} by Power Series", Proceedings of the I.E.E.E., Vol. 55, No. 3, page 413, March, 1967.

7. Faddeva, V. N., Computational Methods of Linear Algebra. Dover Publications, Inc. New York. 1959.

8. Kalman-Englar, "A User's Manual for the Automatic Synthesis Program". NASA Contractor Report. NASA CR-475. June, 1966.

9. Koepcke, R. W., "On the Control of Linear Systems with Pure Time Delay" Joint Automatic Control Conference. Stanford University. Session XV, paper No. 1, page 397. June 24-25-26, 1964

10. Levis, Alexander H., "Error Bounds in some Matrix Calculations", Control Theory Group. Research Note 1967-2, Electronic Systems Laboratory. Department of Electrical Engineering, Mass. Inst. of Tech.,

May, 1967.

11. Liou, M. L., "A Novel Method of Evaluating Transient Response", Proceedings of the I.E.E.E., Vol 54, No. 1, page 20, January, 1966

12. Liou, M. L., "Evaluation of the Transition Matrix". Proceedings of the I.E.E.E., Vol 55, No. 2, page 228, February 1967.

13. MacMillan-Higgins-Naslin, "Progress in Control Engineering", Vol 1, page 19-33. Academic Press, Inc. Publishers, New York. 1962.

14. Michigan Algorithm Decoder (MAD), University of Michigan, August, 1966

15. Ogata, Katsuhiko, State Space Analysis of Control Systems. Prentice Hall, Inc. 1967.

16. Oguztoreli, M. N., Time Lag Control Systems. Academic Press, New York, London. 1967.

17. Rosenberg, R. C., Computer-Aided Teaching of Dynamic Systems Behavior, Ph. D. Thesis, Dept. of Mech. Engr., Mass. Inst. of Tech., Sept.,1965.

18. Schwarz and Friedland, Linear Systems. Mc Graw-Hill Book Company. 1965

19. Tou, Julius T., Modern Control Theory. Mc Graw-Hill Book Company. 1964

20. Vaughan, D. R., "Application of Distributed Systems Concepts to Dynamic Analyses and Control of Bending Vibrations" Douglas report SM-48759. Prepared under contract No. NAS8-11420 by Douglas Aircraft Company, Inc. Missile and Space Systems Division, Santa Monica, California for NASA, August, 1965.

21. Whitney, D. E., "Propagated Error Bounds for Numerical Solution of Transient Response", Proceedings of the I.E.E.E., (letters), Vol 54, page 1084-1085, August, 1966.

22. Whitney, D. E., "Forced Response Evaluation by Matrix Exponential", Proceedings of the I.E.E.E. (letters), Vol 54, page 1089-1090, August, 1966.

[illegible] ...ear System Theory, ... State Space Approach, [illegible]

[illegible]

TRANS

Purpose: to compute the time response of linear time-invariant systems.

Inputs: order of system (M =); sampling time (T =); final time (TF =); number of input signals (R =); the augmented A matrix and the initial state (X(1) =).

Outputs: the transition matrix; the current time; and the state of the system.

Remarks: main program. Subroutine called by TRANS: EXPMAT, and DISTUR.

```
          PROGRAM COMMON A, EM, M, RIJ, R, X
          DIMENSION X(20),Y(20),E(20),PE(20),XI(20)
          DIMENSION EMP(400,H),A(400,H),EM(400,H)
          INTEGER I,J,M,R,WISH
          FORMAT VARIABLE FM
          VECTOR VALUES H=2,1,0
MAGDA     PRINT COMMENT $GIVE ORDER OF SYSTEM (M = )$
          PRINT COMMENT $SAMPLING TIME (T = ), FINAL TIME (TF = )$
          READ DATA
          PRINT COMMENT $ $
          FM=M
          PRINT COMMENT $IS THERE ANY DISTURBING SIGNAL$
          READ FORMAT S3,WISH
          VECTOR VALUES S3 = $ C3*$
          WHENEVER WISH.E.$YES$
          PRINT COMMENT $ $
          PRINT COMMENT $GIVE NUMBER OF INPUT SIGNALS (R = )$
          READ DATA
          M=M+R
          OTHERWISE
          R=0
          END OF CONDITIONAL
          H(2)=M
          PRINT COMMENT $ $
          PRINT COMMENT $GIVE THE A MATRIX (A(1,1)=--,A(2,1)=--)$
          THROUGH LUPE, FOR I=1,1,I.G.M
LUPE      READ DATA
          PRINT COMMENT $ $
          PRINT COMMENT $GIVE INITIAL STATE (X(1)=--)$
          READ DATA
          THROUGH ALICIA, FOR I=1,1,I.G.(M-R)
ALICIA    XI(I)=X(I)
          TA=T
          WHENEVER R.NE.0
          EXECUTE DISTUR. (TA)
          J=M-R+1
          THROUGH JULIA, FOR I=J,1,I.G.M
          XI(J)=X(J)
JULIA     CONTINUE
          END OF CONDITIONAL
          THROUGH ALMA, FOR I=1,1,I.G.(M)
ALMA      E(I)=0.
          TZ=T
          EXECUTE EXPMAT.(T)
          THROUGH FANNY, FOR I=1,1,I.G.M
          THROUGH FANNY, FOR J=1,1,J.G.M
          PRINT FORMAT CUATRO,I,J,EM(I,J)
          VECTOR VALUES CUATRO = $1H ,S8,3HEM(,I4,1H,,I4,3H) =,E14.6*$
FANNY     CONTINUE
          THROUGH MARTA, FOR I=1,1,I.G.(M-R)
          THROUGH MARTA, FOR J=1,1,J.G.M
MARTA     EMP(I,J)=EM(I,J)
          WHENEVER (M-R).L.6
          PRINT COMMENT $ $
          PRINT FORMAT S1, (I=1,1,I.G.(M-R),I)
          VECTOR VALUES S1 = $ ,S6,4HTIME,S8,'FM'(2HX(,I1,1H),S12)/*$
          END OF CONDITIONAL
          TRANSFER TO TERESA
OLGA      TA=TA+TZ
```

```
            WHENEVER R.NE.0
            EXECUTE DISTUR. (TA)
            END OF CONDITIONAL
TERESA      THROUGH ELENA, FOR I=1,1,I.G.(M-R)
            PE(I)=0.
            Y(I)=0.
            THROUGH MARIA, FOR J=1,1,J.G.M
            Y(I)=Y(I)+EMP(I,J)*X(J)
            PE(I)=(EMP(I,J)+RIJ)*E(J)+RIJ*X(J)+PE(I)
MARIA       CONTINUE
ELENA       CONTINUE
            ENORM=0.
            THROUGH ROSA, FOR I=1,1,I.G.(M-R)
ROSA        ENORM=ENORM+.ABS.(PE(I))
            WHENEVER ENORM.GE.(10..P.-07)
            T=TA
            EXECUTE EXPMAT.(T)
            THROUGH ROSANA, FOR I=1,1,I.G.(M-R)
            PE(I)=0.
            Y(I)=0.
            THROUGH ESTHER, FOR J=1,1,J.G.M
            Y(I)=Y(I)+EM(I,J)*XI(J)
            PE(I)=PE(I)+RIJ*XI(J)
ESTHER      CONTINUE
ROSANA      CONTINUE
            OTHERWISE
            TRANSFER TO SARA
            END OF CONDITIONAL
SARA        THROUGH CARMEN, FOR I=1,1,I.G.(M-R)
            X(I)=Y(I)
CARMEN      E(I)=PE(I)
            J=M-R+1
            THROUGH LILIA, FOR I=J,1,I.G.M
LILIA       E(J)=0.
            WHENEVER (M-R).L.6
            PRINT FORMAT S2,TA,X(1)...X(M-R)
            VECTOR VALUES S2 = $ ,S4,F6.2,'FM'(S3,E14.6)*$
            OTHERWISE
            PRINT RESULTS TA
            PRINT RESULTS X(1)...X(M-R)
            END OF CONDITIONAL
            WHENEVER TA.L.TF, TRANSFER TO OLGA
            PRINT COMMENT $END OF EXECUTION$
            PRINT COMMENT $TO CONTINUE, GO TO THE TOP OF A NEW PAGE$
            PRINT COMMENT $AND PRINT AN ASTERISK$
            READ DATA
            TRANSFER TO MAGDA
            END OF PROGRAM
```

EXPMAT

Purpose: to compute the matrix exponential.

Remarks: subroutine called by TRANS.

```
          EXTERNAL FUNCTION (T)
          PROGRAM COMMON A, EM, M, RIJ, R, X
          DIMENSION A(400,H),EM(400,H),TERM(400,H),NTERM(400,H)
          DIMENSION X(20),B(400,H)
          VECTOR VALUES H=2,1,0
          INTEGER K,I,J,L,M,LL,Y,Q
          ENTRY TO EXPMAT.
          H(2)=M
          THROUGH ELENA, FOR I=1,1,I.G.M
          THROUGH ELENA, FOR J=1,1,J.G.M
          B(I,J)=A(I,J)
ELENA     B(I,J)=B(I,J)*T
          AMIN=B(1,1)
          THROUGH DIANA, FOR I=2,1,I.G.M
          WHENEVER B(I,I).L.AMIN, AMIN=B(I,I)
DIANA     CONTINUE
          FAC=EXP.(AMIN)
          THROUGH OLGA, FOR I=1,1,I.G.M
OLGA      B(I,I)=B(I,I)-AMIN
          Y=.ABS.B(1,1)
          THROUGH SARA, FOR I=1,1,I.G.M
          THROUGH SARA, FOR J=1,1,J.G.M
          WHENEVER .ABS.(B(I,J)).G.(Y+0.), Y=.ABS.(B(I,J))
SARA      CONTINUE
          TAP=1.
          YE=Y+0.
          THROUGH ALMA, FOR Q=1,1,Q.G.10
          TAP=2.*TAP
          WHENEVER TAP.GE.YE, TRANSFER TO ESTHER
ALMA      CONTINUE
ESTHER    Y=TAP
          THROUGH YOLIS, FOR I=1,1,I.G.M
          THROUGH YOLIS, FOR J=1,1,J.G.M
          B(I,J)=B(I,J)/(Y+0.)
YOLIS     TERM(I,J)=B(I,J)
          LL=0
GLORIA    MAXH=0.
          MAXV=0.
          THROUGH MARIA, FOR I=1,1,I.G.M
          SUMH=0.
          SUMV=0.
          THROUGH ROSANA, FOR J=1,1,J.G.M
          SUMH=SUMH+.ABS.TERM(I,J)
          SUMV=SUMV+.ABS.TERM(J,I)
ROSANA    CONTINUE
          WHENEVER SUMH.G.MAXH,MAXH=SUMH
          WHENEVER SUMV.G.MAXV,MAXV=SUMV
MARIA     CONTINUE
          NORM=MAXH
          WHENEVER MAXV.L.NORM,NORM=MAXV
          WHENEVER LL.NE.0, TRANSFER TO DELIA
          SOLO=NORM
          K=2.*NORM
          WHENEVER K.L.2, K=2
          IN=K/2
          VECTOR VALUES CINCO = $1H ,2HK=,I4*$
          THROUGH SUSANA, FOR I=1,1,I.G.M
          THROUGH SUSANA, FOR J=1,1,J.G.M
          UNIT=0.
```

```
             WHENEVER J.E.I, UNIT=1.
             EM(I,J)=UNIT+B(I,J)
SUSANA /     CONTINUE
ISABEL       WHENEVER LL.GE.K, TRANSFER TO GLORIA
             LL=LL+1
             THROUGH LILIA, FOR L=1,1,L.G.M
             THROUGH LILIA, FOR I=1,1,I.G.M
             NTERM(L,I)=0.
             THROUGH EVA, FOR J=1,1,J.G.M
             NTERM(L,I)=NTERM(L,I)+B(L,J)*TERM(J,I)
EVA          CONTINUE
             EM(L,I)=EM(L,I)+NTERM(L,I)/(LL+1.)
LILIA        CONTINUE
             THROUGH AURORA, FOR I=1,1,I.G.M
             THROUGH AURORA, FOR J=1,1,J.G.M
AURORA       TERM(I,J)=NTERM(I,J)/(LL+1.)
             TRANSFER TO ISABEL
DELIA        EPS=SOLO/(K+2.)
             RIJ=NORM*SOLO/((K+1.)*(1.-EPS))
             THROUGH JULIA, FOR I=1,1,I.G.M
             THROUGH JULIA, FOR J=1,1,J.G.M
             WW=.ABS.(EM(I,J)*10..P.-7)
             WHENEVER RIJ.G.WW
             K=K+IN
             TRANSFER TO ISABEL
             OTHERWISE
             TRANSFER TO JULIA
             END OF CONDITIONAL
JULIA        CONTINUE
             THROUGH ALICIA, FOR LL=1,1,LL.G.Q
             THROUGH MARTA, FOR L=1,1,L.G.M
             THROUGH MARTA, FOR I=1,1,I.G.M
             TERM(L,I)=0.
             THROUGH MAGUE, FOR J=1,1,J.G.M
             TERM(L,I)=TERM(L,I)+EM(L,J)*EM(J,I)
MAGUE        CONTINUE
MARTA        CONTINUE
             THROUGH OLIVIA, FOR I=1,1,I.G.M
             THROUGH OLIVIA, FOR J=1,1,J.G.M
OLIVIA       EM(I,J)=TERM(I,J)
ALICIA       CONTINUE
             PRINT COMMENT $ $
             PRINT COMMENT $ TERMS OF THE MATRIX EXPONENTIAL$
             THROUGH CARMEN, FOR I=1,1,I.G.M
             THROUGH CARMEN, FOR J=1,1,J.G.M
CARMEN       EM(I,J)=FAC*EM(I,J)
             FUNCTION RETURN
             END OF FUNCTION
```

[illegible]ISTUR

[illegible] the forcing signal vector at the current

[illegible]

Remarks [illegible] called by [illegible].

TIMDEL

Purpose: to compute the time response of linear systems with lumped parameters and time delays.

Inputs: order of system (M =); sampling time (T =); time delay (TD =); final time (TF =); number of input signals (R =); the A matrix; the B matrix; the D_1 matrix; the D_2 matrix; the initial state (X(1,1) =).

Outputs: the plant transition matrices, the control transition matrices if desired; the current time; and the state of the system.

Remarks: main program. Subroutines called by TIMDEL: DELFOR, and PERTUR.

```
         PROGRAM COMMON EM,DELF,M,R,W,A,B,D1,D2,U
         DIMENSION FM(4000,H),DELF(4000,H),X(400,G),A(400,G)
         DIMENSION B(400,G),D1(400,G),D2(400,G),U(400,E)
         INTEGER I,J,K,L,M,N,LL,Z,W,R,REL,MM,WISH,JJ
         FORMAT VARIABLE FM
         VECTOR VALUES G=2,1,0
         VECTOR VALUES E=2,1,0
         VECTOR VALUES H=3,1,0,0
MAGDA    PRINT COMMENT $GIVE ORDER OF SYSTEM (M = )$
         PRINT COMMENT $DESIRED SAMPLING TIME (T = )$
         PRINT COMMENT $TIME DELAY (TD = ), FINAL TIME (TF = )$
         READ DATA
         FM=M
         PRINT COMMENT $ $
         PRINT COMMENT $IS THERE ANY DISTURBING SIGNAL$
         READ FORMAT S3,WISH
         VECTOR VALUES S3 = $  C3*$
         WHENEVER WISH.E.$YES$
         PRINT COMMENT $ $
         PRINT COMMENT $NUMBER OF INPUT SIGNALS (R = )$
         READ DATA
         OTHERWISE
         R=0
         END OF CONDITIONAL
         REL=TD/T+0.2
         G(2)=M
         H(2)=M
         H(3)=M
         PRINT COMMENT $ $
         PRINT COMMENT $GIVE THE A MATRIX (A(1,1)=--,A(2,1)=--)$
         THROUGH MELA, FOR I=1,1,I.G.M
MELA     READ DATA
         PRINT COMMENT $ $
         PRINT COMMENT $GIVE THE B MATRIX (B(1,1)=--,B(2,1)=--)$
         THROUGH MALENA, FOR I=1,1,I.G.M
MALENA   READ DATA
         WHENEVER R.E.0,TRANSFER TO JULIA
         PRINT COMMENT $ $
         PRINT COMMENT $GIVE THE D1 MATRIX (D1(1,1)=--,D1(2,1)=--)$
         THROUGH ALMA, FOR I=1,1,I.G.M
ALMA     READ DATA
         PRINT COMMENT $ $
         PRINT COMMENT $GIVE THE D2 MATRIX (D2(1,1)=--,D2(2,1)=--)$
         THROUGH BERTA, FOR I=1,1,I.G.M
BERTA    READ DATA
JULIA    EXECUTE DELFOR. (T)
         PRINT COMMENT $DO YOU WISH TO HAVE THE TRANSITION MATRICES$
         READ FORMAT S3,WISH
         WHENEVER WISH.E.$YES$
         THROUGH DULCE, FOR L=1,1,L.G.W
         LL=L-1
         WHENEVER R.E.0
         PRINT FORMAT OCHO,LL
         VECTOR VALUES OCHO = $1H ,S8,15HTRANSFER MATRIX,S2,
        14HPHI(,I4,1H)*$
         OTHERWISE
         PRINT FORMAT SEIS,LL,LL
         VECTOR VALUES SEIS= $1H ,S8,15HTRANSFER MATRIX,S2,
        14HPHI(,I4,1H),S8,22HTRANSFER MATRIX DELTA(,I4,1H)*$
```

[redacted]

```
CFLIA    CONTINUF
         WHENEVFR Z.E.1, L=L-1
GLORIA   THROUGH ALICIA, FOR K=1,1,K.G.W*REL
         THROUGH ALICIA, FOR I=1,1,I.G.M
         B(K,I)=X(K+1,I)
         WHENEVER K.E.W, TRANSFER TO ALICIA
         A(K,I)=U(K+1,I)
ALICIA   CONTINUE
         THROUGH MARTA, FOR K=1,1,K.G.W*REL
         THROUGH MARTA, FOR I=1,1,I.G.M
         X(K,I)=B(K,I)
         WHENEVER K.E.W, TRANSFER TO MAR
         U(K,I)=A(K,I)
MARTA    CONTINUE
         TRANSFER TO SONIA
SILVIA   PRINT COMMENT $END OF EXECUTION$
         PRINT COMMENT $TO CONTINUE, GO TO THE TOP OF A NEW PAGE$
         PRINT COMMENT $AND PRINT AN ASTERISK$
         READ DATA
         TRANSFER TO MAGDA
         END OF PROGRAM
```

DELFOR

Purpose: to compute the plant transition matrices and the control transition matrices.

Remarks: subroutine called by TIMDEL.

```
          EXTERNAL FUNCTION (T)
          PROGRAM COMMON EM,DELF,M,R,W,A,B,D1,D2,U
          DIMENSION C(11000,H),A(400,G),B(400,G),EM(4000,H),XX(400,G)
          DIMENSION TERM(400,G),NTERM(400,G),UU(400,G),D1(400,G)
          DIMENSION DELF(4000,H),D2(400,G),U(400,E)
          INTEGER I,J,K,L,M,N,Y,Q,R,W
          VECTOR VALUES H=3,1,0,0
          VECTOR VALUES G=2,1,0
          VECTOR VALUES E=2,1,0
          ENTRY TO DELFOR.
          G(2)=M
          H(2)=M
          H(3)=M
          LINDA=0.
          ROSA=-1.
          THROUGH YOLIS, FOR I=1,1,I.G.M
          THROUGH YOLIS, FOR J=1,1,J.G.M
          WHENEVER J.G.R, TRANSFER TO DELIA
          D1(I,J)=D1(I,J)*T
          D2(I,J)=D2(I,J)*T
DELIA     A(I,J)=A(I,J)*T
          TERM(I,J)=A(I,J)
YOLIS     B(I,J)=B(I,J)*T
          N=0
MAGUE     MAXH=0.
          MAXV=0.
          THROUGH MARIA, FOR I=1,1,I.G.M
          SUMH=0.
          SUMV=0.
          THROUGH ROSANA, FOR J=1,1,J.G.M
          SUMH=SUMH+.ABS.TERM(I,J)
          SUMV=SUMV+.ABS.TERM(J,I)
ROSANA    CONTINUE
          WHENEVER SUMH.G.MAXH,MAXH=SUMH
          WHENEVER SUMV.G.MAXV,MAXV=SUMV
MARIA     CONTINUE
          NORM=MAXH
          WHENEVER MAXV.L.NORM,NORM=MAXV
          WHENEVER LINDA.NE.0., TRANSFER TO CARMEN
          WHENEVER N.NE.0, TRANSFER TO CHELA
          SOLO=NORM
          K=2.*NORM
          WHENEVER K.L.2, K=2
          W=1
          IN=K/2
          THROUGH SALOME, FOR I=1,1,I.G.M
          THROUGH SALOME, FOR J=1,1,J.G.M
          C(1,I,J)=0.
          WHENEVER J.E.I,C(1,I,J)=1.
          EM(W,I,J)=C(1,I,J)
          XX(I,J)=EM(W,I,J)
          UU(I,J)=0.
SALOME    TERM(I,J)=C(1,I,J)
ISABEL    N=0
FANNY     WHENEVER N.GE.K.AND.LINDA.E.0.,TRANSFER TO MAGUE
          WHENEVER N.GE.K.AND.LINDA.NE.0.,TRANSFER TO ELENA
          N=N+1
          THROUGH HILDA, FOR L=1,1,L.G.M
          THROUGH HILDA, FOR I=1,1,I.G.M
```

```
            NTERM(L,I)=0.
            THROUGH LILIA, FOR J=1,1,J.G.M
            WHENEVER LINDA.E.0.,C(N+1,J,I)=0.
            NTERM(L,I)=NTERM(L,I)+A(L,J)*TERM(J,I)+B(L,J)*C(N+1,J,I)
LILIA       CONTINUE
            EM(W,L,I)=EM(W,L,I)+NTERM(L,I)/(N+LINDA)
            WHENEVER R.E.0, TRANSFER TO HILDA
            XX(L,I)=XX(L,I)+NTERM(L,I)/((N+LINDA)*(N+LINDA+1.))
HILDA       CONTINUE
            THROUGH JULIA, FOR I=1,1,I.G.M
            THROUGH JULIA, FOR J=1,1,J.G.M
            C(N+1,I,J)=NTERM(I,J)/(N+LINDA)
JULIA       TERM(I,J)=C(N+1,I,J)
            TRANSFER TO FANNY
CHELA       EPS=SOLO/(K+2.)
            RIJ=NORM*SOLO/((K+1.)*(1.-EPS))
            THROUGH EVA, FOR I=1,1,I.G.M
            THROUGH EVA, FOR J=1,1,J.G.M
            WW=.ABS.(EM(W,I,J)*10..P.-07)
            WHENEVER RIJ.G.WW
            K=K+IN
            TRANSFER TO FANNY
            OTHERWISE
            TRANSFER TO EVA
            END OF CONDITIONAL
EVA         CONTINUE
ELENA       LINDA=LINDA+1.
            ROSA=ROSA+1.
            WHENEVER R.E.0, TRANSFER TO PATY
            THROUGH MARTA, FOR L=1,1,L.G.M
            THROUGH MARTA, FOR I=1,1,I.G.R
            TERM(L,I)=0.
            THROUGH AURORA, FOR J=1,1,J.G.M
            TERM(L,I)=TERM(L,I)+XX(L,J)*D1(J,I)+UU(L,J)*D2(J,I)
AURORA      CONTINUE
MARTA       CONTINUE
            THROUGH IRMA, FOR I=1,1,I.G.M
            THROUGH IRMA, FOR J=1,1,J.G.R
IRMA        DELF(W,I,J)=TERM(I,J)
PATY        THROUGH SONIA, FOR I=1,1,I.G.M
            THROUGH SONIA, FOR J=1,1,J.G.M
            TERM(I,J)=EM(W,I,J)
SONIA       UU(I,J)=XX(I,J)
            TRANSFER TO MAGUE
CARMEN      WHENEVER NORM.LE.10..P.-07,TRANSFER TO DIANA
            THROUGH YOCO, FOR L=1,1,L.G.M
            THROUGH YOCO, FOR I=1,1,I.G.M
            NTERM(L,I)=0.
            THROUGH JOSEFA, FOR J=1,1,J.G.M
JOSEFA      NTERM(L,I)=NTERM(L,I)+B(L,J)*C(1,J,I)
YOCO        CONTINUE
            W=W+1
            THROUGH ELISA, FOR I=1,1,I.G.M
            THROUGH ELISA, FOR J=1,1,J.G.M
            C(1,I,J)=NTERM(I,J)/(ROSA+1.)
            XX(I,J)=C(1,I,J)/(ROSA+2.)
            EM(W,I,J)=C(1,I,J)
ELISA       TERM(I,J)=C(1,I,J)
            TRANSFER TO ISABEL
```

```
DIANA     CONTINUE
          FUNCTION RETURN
          END OF FUNCTION
```

PERTUR

Purpose: to compute the forcing signal vector at the current time. The program keeps track of the past.

Remarks: subroutine called by TIMDEL.

```
EXTERNAL FUNCTION (TA,LL)
PROGRAM COMMON EM,DELF,M,R,W,A,B,D1,D2,U
DIMENSION EM(4000,H),DELF(4000,H),A(400,G),B(400,G)
DIMENSION D1(400,G),D2(400,G),U(400,E)
INTEGER I,LL,R,W,M
VECTOR VALUES G=2,1,0
VECTOR VALUES E=2,1,0
VECTOR VALUES H=3,1,0,0
ENTRY TO PERTUR.
G(2)=M
H(2)=M
E(2)=W
H(3)=M
U(LL,1)=----
U(LL,2)=----
---
U(LL,R)=----
FUNCTION RETURN
END OF FUNCTION
```

CS-TR Scanning Project
Document Control Form

Date : 12/ 11 / 95

Report # LCS-TR-45

Each of the following should be identified by a checkmark:
Originating Department:

- ☐ Artificial Intellegence Laboratory (AI)
- ☒ Laboratory for Computer Science (LCS)

Document Type:

- ☒ Technical Report (TR)
- ☐ Technical Memo (TM)
- ☐ Other: ____________

Document Information

Number of pages: 104 (111-images)
Not to include DOD forms, printer intstructions, etc... original pages only.

Originals are:

- ☐ Single-sided or
- ☒ Double-sided

Intended to be printed as :

- ☐ Single-sided or
- ☒ Double-sided

Print type:

- ☒ Typewriter
- ☐ Offset Press
- ☐ Laser Print
- ☐ InkJet Printer
- ☐ Unknown
- ☐ Other: ____________

Check each if included with document:

- ☒ DOD Form
- ☒ Funding Agent Form
- ☒ Cover Page
- ☐ Spine
- ☐ Printers Notes
- ☐ Photo negatives
- ☐ Other: ____________

Page Data:

Blank Pages(by page number): ____________

Photographs/Tonal Material (by page number): ____________

Other (note description/page number):

Description : Page Number:

IMAGE MAP! (1-104) UN#'ED TITLE PAGE, ii-V, UN# BLANK,
1-97, UN#'ED BLANK
(105-111) SCANCONTROL, COVER, FUNDING AGENT, DOD, TRGT'S (3)

Scanning Agent Signoff:

Date Received: 12/ 11 / 95 Date Scanned: 12/ 22/ 95 Date Returned: 12/ 28/ 95

Scanning Agent Signature: Michael W. Cook

Rev 9/94 DS/LCS Document Control Form cstrform.vsd

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Massachusetts Institute of Technology Project MAC	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP None

3. REPORT TITLE

On the Simulation of Dynamic Systems with Lumped Parameters and Time Delays

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Master's Thesis, Mechanical Engineering, August 1967

5. AUTHOR(S) *(Last name, first name, initial)*

Leal-Cantu, Nestor

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
October 1967	102	23

8a. CONTRACT OR GRANT NO. Office of Naval Research, Nonr-4102(01) b. PROJECT NO. NR 048-189 c. RR 003-09-01 d.	9a. ORIGINATOR'S REPORT NUMBER(S) MAC-TR-45 (THESIS) 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency 3D-200 Pentagon Washington, D. C. 20301

13. ABSTRACT

This thesis developes a method for digital simulation of linear time-invariant dynamic systems with lumped parameters and time delays. Ordinarily, such systems can be described by a linear matrix differential-difference equation, which can be transformed to an infinite-dimensional difference equation whose solution is obtained in a recursive way.

As the present method depends on the accuracy of evaluation of the matrix exponential, a simple computational procedure based on the truncation of the infinite series for e^{AT} is described.

In addition, an algorithm is given which ensures that the transient state of an unforced linear time-invariant dynamic system with zero time delay is calculated to a specified accuracy.

Several sample problems are included.

14. KEY WORDS

Computers	Machine-aided cognition	Simulation
Digital simulation	Multiple-access computers	Time-sharing
Linear dynamic systems	On-line computers	Time-shared computers

DD FORM 1 NOV 65 1473 (M.I.T.)

UNCLASSIFIED

Security Classification

Scanning Agent Identification Target

Scanning of this document was supported in part by the **Corporation for National Research Initiatives,** using funds from the **Advanced Research Projects Agency** of the **United states Government** under Grant: **MDA972-92-J1029.**

The scanning agent for this project was the **Document Services** department of the **M.I.T Libraries.** Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences.**