USE OF LOW ORDER MODELS

FOR NEAR-OPTIMAL CONTROL OF HIGH-ORDER SYSTEMS

USE OF LOW ORDER MODELS
FOR NEAR-OPTIMAL CONTROL OF HIGH-ORDER SYSTEMS

by

HUIBREGT de BRUIN, B. Eng.

A Thesis

Submitted to the Faculty of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree

Master of Engineering

McMaster University

ACKNOWLEDGEMENTS

I wish to express my gratitude to Dr. N. K. Sinha for his encouragement, guidance and many valuable contributions throughout this project.

I would also like to thank my colleagues for their helpful suggestions in several discussions.

Finally, special thanks to my wife Joanne, who helped prepare this thesis in its final written form.

MASTER OF ENGINEERING
(Electrical Engineering)

McMASTER UNIVERSITY
Hamilton, Ontario

TITLE: Use of Low Order Models for Near-Optimal Control of High Order Systems.

AUTHOR: Huibregt de Bruin, B. Eng. (McMaster University)

SUPERVISOR: N. K. Sinha, B.Sc. Eng. (Banaras),
Ph. D. (University of Manchester).

NUMBER OF PAGES: vi; 106.

SCOPE AND CONTENTS: Ten different reduced models, of a particular test system are selected. Two cost functions are selected and the test system minimum cost is found for each. The model optimal controls are found for each cost function and are used to provide sub-optimal control of the system using two different methods. The system cost is calculated for each case and compared to the minimum attainable. The reduction methods are compared with a view to application for the near-optimal control of a linear system.

TABLE OF CONTENTS

TABLE OF CONTENTS Continued. Page

LIST OF ILLUSTRATIONS

CHAPTER I

INTRODUCTION

One of the goals of modern control engineering, is to control a particular system in some optimal manner. Modern processes such as nuclear reactors, high-speed rolling mills, jet aircraft controllers, and spacecraft systems usually specify fine tolerances of operational limits. In controlling these processes, the control problem may involve minimizing various parameters such as control energy and the time required to go from one state to another. Although a number of computational methods have been developed for the solution of the optimal control problem, these are not suitable for the control of many systems such as those mentioned. Because of the increasingly comprehensive nature and complexity of these systems, these methods are not suitable due to the large amount of computation required. This is especially evident for the case of on-line optimal control. In addition, some of these methods require a complete and precise knowledge of the system parameters, which are not often known.

One way of overcoming these difficulties, is to obtain a reduced linear model of the high-order system, which is compu-

tationally or analytically more tractable than the complex system. This low-order model can then be used for an approximate computation of the optimal control. The resulting sub-optimal control may often be sufficiently close to the actual optimum, but in any case, it may serve as the first approximation to the optimum. The reduced model may also be particularly useful for the adaptive control of a system with slowly varying parameters.

In the last five years a number of new methods have been proposed for determining a low-order model for a high-order system. Since the reduction techniques are basically different, a large number of different low-order models can be obtained for a given system. It would be desirable therefore, to determine which of these models would be most suitable for determining the sub-optimal control for the system.

A number of different reduction methods have been proposed by various authors [1,2,3,4,5,6,7,8,9] and applied to the same test system [5,6,7]. This test system, selected so that it would not be particularly suited for reduction by any one method, is a realistic aircraft control system. Of the various models derived, ten models representing as many different reduction methods

as possible, were selected. In order to make a reasonable comparison of the model suitability, all models selected were of second order.

To investigate the suitability of each model for providing a sub-optimal control of the system, two quadratic cost functions to be minimized for the control interval were selected. For the first, the closed-loop optimal control of each reduced model was calculated and used to control the system sub-optimally in an open-loop sense. The reduced model optimal feedback coefficients were then used to provide a closed-loop sub-optimal control for the system. For the second cost function, the optimal feedback coefficients were again calculated for each model and used to provide sub-optimal feedback control for the system. For each model, the resultant cost function was computed and compared with the minimum attainable for the system. A comparison of the models was made with a view to the relative difficulty of finding the model optimal feedback control and the suitability of the resulting feedback coefficients in providing a sub-optimal control for the system.

The material contained in the following chapters follows the order of the preceding discussion. The principle and appli-

cation of each method of reduction considered is discussed in Chapter 2. Chapter 3 gives a brief summary of optimal control theory, stressing closed-loop or the feedback implementation of optimal control. The reduced models considered are mentioned in Chapter 4, along with the procedures followed in using each model to provide sub-optimal control for the system. The results of the sub-optimal control of the system, and a comparison of model features and suitability are included in Chapter 5. The conclusions of this work are then drawn in Chapter 7.

CHAPTER 2

METHODS OF REDUCTION

2.0 Introduction

In the past five years a number of different methods have been proposed for determining a low-order model for a high-order system. They can be divided into two groups; (i) those which neglect the modes of the original system that contribute little to the overall response and (ii) those which determine an optimum model of a given order so that the error between the response of the model and that of the system for the same input is minimized with respect to a specified criterion. A number of methods, each using a different technique, have been selected and are described briefly in this chapter. The modal methods are discussed first and then the trajectory fitting methods. Lastly, a transfer function reduction method is described which does not fall into either of the two groups described above.

2.1 Problem Formulation

The general problem in system model reduction may be

stated as follows : given the n^{th} order linear, time invariant, controllable system described by

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

$$\underline{y}(t) = D\underline{x}(t) \qquad \ldots\ldots \text{(2.1-1)}$$

where $\underline{x}(t)$ is an n-dimensional state vector, $\underline{u}$ is a p-dimensional vector of forcing functions and $\underline{y}$ is the k-dimensional output vector, find an m^{th} order system ($k \leq m < n$) described by

$$\dot{\underline{x}}_r(t) = A_r\,\underline{x}_r(t) + B_r\underline{u}(t)$$

$$\underline{y}_r(t) = D_r\underline{x}_r(t) \qquad \ldots\ldots \text{(2.1-2)}$$

such that for a specified set of inputs the reduced system response is a satisfactory approximation to the original system response. The problem may also be equivalently stated in terms of the discrete-time analog of equations 2.1-1 and 2.1-2. For single input-output systems the problem may be stated in terms of the transfer functions of the system and reduced model.

2.2 Modal Reduction Methods

The reduction methods proposed by Davison [1,2] and Mitra[4] both result in reduced systems which retain specified eigenvalues of the original system. They are also both projection methods although only Mitra has explicitly discussed this aspect. Their common characteristics will be considered first therefore. Mitra[4] has shown that these projection methods can usefully be broken down into two steps. The first step consists in choosing the modes to be discarded and the construction of a dynamic system with the same order as the original in which these modes are uncontrollable ("decontrolling" step). The second, "contraction" step, is the synthesis of an m^{th} order controllable system from the decontrolled system by retaining the controllable modes.

The first problem is then to find a projection operator P which operates on the original system to produce a trajectory confined to an m-dimensional subspace, S_1 of the original n-dimensional state space S. That is, the decontrolled trajectory is given by

$$\underline{\hat{x}}(t) = P\ \underline{x}(t)$$

However, for $\underline{\hat{x}}(t)$ to be a basis for a practically useful model, it

should also be the solution of a differential equation. The additional requirement imposed is that $\hat{\underline{x}}(t)$ should also satisfy

$$\dot{\hat{\underline{x}}} = \hat{A}\,\hat{\underline{x}}(t) + \hat{B}\,\underline{u}(t)$$

It has been shown that if $\hat{\underline{x}}(t)$ is to satisfy both equations, it is necessary and sufficient that the projection be along a subspace S_2, which is invariant under the linear transformation A. In this case it is easy to show that

$$\hat{A} = PAP^{-1}$$

$$\hat{B} = PB$$

There is however, no a priori restriction on the choice of the subspace S_1, on which to project, except that it must be disjoint from S_2. The choice of subspace on which to project is the fundamental difference between the method due to Davison and that proposed by Mitra.

In the method due to Davison, the subspace S_1 projected on, is also invariant under the transformation A. The projector which projects both on and along subspaces invariant under the linear transformation A, may be found by partitioning the matrix, the columns of which are the generalized eigenvectors of A.

$$T = \left[\begin{array}{c|c} T_1 & T_2 \end{array}\right] \qquad \ldots\ldots\ (2.2\text{-}1)$$

where the eigenvectors comprising T_1 are those correspondending

to the eigenvalues to be retained. By defining

$$S = T^{-1} = \begin{bmatrix} S_1 \\ \cdots\cdots \\ S_2 \end{bmatrix}$$

it has been shown [4] that the required projector is given by

$$P_{oa} = T_1 S_1$$

This equation is valid for complex eigenvectors provided that both parts of the complex conjugate pair are included in either T_1 or T_2.

In the resulting decontrolled system, since the (n-m) uncontrollable modes of $\hat{A}$ are not excited, only m of the n components of $\underline{\hat{x}}$ are linearly independent. Contraction consists of choosing a suitable basis of dimension m, and a suitable set of m variables. A particular contraction may be specified by defining an n x m matrix

$$C = \begin{bmatrix} C_1 \\ \cdots \\ C_2 \end{bmatrix} \begin{matrix} m \\ \\ n-m \end{matrix} \qquad \ldots\ldots \text{(2.2-2)}$$

which spans the controllable subspace of $\hat{A}$, $\hat{B}$, that is, S_1.

The coefficient matrices of the reduced system are then given by [4].

$$A_r = C_1^{-1} \hat{A}_{11} C_1 + C_1^{-1} \hat{A}_{12} C_2$$

$$B_r = C_1^{-1} \hat{B}_1$$

$$D_r = D_1 C_1 + D_2 C_2 \qquad \ldots\ldots \text{(2.2-3)}$$

where $\hat{A}$, $\hat{B}$, and D, have been partioned to conform to the partioning of C. This representation of the contraction step is valid for both Mitra's and Davison's methods.

For the method proposed by Davison, projection is on and along subspaces invariant under A, and hence

$$C = T_1$$

where T_1 is given by equation 2.2-1.

If it is assumed that

$$D = [I_m 0]$$

an additional transformation is performed on the reduced system to force

$$D_r' = [I_m]$$

The reduced system equations are then given by

$$A_r' = A_{11} + A_{12}\, T_{21}\, T_{11}^{-1}$$

$$B_r' = T_{11}\, S_1\, B \qquad \ldots\ldots (2.2\text{-}4)$$

$$D_r' = I_m$$

The modification suggested by Davison (which provides the correct steady-state step response) is obtained by modifying the measurement equation from equation 2.1-2 to

$$\underline{y}_r(t) = D_r\, \underline{x}_r(t) + Z\underline{u}(t) \qquad \ldots\ldots (2.2\text{-}5)$$

where Z is a correction term given by

$$Z = D_r A_r^{-1} B_r - DA^{-1} B$$

Equations 2.2-4 and 2.2-5 give the reduced model proposed by Davison. The method proposed by Mitra involves choosing a subspace S_1 so as to minimize the projection error. This projection error is defined as

$$E(T) = \sum_{i=1}^{p} \int_0^T (\underline{x}^i(t) - \hat{\underline{x}}^i(t)^T Q(\underline{x}^i(t) - \hat{\underline{x}}^i(t))\, dt.$$

where $\underline{x}^i(t)$ and $\hat{\underline{x}}^i(t)$ are the responses of the original and decontrolled systems to the i^{th} input, p being the forcing function dimension. The weighting matrix Q is at least positive semi-definite. The system response matrix is defined as

$$W(T) = \sum_{i=1}^{p} \int_0^T \underline{x}^i(t)\, \underline{x}^i(t)^T dt.$$

For a specified class of inputs and a specified order of reduction the projection error is a function only of the choice of subspace on which to project, that is, the choice of modes to be retained. Mitra [4] has derived an algorithm to solve the problem of choosing this subspace. The following matrices R, R_1, and R_2 are defined as

$$R = [R_1\ R_2] = [r_1, r_2 \cdots r_m\ \ r_{m+1} \cdots r_n]$$

and S, S_1, and S_2 as

$$S = [S_1\ S_2] = [s_1, s_2, \cdots s_m\ s_{m+1}\ \cdots s_n]$$

such that

$$R^*S = I$$

where * denotes complex conjugate transpose.

The columns of R_2 span the invariant subspace S_2, along which to project. The required projector is then given by

$$P_a = R_1 S_1^*$$

or equivalently by

$$P_a = I - R_2 S_2^*$$

As the columns of R_2 span a subspace invariant under A they must be a linear combination of the eigenvectors which span that subspace. If these eigenvectors which correspond to the modes to be neglected, are denoted by

$$T_2 = [\, t_{m+1},\ t_{m+2} \ \cdots\ t_n]$$

then

$$R_2 = T_2\, G$$

where G is a $(n-m) \times (n-m)$ non-singular matrix. It has been shown that G can be defined as

$$G = [\Lambda_Q^{1/2} H^*]^{-1}$$

where Λ_Q and H form the eigen-system of

$$T_2^* \, QT_2$$

S_2 is given by

$$S_2 = W(T)^{-1} R_2 [R_2^* W(T)^{-1} R_2]^{-1}$$

R_1 may be found as a (non-unique) solution to the overdetermined set of equations

$$S_2^* R_1 = 0 \qquad \ldots\ldots (2.2\text{-}6)$$

The projector defined by the above procedure, minimizes the projection error for projection along a specified subspace. A search must then be made to determine which invariant subspace to project along.

The contraction process for the optimal projection method is also defined by equations 2.2-2 and 2.2-3. However, for this method C is defined by

$$C = R_1$$

where R_1 is defined by equation 2.2-6. The optimal projection method requires considerably more computation than that proposed by Davison.

In summary, both methods of reduction require knowledge of all parameters of the original system. In addition, both require finding the eigenvalues and eigenvectors of this system. They are

directly applicable to multiple input-output systems and can easily be extended to discrete time systems. Since both methods require the retention of certain modes and neglect of the rest, these methods are most applicable to those systems which have dominant eigenvalues.

2.3 Anderson's Method

Anderson [3] presents a method of system reduction based on a geometrical consideration of the reduction problem as developed from the theory of linear vector spaces. The state space equations are solved at regular time intervals up to some limit at which only insignificant response changes occur. These solutions are then substituted in the state space equations which represent the unknown low order system and the parameters evaluated which will give these solutions. The method is developed for discrete time systems as represented by the following equations

$$\underline{x}(i) = F\,\underline{x}\,(i-1) + E\,\underline{u}(i-1) \qquad \ldots\ldots\ (2.3\text{-}1)$$

$$\underline{y}(i) = D\underline{x}\,(i)$$

which are of order n.

The reduced model of order m is given by the equations

$$\underline{x}_r(i) = F_r\underline{x}_r(i-1) + E_r\underline{u}(i-1)$$

$$\underline{y}_r(i) = D_r \, \underline{x}_r(i)$$

For simplicity, it is assumed that the order m of the reduced system, is the same as the order of the original system output vector and hence, D_r can be assumed to be I_m. Also, it is assumed that a linear transformation has been applied, if necessary, so that

$$D = [I_m \; 0]$$

If the sequences which make up the trajectory of the original system are written as

$$\{ \underline{x}(0), \underline{x}(1), \ldots\ldots \underline{x}(k-1), \underline{x}(k) \}$$

$$\{ \underline{u}(0), \underline{u}(1), \ldots\ldots \underline{u}(k-1) \}$$

$$\{ \underline{y}(0), \underline{y}(1), \ldots\ldots \underline{y}(k-1), \underline{y}(k) \}$$

and the $\underline{x}$ sequence written as two sequences of length k,

$$\{ \underline{x}(0), \underline{x}(1), \ldots\ldots \underline{x}(k-1) \}$$

$$\{ \underline{x}(1), \underline{x}(2), \ldots\ldots \underline{x}(k) \}$$

these are related by equation 2.3-1. Combining x(k) and u(k) as a single vector, equation 2.3-1 can be written as

$$[\underline{x}(1), \underline{x}(2), \ldots \underline{x}(k)] = [FE] \begin{bmatrix} \underline{x}(0) & \ldots \underline{x}(k-1) \\ \underline{u}(0) & \ldots \underline{u}(k-1) \end{bmatrix}$$

The reduction is achieved by seeking $[F_r \; E_r]$ such that

$$[\underline{y}(1), \underline{y}(2), \ldots \underline{y}(k)] = [F_r \; E_r] \begin{bmatrix} \underline{y}(0) \ldots \underline{y}(k-1) \\ \underline{u}(0) \ldots \underline{u}(k-1) \end{bmatrix}$$

Normally, this equation cannot be satisfied exactly since the $\underline{y}(k)$ are not only functions of $\underline{y}(k-1)$ and $\underline{u}(k-1)$ but also of the (n-m) state variables which are not directly measured. However, a set $[F_r \; E_r]$ can be found, such that given the sequences

$$\underline{y}(0) \ldots \underline{y}(k-1)$$

$$\underline{u}(0) \ldots \underline{u}(k-1)$$

it will generate a sequence $[\omega(1) \ldots \omega(k)]$ which minimizes

$$e(k)=tr\{ [\underline{y}(1)-\omega(1), \underline{y}(2)-\omega(2), \ldots \underline{y}(k)-\omega(k)]^T [\underline{y}(1)-\omega(1) \ldots \underline{y}(k)-\omega(k)]\}$$

and which lies in the row space of the given sequence. The solution is well known and is given by

$$[F_r \; E_r] = [\underline{y}(1) \ldots \underline{y}(k)] \begin{bmatrix} \underline{y}(0) \ldots \underline{y}(k-1) \\ \underline{u}(0) \ldots \underline{u}(k-1) \end{bmatrix}^+$$

where + denotes the pseudo-inverse of a matrix. **The pseudo-inverse is the inverse of a non-square matrix.**

Anderson's method can be extended to continuous systems. Since the test system to be described later is single input-output, D is a vector and hence [y(1), y(2) ... y(k)] is a vector and not equal to the matrix $[\underline{x}(1), \underline{x}(2), \ldots \underline{x}(k)]$

For the continuous system

$$[\dot{\underline{x}}(0), \dot{\underline{x}}(T), \ldots \dot{\underline{x}}(kT)] = [A\ B] \begin{bmatrix} \underline{x}(0), \ldots \underline{x}(kT) \\ \underline{u}(0), \ldots u(kT) \end{bmatrix}$$

The reduced system is to satisfy

$$[\dot{\underline{x}}(0), \dot{\underline{x}}(T), \ldots \dot{\underline{x}}(kT)] = [A_r\ \ B_r] \begin{bmatrix} \underline{x}(0), \ldots \underline{x}(kT) \\ u(0), \ldots u(kT) \end{bmatrix}$$

as closely as possible. The dimension of $\underline{x}(kT)$ is m, the order of the reduced model. The solution, as before, is of the form of a pseudo-inverse solution

$$[A_r\ B_r] = [\dot{\underline{x}}(0), \ldots \dot{\underline{x}}(kT)] \begin{bmatrix} \underline{x}(0), \ldots \underline{x}(kT) \\ u(0), \ldots u(kT) \end{bmatrix}^{+}$$

that is

$$[A_r\ B_r] = [\dot{\underline{x}}(0), \ldots \dot{\underline{x}}(kT)] \begin{bmatrix} \underline{x}(0), \ldots \underline{x}(kT) \\ u(0), \ldots u(kT) \end{bmatrix}$$

$$\left\{ \begin{bmatrix} \underline{x}(0), \ldots \underline{x}(kT) \\ u(0), \ldots u(kT) \end{bmatrix} \begin{bmatrix} \underline{x}(0), \ldots \underline{x}(kT) \\ u(0), \ldots u(kT) \end{bmatrix}^{T} \right\}^{-1}$$

If the sequences of $\underline{x}$ and $\dot{\underline{x}}$ are calculated for the input sequence u, (u being constant between samples) then $[A_r\ B_r]$ can be directly solved. As can be seen from the above brief description, all system parameters must be known and the system vector differential equation solved for kT samples. However, computationally this method is much

easier than the previous two methods discussed, as it does not require computation of eigenvalues and eigenvectors. It also optimizes the trajectory fit by minimizing the mean squared error between the system and model trajectories. Unfortunately, it does require considerable data storage, especially if the system has a long settling time. Also, as the minimization is done over a finite time interval, the steady-state error between the two trajectories is not forced to be zero.

2.4 Iterative Reduction Method

Sinha and Pille [5] proposed a method of system model reduction that requires only the measured input-output data for the system at the sample points. This method is based on an iterative application of the matrix pseudo-inverse algorithm. It determines the model of a specified order that minimizes the mean squared error between the responses of the system and the model to a given input.

The reduced discrete model may be expressed in terms of the pulse transfer function

$$H(z) = \frac{C(z)}{R(z)} = \frac{a_0+a_1z^{-1}+ \ldots +a_mz^{-m}}{1 -b_1z^{-1} \ldots -b_nz^{-n}}$$

or the equivalent difference equation

$$c_i = \sum_{j=0}^{m} a_j r_{i-j} + \sum_{j=1}^{n} b_j c_{i-j} \qquad \ldots(2.4\text{-}1)$$

where r(iT) and c(iT) are the input and response of the system at t=iT, and T is the sampling interval.

If the parameter vector $\underline{\phi}$ is defined as

$$\underline{\phi}^T = [a_0, a_1, \ldots a_m\ b_1, b_2, \ldots b_n]$$

and i ranges from 1 to some integer k

$$A_k\phi = \underline{c}_k$$

A_k is the information matrix, whose i^{th} row corresponds to equation 2.4-1, and

$$\underline{c}_k^T = [c_1, c_2, \ldots c_k]$$

Again, as for Anderson's method the solution which minimizes the mean squared error between the system and model responses is given by

$$\hat{\underline{\phi}}_k = A_k^+ \underline{c}_k$$

where A_k^+ is the pseudo-inverse of A_k.

To solve the above equation requires storage of a large amount of data, especially if the mean squared error is to be minimized over a sufficiently large interval with a reasonably high sampling rate. To over-come this problem, a recursive algorithm is developed where a row is added to A_k and an alement to $\underline{c}_k$ for each addi-

tional pair of input-output data,

that is

$$A_{k+1} = \begin{bmatrix} A_k \\ \underline{a}^T_{k+1} \end{bmatrix}$$

$$\underline{c}_{k+1} = \begin{bmatrix} \underline{c}_k \\ c_{k+1} \end{bmatrix}$$

where

$$\underline{a}_{k+1} = [r_{k+1}\ F_k \cdots r_{k-m+1}\ c_k\ c_{k-1} \cdots c_{k-n+1}]$$

Then the recursive algorithm for $k \geq m+n+1$

$$\hat{\underline{\phi}}_{k+1} = \hat{\underline{\phi}}_k + \frac{P_k \underline{a}_{k+1}(c_{k+1} - \underline{a}^T_{k+1}\hat{\underline{\phi}}_k)}{1 + \underline{a}^T_{k+1} P_k \underline{a}_{k+1}}$$

$$P_{k+1} = P_k - \frac{P_k \underline{a}_{k+1}[P_k \underline{a}_{k+1}]^T}{1 + \underline{a}^T_{k+1} P_k \underline{a}_{k+1}}$$

The algorithms require no matrix inversion and the number of input-output data points required in storage is just $m + n + 2$.

To start the algorithm, $\hat{\underline{\phi}}_k$ and P_k must be determined for the case where $k = p = m + n + 1$. This can be done since A_k is a square matrix

and the solutions are given by

$$\hat{\underline{\phi}}_p = A_p^{-1} \underline{c}_p$$

$$P_p = [A_p^T A_p]^{-1}$$

Recursive relationships have also been established so that matrix inversion may be avoided completely.

The corresponding continuous-time system transfer function H(s) can be determined from the low-order pulse transfer function H(z). If the input between sampling instants is held constant, H(z) may be regarded as the z- transfer function of H(s) preceded by a sampler and a zero-order hold. This requirement is met in the application of this method to the test system, when the input is taken as a unit step.

Although this method is based on the same principle as the Anderson method, the implementation is much easier. In addition to requiring only the input-output data at sample points, the requirement for storage of this data depends only on the order of the assumed model, and not the number of iterations considered. Also, since this method requires no matrix inversions, it yields better results than those obtained by the other methods discussed. However, as for Anderson's method, steady-state error between the system and model responses is not forced to be zero.

2.5 Pattern Search Reduction Method

In the two trajectory-fitting reduction methods discussed, the objective function to be minimized is constrained to be the sum of squares of errors. Sinha and Bereznai [6] proposed a method of reduction which can provide an optimum low-order model with respect to any specified error criterion. If the output trajectory of the system given by equations 2.1-1 is given by the sequence

$$\{ \underline{y}(0), \underline{y}(1), \ldots \underline{y}(i), \ldots \underline{y}(I) \}$$

where $\underline{y}(i) = \underline{y}(t_i)$,

a reduced model, given by equations 2.1-2 is determined such that its output trajectory given by the sequence

$$\{ \underline{y}_r(0), \underline{y}_r(1), \ldots \underline{y}_r(i), \ldots \underline{y}_r(I) \}$$

minimizes the scalar error function

$$J = f[\omega_i^T (\underline{y}(i) - \underline{y}_r(i))]$$

where ω_i^T is a weighting vector.

Alternatively, for a given value of J, the lowest order model (m) is determined such that the resultant error is less than or equal to J.

Since the choice of error criterion has a direct effect on the parameters of the reduced model, it is important which criterion is chosen for minimization. In current practice the error function takes on one of two forms. For the single input-output case the error function can be of the form

$$J = \sum_{i=0}^{I} \omega_i || y(i)-y_r(i)||^p$$

which is the sum of the weighted norms of the output error raised to some power p. This function can represent the area between the output curves or the mean squared error, depending on the values of ω_i and p. The error function can also take the form

$$J = \max_{i=0,I} \{ \omega_i || y(i)- y_r(i)|| \}$$

which retains the value at one particular sample only, where the deviation is maximum. Another form of this function can be used to minimize the maximum perpendicular distance between the output curves, which may be more useful than minimizing the maximum vertical error.

After the error function to be minimized has been chosen, a pattern search technique is employed to find the model which minimizes this function. Depending on whether the step response of the system has overshoot or not, a simple second order model with a pair of complex conjugate poles, or a simple first order model, is chosen as starting point. If the optimum set of parameters has

been found by the pattern search program for a particular model order, the order is increased by one and a new pattern search initiated. This process is continued until the error criterion is satisfied, or the desired order is reached.

As for the previous method discussed (Sinha and Pille) the parameters of the system need not be known, and the output sequence can be obtained either by solution of the vector differential equation at specified time intervals, or direct measurement of the system output. This method requires considerably more computer time than the iterative reduction method, and limitations arise because of the poor convergence properties of the pattern search algorithm. However, the flexibility provided by the choice of the criterion of optimization, may often lead to a model which is more acceptable than that obtained using the least squares criterion. The pattern search method can also be used to optimize, with respect to a specific error criterion, a reduced model derived using some other reduction method. A final advantage is thataswith the iterative reduction method, it can be used in system identification.

2.6 Step Response Reduction Method

Fellows, Sinha and Wismath [7] proposed a method of model reduction, which allows direct calculation of the parameters of a second order reduced model, from the step response of the system. The features of the step response of a system most commonly specified are (i) M- maximum overshoot in the response, (ii) t_p - time required to reach first peak, (iii) S_0 - initial slope of the

response curve, (iv) A- steady-state response to unit step, (v) t_1 - time required for the response to first reach half of the steady state value, and (vi) S_1 - slope of the curve at t_1. A second order model is then found which will meet these specifications precisely.

Depending on pole - zero configurations, second order systems can be divided into four classes. As most physical systems are normally of the low-pass type, with more poles than zeros, the case with two finite zeros is not considered. The four classes, along with their transfer functions, can be written as:

(i) System with two real poles and no finite zero

$$G_1(s) = \frac{K}{(s+\alpha)\ (s+\beta)}$$

(ii) System with a pair of complex conjugate poles and no finite zero.

$$G_2(s) = \frac{K}{(s+\alpha)^2+\beta^2}$$

(iii) System with two real poles and one finite zero,

$$G_3(s) = \frac{K(s+\delta)}{(s+\alpha)\ (s+\beta)} \qquad \text{and}$$

(iv) System with a pair of complex poles and one finite zero.

$$G_4(s) = \frac{K(s+\delta)}{(s+\alpha)^2+\beta^2}$$

where K, α, β, and δ are real numbers and α and β

are constrained to be positive to ensure stability. $G_1(s)$ and $G_2(s)$ have zero initial slope and in addition $G_1(s)$ represents an overdamped system. The step response of each second order model is well known and if written in the time domain, can be used to solve the model parameters in terms of the specified step response features of the original system. For example, for a system response which has zero initial slope and overshoot, $G_2(s)$ would be chosen. Its time response and its derivative are given by

$$c(t) = \frac{K}{\alpha^2+\beta^2} - \frac{K}{\beta\sqrt{\alpha^2+\beta^2}} e^{-\alpha t} \text{Sin}(\beta t + \tan^{-1}\beta)$$

$$\frac{dc(t)}{dt} = \frac{K}{\beta\sqrt{\alpha^2+\beta^2}} e^{-\alpha t}[\alpha \text{Sin}(\beta t + \tan^{-1}\beta/\alpha) - \beta \text{Cos}(\beta t + \tan^{-1}\beta/\alpha)]$$

If the system step response features to be satisfied are given as (A), (M), and (t_p) the model parameters can be derived directly from the resulting equations:

$$\alpha = \frac{1}{t_p} \ln \left(\frac{A}{M-A}\right)$$

$$\beta = \frac{\pi}{t_p}$$

$$K = A(\alpha^2+\beta^2)$$

The other classes of second order models result in transcendental equations, which although more difficult, can still be solved. This reduction method's strongest feature is its simplicity. It requires no knowledge of the system other than its unit step response and with the use of a transfer function matrix, can be used to reduce multiple input-output system models. Although it does not optimize the parameters, in order to get the closest fit between trajectories, it can stress certain features of the system step response. Since the method requires very little computation, the model derived can be used as the first approximation for an optimization procedure, such as the pattern search method previously mentioned.

2.7 Chen & Shieh Method

Chen and Shieh [8] proposed a method of reduction based on the continued-fraction expansion of the system transfer function, in polynomial form, starting from the constant terms. If the system transfer function is given by

$$G(s) = \frac{b_0+ b_1s^1+ \ldots + b_ks^k}{a_0 + a_1s^1+ \ldots+a_ns^n}$$

the continued fraction is

$$G(s) = \cfrac{1}{H_1 + \cfrac{S}{H_2 + \cfrac{S}{H_3 + \cfrac{S}{H_4 + \cfrac{S}{\ddots}}}}} \qquad \ldots(2.7\text{-}1)$$

This continued-fraction expansion is equivalent to a Taylor-series expansion about s = 0. Also from a consideration of the final value theorem, it follows that the quotients in the expansion are in order of decreasing significance of their contributions to the response as steady state is approached. Truncating the continued fraction at a suitable stage gives a reduced system transfer function of the desired accuracy. For a reduced model of order m, equation 2.7-1 is truncated after 2 m terms. Computationally, the application of this method is quite simple and can even be done by hand. Although the transfer function parameters must be known, the poles and zeros need not be calculated. The resulting model gives the correct steady-state response, but the approximation to the initial transient response may not be good. Furthermore, the stability of the model is not guaranteed, even if the original system is stable.

Chuang[9] proposed a modification of this method so that the initial system response may be more closely modeled. It consists of having alternate continued-fraction expansion from constant terms

first and then from highest-order terms. This is equivalent to Taylor series expansions about s = 0 and s = ∞. The resulting expansion is of the form

$$G(s) = \cfrac{1}{H_1 + \cfrac{s}{H_2 + \cfrac{1}{H_3 + \cfrac{s}{H_4 + \cfrac{1}{\ddots}}}}}$$

Again 2 m terms are required for a reduced model of order m. Although this modification may improve the Chen & Shieh model, it can not be used for those systems for which the number of poles exceeds the number of zeros by more than one. That is

$$n - k \geq 1$$

For these cases some of the coefficients become zero and hence the model is not as accurate.

The model reduction methods described in this chapter represent as many different techniques as possible. They have been applied to the same test system by Fellows, Sinha and Wismath [7], Sinha and Pille[5], and Sinha and Bereznai[6]. The models derived and their application to optimal control will be given in Chapter 4.

CHAPTER 3

OPTIMAL CONTROL OF A LINEAR DYNAMIC SYSTEM

3.0 Introduction

In modern control systems, it is desirable to control the system in such a way that a certain criterion is maximized or minimized. Much theory has been developed in recent years, so that this optimal control may be found, with some degree of facility. This optimal control may either take the form of an open-loop driving function or sequence, or a closed-loop feedback control. As this theory can be found in most texts dealing with modern control systems [12,13] it will only be briefly described in this chapter. Although both open and closed-loop optimal control are discussed, the greater emphasis is on closed-loop or feedback implementation of optimal control.

3.1 General Optimal Control

In optimal control theory the basic problem is to find the optimal control u(t) or the optimal control law

$$u = k[\underline{x}(t),t]$$

which transfers the system given by

$$\dot{\underline{x}} = f(\underline{x}, u, t)$$

from some given initial state to some final state, while minimizing the integral cost function

$$J = \int_{t_i}^{t_f} L(\underline{x}, u, t)\, dt$$

where t_i and t_f are the initial and final times and L is a positive definite function of x, u, and t.

It has been shown [12] that solution of the above problem requires solving the Euler equations, subject to the boundary conditions of the problem. To simplify solution of the Euler equations, the Pontryagin state function

$$H(\underline{x}, u, \lambda, t) = \lambda^T f(\underline{x}, u, t) + L(\underline{x}, u, t)$$

is formed, where the λ are Lagrangian multiplier or costate functions. The Euler equations can then be solved to find the optimal control.

While theoretically straightforward, the solution of the resulting Euler differential equations presents several practical problems. The equations are in general, non-linear, and time varying and require numerical solution on a digital computer. Although this problem is not serious, it is almost

insignificant compared to the two-point boundary-condition problem that is also involved. If the system is of order n, 2n boundary conditions will be given for the solution of the resulting 2n differential equations. However, for the basic case where $\underline{x}(t_i)$ and $\underline{x}(t_f)$ are specified, n of these conditions are given at the initial time and n at the terminal time. It is not possible to integrate the differential equations forward in time from the initial conditions or backward in time from the terminal conditions. More generalized boundary conditions make the problem even more computationally difficult. This problem, although it can be solved using several elegant computational schemes developed, yields an open-loop optimal control valid only for that particular set of boundary conditions. If either the initial state is changed, or any disturbances act on the system the control u(t) found is no longer optimal. This is a weakness of open-loop optimal control and a more desirable solution is the closed-loop or feedback optimal control.

3.2 Closed-Loop Optimal Control With Quadratic Cost

By combining the Pontryagin method with concepts from the second method of Liapunov, it is possible to remove the necessity for solving the two-point boundary condition problem, for the particular case of a linear system with a quadratic cost function.

This approach also yields an optimal control law $u(\underline{x},t)$. This optimal control is valid for a family of initial conditions, and since it is a function of $\underline{x}$, results in closed-loop optimal control, thus minimizing the effects of system disturbances. It has been shown [12] that by defining $V(\underline{x}, t)$ as the minimum value of the cost function for an initial state $\underline{x}$ at time t, that is

$$V(\underline{x},t) = \int_{t}^{t_f} L[\underline{x}(\tau), u(\underline{x},\tau), \tau]d\tau \qquad \ldots(3.2\text{-}1)$$

the necessity of solving a two-point boundary condition problem is removed. The optimal control can be found in terms of $\underline{x}$, $\nabla V(x,t)$ and t, and $V(x,t)$ found by solving the Hamilton-Jacobi equation

$$H[x,\nabla V(x,t),t] + \frac{\partial V(x,t)}{\partial t} = 0$$

where ∇ represents differentiation with respect to $\underline{x}$. Although it is almost impossible to solve the Hamilton-Jacobi equation even for trivial problems, it is possible to obtain a solution in a relatively straightforward manner for one problem of significant practical importance.

This problem, known as the linear control problem, is the optimal control of the linear system

$$\dot{\underline{x}} = A\underline{x} + B\,u$$

with a cost function

$$J = \int_0^T (\underline{x}^T Q \underline{x} + u^T P u)\, dt \qquad \ldots.(3.2\text{-}2)$$

where Q is at least positive semidefinite symmetric, P is positive definite symmetric, and u is unconstrained. Although more general quadratic cost functions have been developed [13], these give rise to much greater computational difficulty, and as a result will not be considered here.

It has been shown also, that if V(x,t) is given by

$$V(\underline{x},t) = \underline{x}^T R(t) \underline{x} \qquad \ldots.(3.2\text{-}3)$$

where $\underline{x}$ is the initial state of the system, the Hamilton-Jacobi equation can be solved to yield the well known matrix Riccati equation

$$\dot{R}(t) + Q - R(t)\, BP^{-1}B^T R(t) + R(t)\, A + A^T R(t) = 0 \qquad \ldots(3.2\text{-}4)$$

Since for fixed terminal state,

$$V(\underline{x},t) = 0 \quad \text{for } \underline{x} = \underline{x}(t_f) \text{ and } t=t_f$$

and for the free terminal state,

$$V(\underline{x},t_f) = 0 \quad \text{for any } \underline{x}$$

$R(t)$ is a positive definite symmetric matrix for all $t<t_f$ and

$R(t_f) = 0$

Although analytic methods are available for solving equation 3.2-4, they are unwieldy for any systems of higher than second order. On the other hand a solution may be obtained easily on a digital computer, if the equation is integrated backward in time from the known terminal condition, over the time of interest. The optimal control, found by differentiating the Pontryagin state function with respect to u can then be given as

$$u(\underline{x},t) = -K^T(t)\underline{x} \qquad \ldots.(3.2\text{-}5)$$

where $K^T(t) = P^{-1}B^TR(t)$

The elements of K(t) are referred to as feedback coefficients, since the optimal control consists of a time-weighted linear combination of the state variables.

If the time limit in equation 3.2-2 is taken as ∞ rather than T, the R matrix becomes constant, since for an infinite time interval $V(\underline{x},t_1) = V(\underline{x},t_2)$ for finite t_1 and t_2. If an analytic solution of equation 3.2-4 is known, the constant R matrix may be found by performing a limit operation on the solution. The constant R matrix may also be found by integrating the matrix Riccati-equation backward in time, from the known terminal condition $R(\infty) = 0$ until a steady-state solution is reached. Alternatively, if R is constant $\dot{R} = 0$

and hence equation 3.2-4 becomes the reduced or degenerate Riccati equation

$$A^T R + R A - R B P^{-1} B^T R + Q = 0 \qquad \ldots(3.2\text{-}6)$$

For this method, the solution of $n(n+1)/2$ nonlinear algebraic equations is required. In addition, to obtain a unique solution, the requirement that R be positive definite must be enforced.

In this chapter a few of the concepts of optimal control have been mentioned. The reasons for the desirability of using linear optimal control, or feedback optimal control with quadratic cost, have also been mentioned. Because of these reasons only feedback optimal control will be considered in the following chapters. The next chapter deals with the application of these concepts to provide near-optimal control of the test system.

CHAPTER 4

APPLICATION OF REDUCED MODELS IN THE CONTROL OF A LINEAR DYNAMICAL SYSTEM

4.0 General

Very little work has been done to date, in using a reduced model to find the optimal or sub-optimal control of the original system. Mitra [10] has done some analytical work in this area,but considers only the optimal-projection reduction method.

This chapter deals with the use of reduced models to provide suboptimal control of a test system. The models, derived using the reduction methods described in Chapter 2, are first listed, and some comments made regarding the model characteristics. Their application to the control of the test system is then discussed. Finally, some comments are made regarding those model properties which affect their application to optimal control.

4.1 The Reduced Models of the Test System

Fellows, Sinha, and Wismath [7], have used the modal reduction mthods, Anderson's Method, the Chen and Shieh Method and the step response method to derive reduced models for the test system, described in Appendix A. Sinha and Pille [5] have also derived a reduced model for this test system using the iterative reduction method. Sinha and Bereznai [6] have derived

a number of reduced models for the same system, using various criteria for optimization and the pattern search reduction method. Of these, four were selected for use in sub-optimal control of the test system. These models, chosen to represent as many different error criteria as possible, are as follows:

(a) the model which minimizes the maximum perpendicular error between system and model responses (minimax $\perp$),

(b) the model which minimizes the sum of the absolute values of the sample error ($\Sigma|e|$),

(c) the model which minimizes the sum of the squares of the sample error ($\Sigma\ e^2$), and

(d) the model which minimizes the sum of the squares of the sample errors with no error in steady-state response to a step input.

All models mentioned above, were constrained to be second order models, since, for a proper comparison all reduced models would have to be of the same order. These models, along with their pole locations appear in Table I.

As can be seen from Table 1, the two modal reduction methods have retained the pair of system complex poles nearest the $j\omega$ axis.

TABLE I

REDUCED MODELS OF THE TEST SYSTEM

Method of Reduction	Transfer Function	Pole Locations
Davison's Method	$\frac{-.050525s + .55576}{s^2 + 4.112593s + 5.02966}$	$-2.05297 \pm j0.895322$
	Steady-State Unit Step Response=.1105	
Optimal Projection Method	$\frac{-.015929s + .56478}{s^2 + 4.0488s + 5.0277}$	$-2.02438 \pm j0.96465$
	Steady-State Unit Step Response = .11233	
Anderson's Method	$\frac{.30961}{s^2 + 1.902574s + 2.687909}$	$-.9512 \pm j1.335$
	Steady-State Unit Step Response = .1152	
Sinha and Pille Iterative Method	$\frac{.3302}{s^2 + 2.0954s + 2.8886}$	$-1.0477 \pm j1.3375$
	Steady-State Unit Step Response = .114	
Sinha and Bereznai Pattern Search (minimax $\perp$)	$\frac{.0254s + 0.2967}{s^2 + 2.4257s + 2.5581}$	$-1.213 \pm j1.043$
	Steady-State Unit Step Response = .1160	

TABLE I (Continued)

Sinha & Bereznai Pattern Search ($\Sigma\|e\|$)	$\frac{.1536s + 0.01329}{s^2 + 1.3456s + 0.1196}$	-.0957, - 1.250
Steady-State Unit Step Response = .1112		
Sinha & Bereznai Pattern Search (Σe^2)	$\frac{.3960}{s^2 + 2.6569s + 3.4191}$	$-1.328 \pm j1.286$
Steady-State Unit Step Response = .1158		
Sinha & Bereznai Pattern Search (Σe^2 with s.s. constraint)	$\frac{0.1019s + .05359}{s^2 + 1.0718s + 0.4823}$	$-0.536 \pm j0.442$
Steady-State Unit Step Response = .1111		
Step Response Method	$\frac{.209768}{s^2 + 1.690396s + 1.887915}$	$-0.84520 \pm j1.08331$
Steady-State Unit Step Response = .1111		
Chen & Shieh's Method	$\frac{0.1299s + 0.01105}{s^2 + 1.14644s + 0.09941}$	-1.04822, - 0.09822
Steady-State Unit Step Response = .1112		

SYSTEM STEADY-STATE UNIT STEP RESPONSE = .11111

The authors[7] felt that the effect of the real pole of the system which is much closer to the origin, could be neglected without much error, as there is also a real zero very near to it. As the other reduction methods do not consider the poles or eigenvalues of the original system, new poles have been generated so that they are able to duplicate more closely the response of the test system.

The fact that a mean squared error criterion inherently stresses the closeness of fit between the transient part of the system and model responses, is also confirmed in Table 1. The models derived using Anderson's method, Sinha and Pille's method and Sinha and Bereznai's Σe^2 method, which all employ mean squared error minimization techniques, have significant steady-state step response errors. In contrast, those which stress steady-state response such as Sinha and Bereznai's $\Sigma|e|$ method and Chen and Shieh's method, have insignificant steady-state step response errors.

Those features of the reduction methods and their resulting models, which affect their application to near-optimal control of the test system, will be discussed in later chapters. The next section deals with the use of the reduced models in controlling the test system.

4.2 Use of Reduced Models in Open-Loop Sub-Optimal Control

As stated in Chapter 2, some of the reduction methods use trajectory fitting techniques to derive the reduced model, that is, they derive a model whose output trajectory approximates that of the original system, if both are driven by the same input. It was decided therefore, to investigate the response of the system, if it were driven by the optimal control derived for each model.

In order to use the optimal control theory discussed in Chapter 3, it was necessary to represent each model in phase variable form, as described in Appendix C. A cost function was then selected, and the optimal feedback coefficients found for each model. The details of the derivation of the feedback coefficients, for each type of reduced model are also given in Appendix C. The reduced model was then driven from some initial state $\underline{x}_r(t_i)$ to some final state $\underline{x}_r(t_f)$, using closed-loop optimal control. The resulting model optimal control

$$u_r(\underline{x}_r) = -K_r^T \underline{x}_r(t) \qquad .....(4.2\text{-}1)$$

was then used to drive the original system, using system initial and final conditions equivalent to those of the model. This sub-optimal control was applied to the system in an open-loop sense, as shown in Figure 4.1.

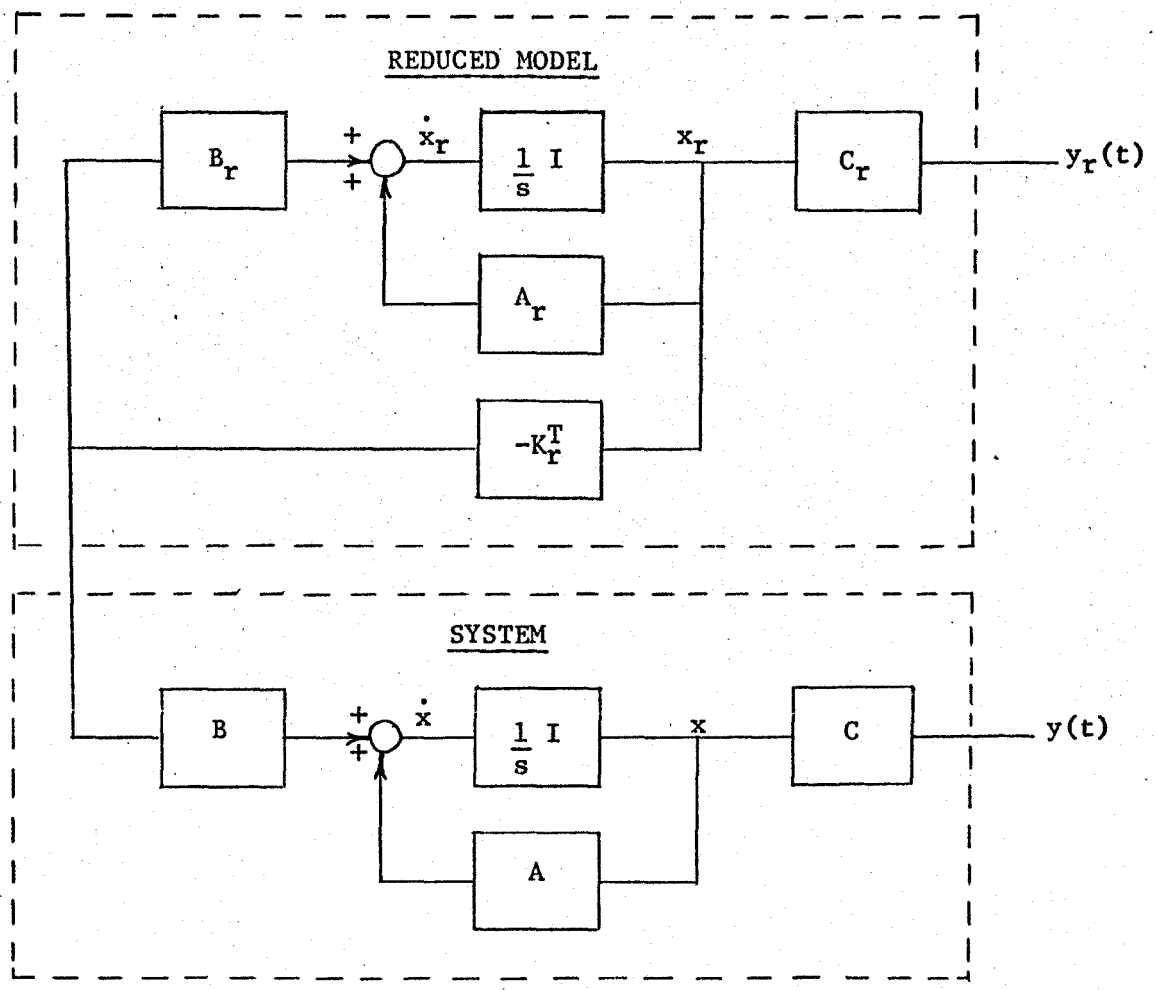


Figure 4.1

Block Diagram of Open-Loop
Sub-Optimal Control of a
Linear Dynamical System

For this method of control, the quadratic cost function selected was

$$J_1 = \int_0^\infty (x_1(t)^2 + u(t)^2)dt \qquad .....(4.2\text{-}2)$$

where $x_1(t) = y(t)$ for the system

$x_1(t) = y_r(t)$ for the model.

As stated in Chapter 3, the infinite time interval results in constant feedback coefficients, hence simplifying considerably, the determination of the model optimal control $u_r(\underline{x}_r)$. Each model was driven from

$$\underline{x}_r^T(0) = [1\ 0] \qquad \text{to} \qquad \underline{x}_r^T(\infty) = [0\ 0]$$

and the model response and optimal control calculated at .04 second intervals. In order to drive the system to zero, a total time interval of 80 seconds was required. The model optimal control was therefore calculated for a time interval of 80 seconds. Each model optimal control was then used to drive the system from

$$\underline{x}^T(0) = [\ 1\ 0\ \ldots\ 0] \quad \text{to } \underline{x}^T(80) = [\ 0\ 0\ \ldots\ 0]$$

and the system response obtained for the same 80 second time interval. In addition, the cost function given by equation 4.2-2, was evaluated for the system for each case of sub-optimal control.

To provide a basis for comparison, the minimum cost was found for the system by obtaining the system optimal feedback coefficients, using the method described in Appendix C.2. The system was again driven from

$$\underline{x}^T(0) = [1\ 0\ \ldots\ 0] \quad \text{to } \underline{x}^T(80) = [\ 0\ 0\ \ldots\ 0],$$

this time using closed-loop optimal control and the cost calculated. This minimum cost can also be directly evaluated from equation 3.2-3, and for this set of initial conditions is simply equal to R_{11}, the first element of the R matrix solution of equation 3.2-4 for the test system. The system and model responses and costs were calculated using the methods and program described in Appendix B. The results of this investigation are summarized in Table 2 in Chapter 5.

4.3 Use of Reduced Models in Closed-Loop Sub-Optimal Control

In the previous section, a method was described, which uses a reduced model to provide sub-optimal control of the original system. However, since the resulting control is open-loop, this method has certain undesirable features. As mentioned in Chapter 3, noise acting on the system will cause this control to be a poorer approximation of the optimal. In addition, as the initial state of the system changes, the initial state of the reduced model must also

be changed accordingly. This requires a new calculation of the model optimal control, for each set of initial conditions for the system.

To circumvent these limitations, it was decided to attempt driving the system using some form of feedback optimal control. Since the reduced model is an approximation of the system, the model optimal feedback coefficients can also be regarded as an approximation of the system optimal feedback coefficients. Once the model feedback coefficients have been calculated for a particular cost function, they can be used to provide closed-loop sub-optimal control of the system. These coefficients are used to provide a feedback path, as shown in Figure 4.2 and are valid for a family of system initial conditions. The feedback coefficient vector K' is given by

$$K_s^T = [K_{r1}, K_{r2}, 0, \ldots 0]$$

where K_{r1} and K_{r2} are the optimal feedback coefficients derived for the reduced model. It waa decided to minimize the same cost function as before, given by equation 4.2-2. The required model coefficients are therefore those derived for section 4.2, that is open-loop sub-optimal control.

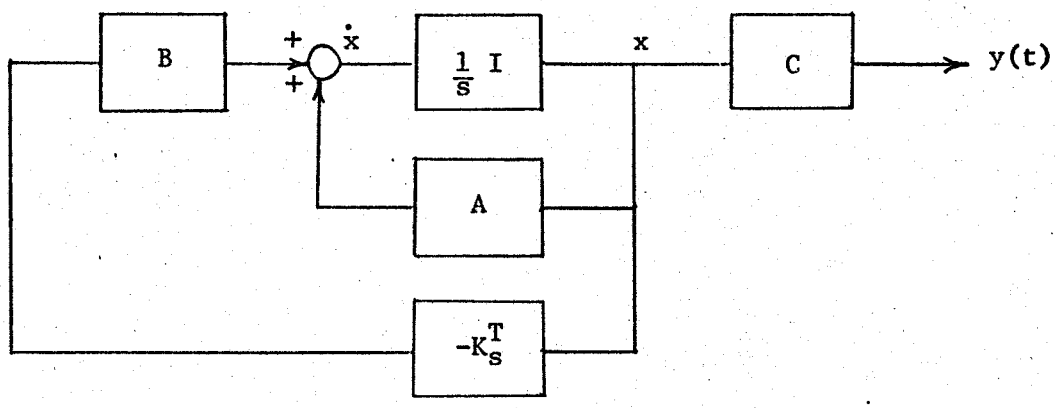


Figure 4.2

Block Diagram of Closed-Loop
Sub-Optimal Control of a
Linear Dynamical System

For each pair of model coefficients, the system was driven from

$$\underline{x}(0)^T = [\ 1\ 0\ \ \ldots\ 0]\ \text{ to }\ \underline{x}(80) = [\ 0\ \ldots 0]$$

using the configuration of Figure 4.2. The system response

$x_1(t) = y(t)$ and the sub-optimal control

$$u(t) = K_s^T \underline{x}(t)$$

were again calculated at .04 second intervals, for a total time interval of 80 seconds. The cost function given by equation 4.2-2 was calculated for each case. The details of the system response and cost function calculations are given in Appendix B. The results of this investigation are summarized in Table 3, in Chapter 5. In addition to the reasons already mentioned previously, this method of control is desirable because of its simplicity of application.

As can be seen from the results and discussion given in Chapter 5, the cost function

$$J_1 = \int_0^\infty (x_1(t)^2 + u(t)^2)dt$$

seems to be somewhat insensitive to feedback coefficient variation, at least for this particular test system. It was decided therefore, to investigate near-optimal control of the test system using another cost function. The cost function chosen was

$$J_2 = \int_0^\infty (x_1(t)^2 + 10\ x_2(t)^2 + .1\ u(t)^2)dt \quad ....(4.2\text{-}3)$$

if

$$x_1(t) = y(t) \quad .... (4.2\text{-}4)$$

and $x_2(t) = \dot{y}(t)$ ....(4.2-4)

this cost function can be represented by

$$J_3 = \int_0^{\infty} (y(t)^2 + \dot{y}(t)^2 + u(t)^2)dt \quad ....(4.2-5)$$

where y(t) is either the system or model output.

Since some reduction methods employ trajectory-fitting techniques, the model output and its derivative with respect to time, are a good approximation of the system output and its derivative. Also, since for sub-optimal control, both model and system cost functions must be the same, it was felt that the best choice of cost functions are those whose terms are limited to the output, its derivative with respect to time and the control input. This requires that the state variables for both system and reduced models be in phase variable form. That is, the n state variables for a system of order n, must represent the output and its n-1 derivatives.

From appendix A.1, it can be seen that for the test system, the conditions of equations 4.2-4 are met by modifying the control vector B to include the effects of the system zero.

These conditions are also satisfied for those reduced models which have no zero. However, as also shown in Appendix C.2, if the reduced model transfer function is given by

$$G_r(s) = \frac{a\,s + b}{s^2 + ds + c} \qquad \ldots\ (4.2\text{-}6)$$

even modifying the control vector B_r will not satisfy the conditions of equations 4.2-4. For these models

$$\begin{aligned} y(t) &= x_1(t) \\ \dot{y}(t) &= x_2(t) + a\,u(t) \end{aligned} \qquad \ldots\ldots (4.2\text{-}7)$$

If these results are substituted in equation 4.2-5 it is not possible to have a cost function of the general quadratic form given by equation 3.2-2. As a result, the cost functions for the system and reduced models without zeros are equal and can be given by equation 4.2-5. The cost function for those models having a zero is somewhat different from the system cost and can be given by equations 4.2-3 and 4.2-7.

The optimal feedback coefficients for both reduced models and system were again calculated using the appropriate cost functions as outlined in Appendix C.2. Unfortunately, with

the exception of one model, for the reduced models having a zero, either one of the feedback coefficients calculated was negative. As there is a possibility of the resulting closed-loop system becoming unstable, with a negative feedback coefficient, it was decided to define the state variables of these reduced models differently. If the reduced model transfer function given by equation 4.2-6 is written as

$$G_r(s) = \frac{x_1(s)}{u(s)} \cdot \frac{y(s)}{x_1(s)}$$

$$= \frac{b}{s^2+ds+c} \cdot ((a/b)s+1))$$

The state variable representation becomes

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -c & -d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ b \end{bmatrix} u$$

$$y(t) = \begin{bmatrix} 1 & a/b \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad .....(4.2\text{-}8)$$

The model optimal feedback coefficients were calculated for the models having a zero using the cost function given by equation 4.2-3, with the state variables defined by equation 4.2-8. Al-

though the cost function is different from that for the system, because of the definiton of the state variables, the resulting feedback coefficients are positive and can be used in sub-optimal control. In addition, this definition of the state variables, allows one to treat the model as having no zero and hence to derive the feedback coefficients directly, using the formulae given in Appendix C.1. The choice of state variable representation and resulting effects will be discussed further in the next chapter.

The system was driven from

$$\underline{x}^T(0) = [\ 1\ 0\ \ldots\ 0] \quad \text{to} \quad \underline{x}^T(80) = [\ 0\ \ldots\ 0]$$

using the system optimal feedback coefficients derived for the cost function J_3 given by equation 4.2-5, and J_3 was evaluated for the 80 second total time interval. Again, as stated for the first cost function considered, the minimum cost is also equal to R_{11}. The system was also driven sub-optimally from

$$\underline{x}^T(0) = [\ 1\ 0\ \ldots\ 0] \quad \text{to} \quad \underline{x}^T(80) = [\ 0\ \ldots\ 0]$$

using the configuration shown in Figure 4.2. The model feedback coefficients used, were those derived for the second cost function as explained earlier in this section. The sub-optimal cost given

by J_3, was also calculated for each pair of feedback coefficients. The system responses and costs for the above cases of optimal and sub-optimal control, were calculated using the method and program given in Appendix B, using time intervals of .04 seconds. The results of this investigation are summarized in Table 4 in Chapter 5.

As can be seen from Chapter 5, the results obtained for closed-loop sub-optimal control are better than those for the open-loop method. It was therefore decided that only the closed-loop method would be applied for the second cost function.

In summary, this chapter has dealt only with the procedures followed for each investigation. Discussion of the relative merits of the different procedures or techniques has been kept to a minimum in order to keep the chapter more coherent. For the same reason, discussion of the results has been limited to those details required for an explanation of the procedures followed. A thorough discussion of the results obtained for each investigation, is given in the next chapter.

CHAPTER 5

COMPARISON OF THE SUITABILITY OF REDUCED MODELS FOR NEAR-OPTIMAL CONTROL OF THE TEST SYSTEM

5.0 General.

In the preceding chapter, the ten reduced models were used to provide an approximation of the optimal control for the test system. The methods of application were described and some comments made regarding the model characteristics. In this chapter, the results are given for each method of sub-optimal control and cost function used. Those features of the reduction methods and their resulting models, which affect their application to near-optimal control of the test system are also discussed. The presentation of results follows the same order as that of the procedures described in Chapter 4. From the results obtained, some observations are made regarding the suitability of each method of sub-optimal control, and the suitability of each model for providing this control.

5.1 Results of the Cases of Sub-Optimal Control Investigated.

This section presents the results derived for the procedures outlined in Chapter 4. They are divided into three cases, (i) open-loop sub-optimal control using the cost function given by equation 4.2-2, (ii) closed-loop sub-optimal control using the same cost

function, and (iii) closed-loop sub-optimal control using the cost function given by equations 4.2-3 and 4.2-5.

The results for the open-loop sub-optimal control method are summarized in Table 2. The R matrix solutions are not given, as these are only used to calculate the model optimal feedback coefficients.

The results for the closed-loop sub-optimal control method using the cost function given by equation 4.2-2 are summarized in Table 3. The system optimal feedback coefficients are included for comparison. In both tables, the models are ordered according to their suitability for providing near-optimal control for the system.

The optimal system cost was calculated both by driving the system using optimal feedback control and evaluating equation 3.2-3. The results for each are identical, hence providing a check for the accuracy of the system response calculations.

The system optimal output trajectory is shown in Figure 5.1. Since the sub-optimal system trajectories are almost identical for the cost function considered, only two are shown in Figure 5.2. All the other system trajectories lie between these. The figure time interval was selected as 40 seconds, since for the rest of the

TABLE 2

RESULTS FOR OPEN-LOOP SUB-OPTIMAL CONTROL OF THE TEST SYSTEM

REDUCTION METHOD	SYSTEM COST
	OPT. SYSTEM COST
Pattern Search $\Sigma\|e\|$	1.00032
Chen & Shieh Method	1.00037
Pattern Search Σe^2 + s.s. constraint	1.00725
Pattern Search minimax $\perp$	1.01080
Optimal Projection Method	1.01093
Davison's Method	1.01101
Pattern Search Σe^2	1.01121
Step Response Meth.	1.01147
Iterative Method	1.01156
Anderson's Method	1.01171

Optimal System Cost = 6.2490

TABLE 3

RESULTS FOR CLOSED-LOOP SUB-OPTIMAL CONTROL OF THE TEST SYSTEM

REDUCTION METHOD	K_r^T FOR MODEL	SYSTEM COST / OPT. SYSTEM COST
Pattern Search min $\Sigma\|e\|$	[.11191 .04089]	1.0
Chen & Shieh Method	[.11151 .04802]	1.0
Pattern Search Σe^2 with s.s.	[.10248 .05136]	1.00009
Pattern Search minimax $\perp$	[.063007 .023785]	1.00219
Pattern Search Σe^2	[.057717 .021688]	1.00270
Anderson's Method	[.05740 .030097]	1.00280
Iterative Method	[.05697 .0271302]	1.00282
Opt. Projection Meth.	[.05403 .018382]	1.00302
Step Response Meth.	[.05538 .032698]	1.00304
Davison's Method	[.048960 .01338]	1.00360

SYSTEM OPTIMAL FEEDBACK COEFFICIENTS

[.11552 .06676 .016387 .001305 7.2799×10^{-5} 1.4469×10^{-6} 1.7307×10^{-8}]

control interval, the response approaches zero asymptotically.

The results for the closed-loop sub-optimal control method using the cost function given by equations 4.2-3 and 4.2-5, are summarized in Table 4. The model R matrix solutions, although not given, are those found if the model state space equations are given by 4.2-8. The optimal system costs, calculated by driving the system using optimal control and evaluating equation 3.2-3, are identical and are included in Table 4.

The R matrices, derived if the model state variables are defined by equation 4.2-7, are positive definite. However, the resulting feedback vector has one negative term because of the definition of the control vector elements. Positive feedback coefficients occur for the Pattern Search (minimax $\perp$) model and are given by

$$K_r^T = [1.7324 \quad 4.1704]$$

The sub-optimal system cost using these coefficients is 6.24710 and is the minimum sub-optimal cost found for both methods of model representation. The system optimal feedback coefficients are

$$K^T = [1.446 \quad 6.3039 \quad .80605 \quad .052908 \quad 1.8112\text{x}10^{-3} \quad 3\text{x}10^{-5} \quad 2.7\text{x}10^{-7}]$$

The system optimal output trajectory for the second cost function is shown in Figure 5.3 and the optimal control in Figure 5.4. Figures 5.5 and 5.6 show the sub-optimal trajectories and controls respectively, for the best and worst approximations of the optimal control.

TABLE 4.
RESULTS FOR CLOSED-LOOP SUB-OPTIMAL CONTROL OF THE TEST SYSTEM USING J2

REDUCTION METHOD	K_r^T FOR MODEL	SYSTEM COST / MIN. SYSTEM COST
Pattern Search $\Sigma\|e\|$	[.53943 .890801]	1.02206
Chen & Shieh Method	[.53960 .9483]	1.02251
Pattern Search Σe^2 with s.s.	[5394026 2.8064]	1.04006
Pattern Search minimax $\perp$	[.56163 4.8868]	1.06143
Step Response Method	[.53939 4.9831]	1.06625
Opt. Projection Method	[.54498 5.2135]	1.06843
Davison's Method	[.53657 5.11766]	1.06853
Pattern Search Σe^2	[.56088 5.4499]	1.06898
Iterative Method	[.5540 5.6385]	1.07272
Anderson's Method	[.55799 5.7446]	1.07351

Minimum System Cost = 6.17478

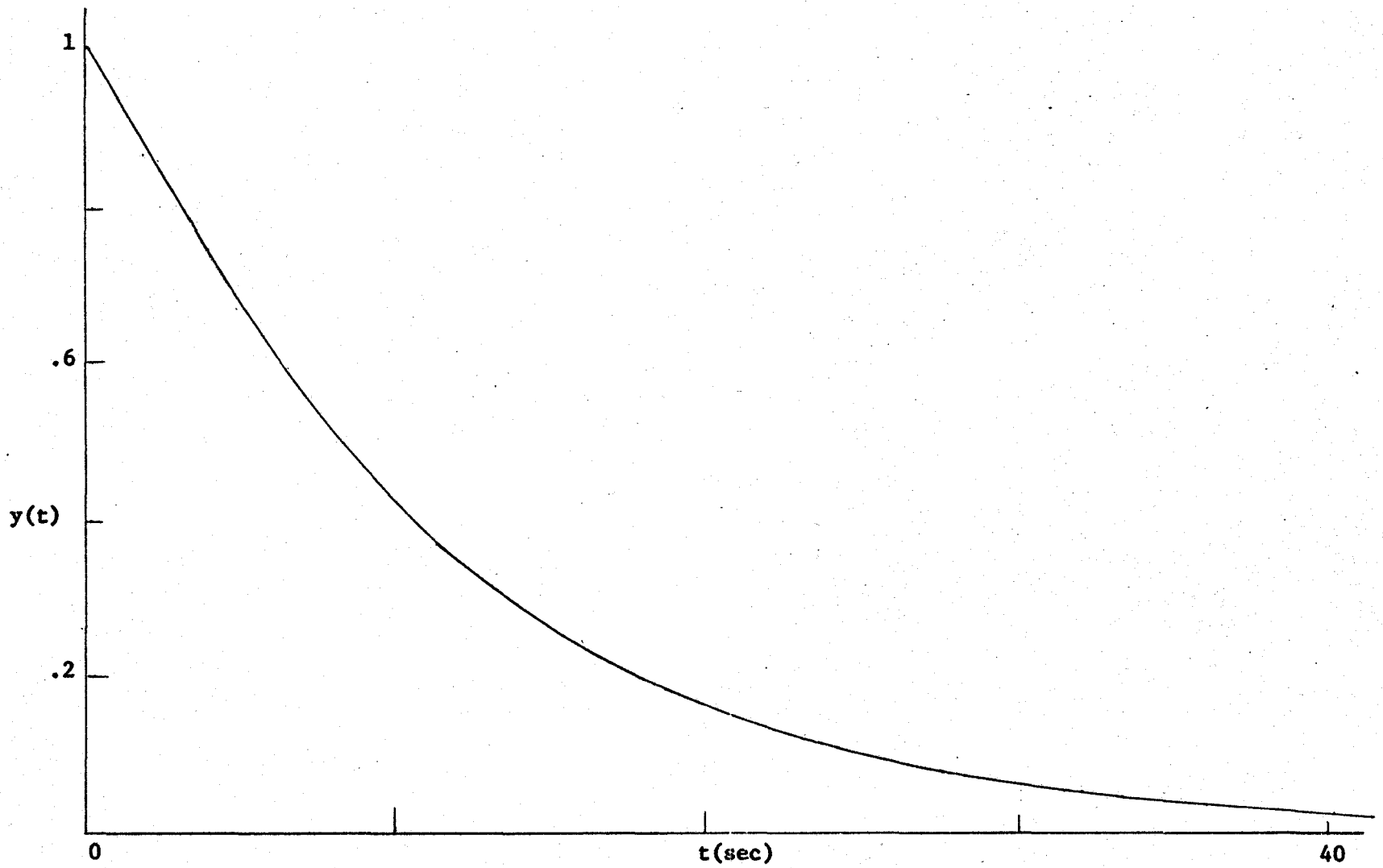


Figure 5.1 Optimal Response Of the System For First Cost Function

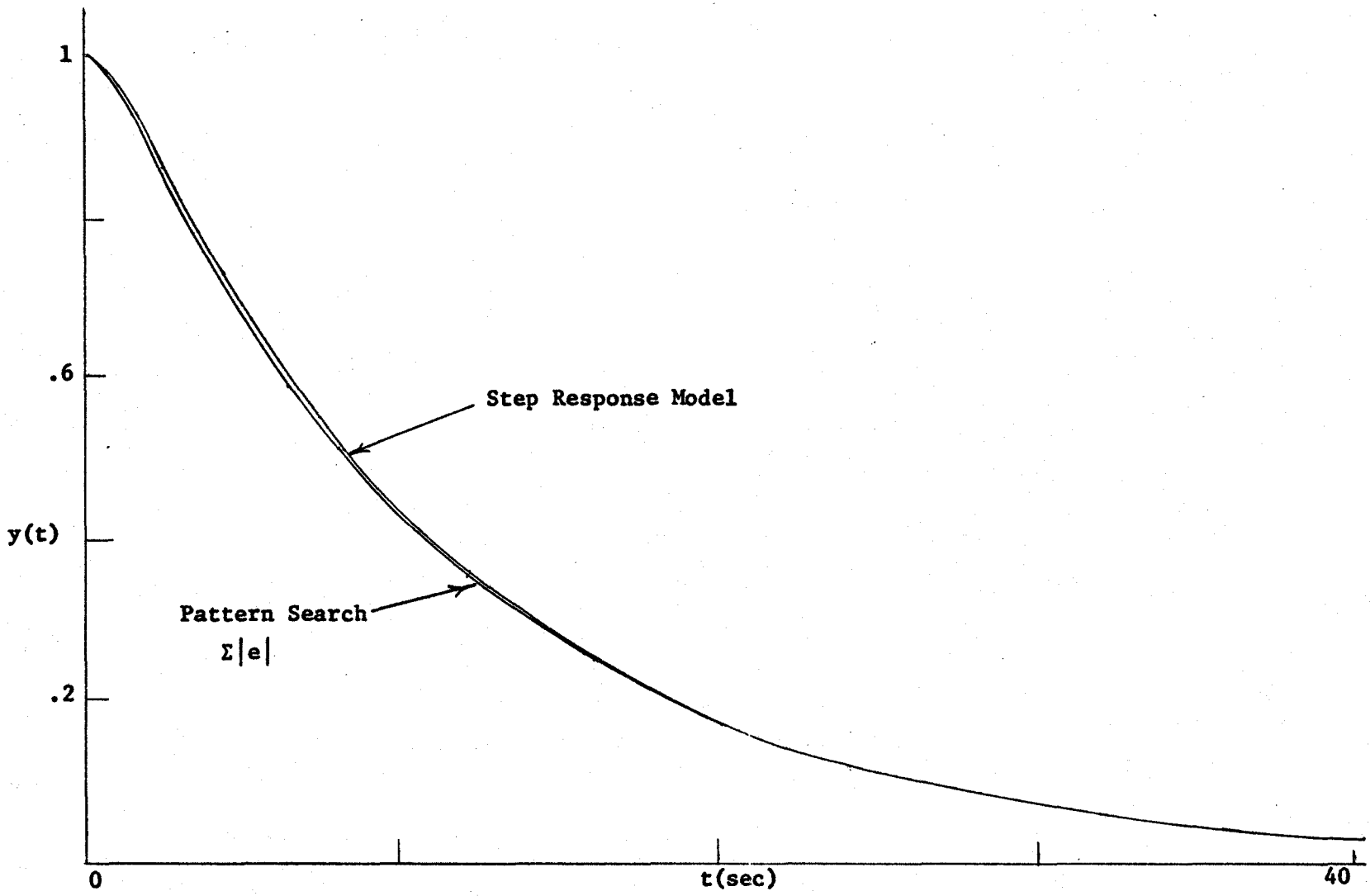


Figure 5.2 Sub-Optimal Response of the System for two Reduced Models

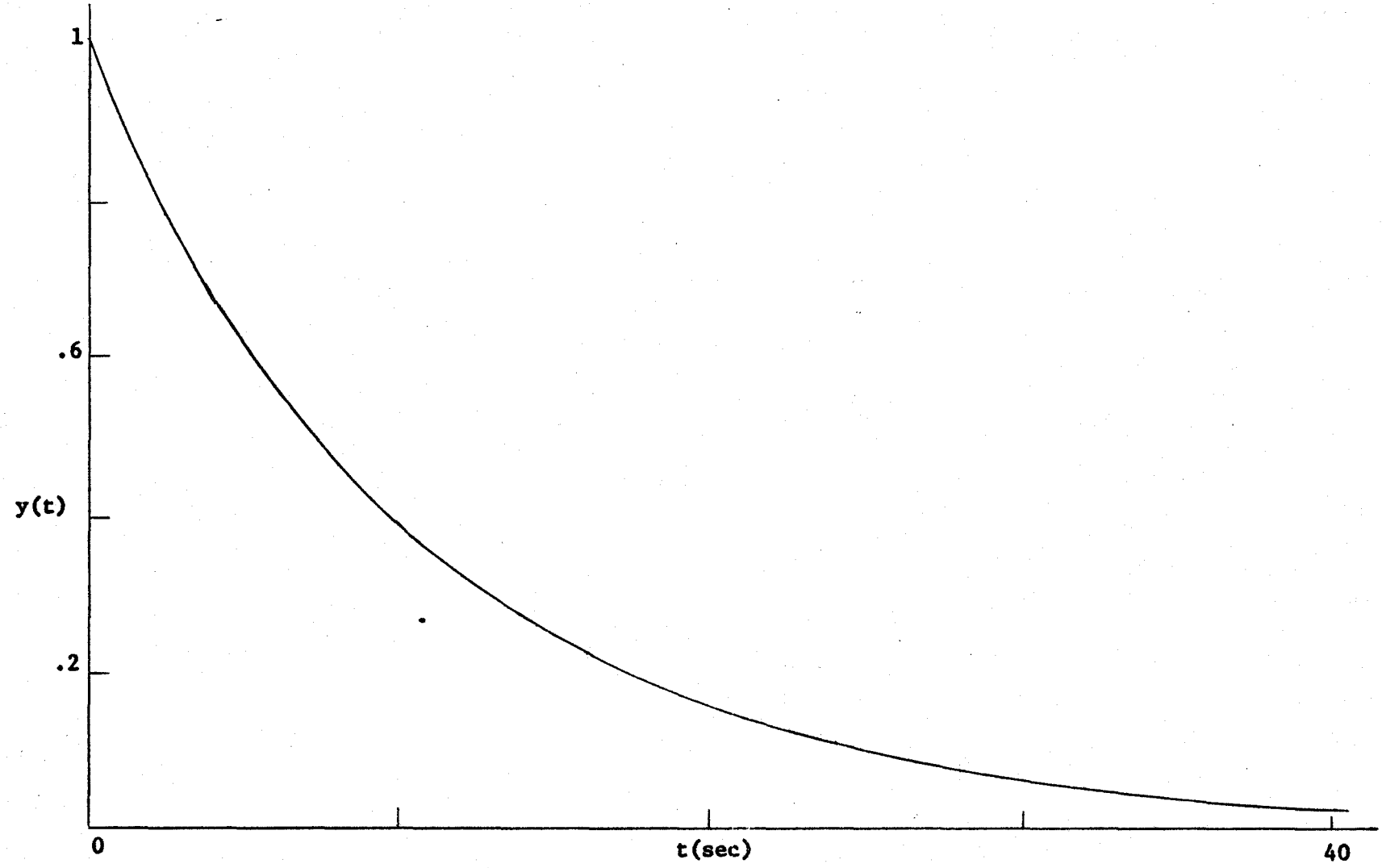


Figure 5.3 Optimal Response of the System for Second Cost Function

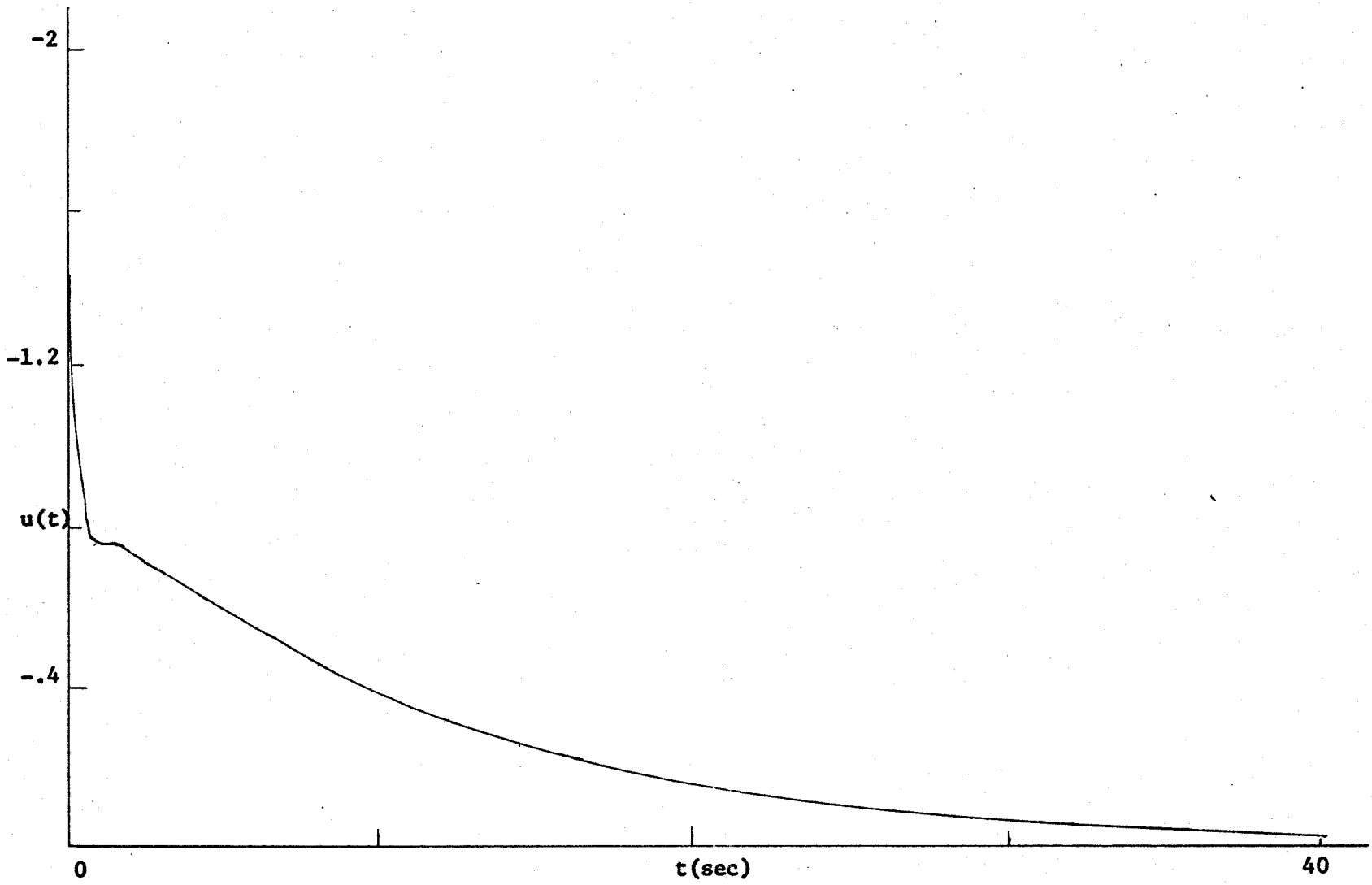


Figure 5.4 Optimal Control For the System for Second Cost Function

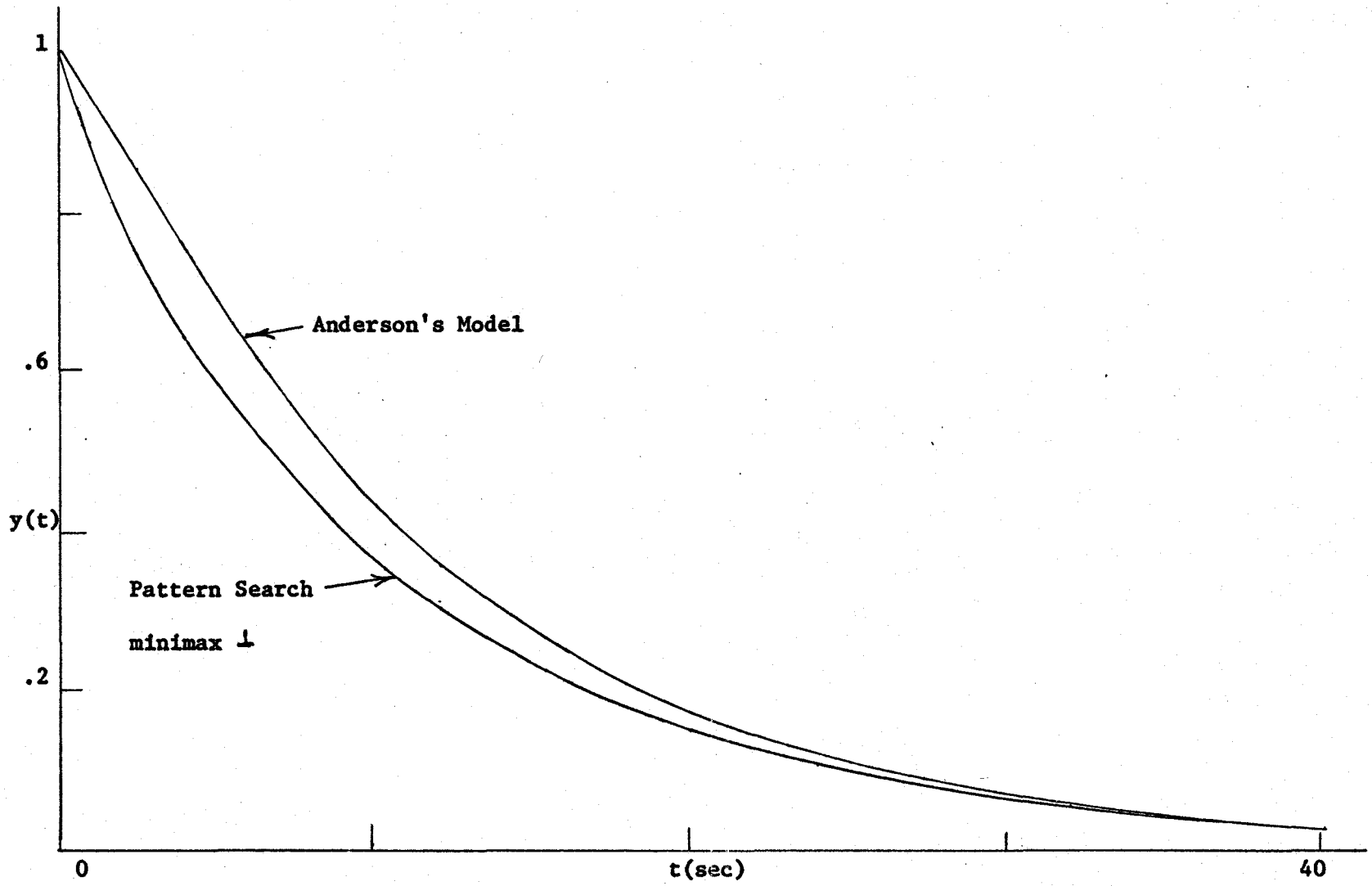


Figure 5.5 Sub-optimal Response of the System For Two Reduced Models For Second Cost Function

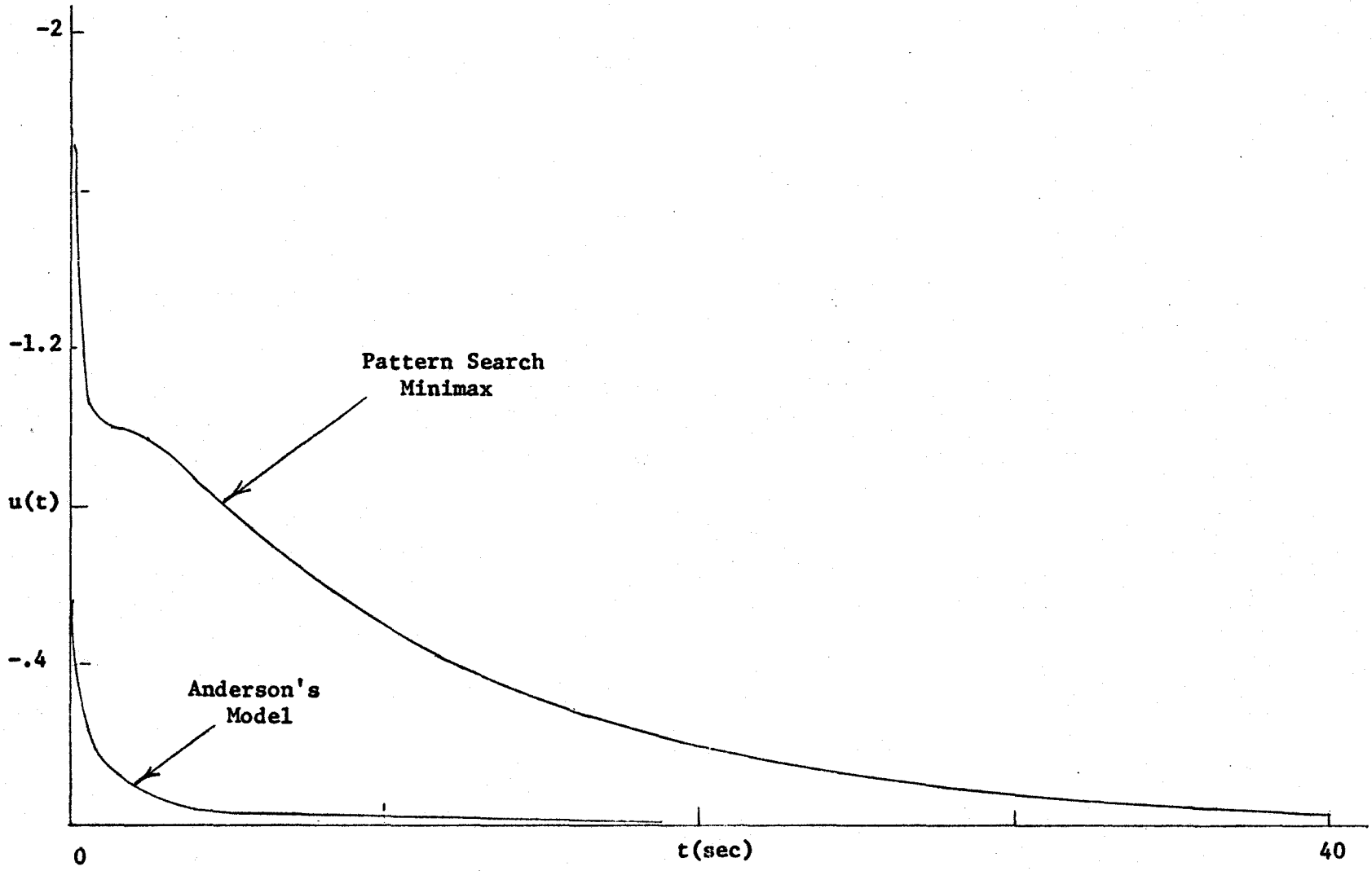


Figure 5.6 Sub-Optimal Control For The System For Two Reduced Models For Second Cost Function

Since ten models are studied it is felt that the two extremes suffice since all the other sub-optimal trajectories and controls lie between them.

This section has presented the results for each case of sub-optimal control separately. The next sections will consider all the results as a whole.

5.2 Comparison with Respect to Computational Effort Required.

From Tables 2 and 3, it can be seen that the closed-loop method of control approximates the optimal control for the system more closely than the open-loop method for every model used. The system costs are very close for the two methods of control, but this is attributable to the fact that the first cost function is very insensitive to feedback coefficient variation because of their small magnitude. The closed-loop method of sub-optimal control results in a closer approximation of the true optimum. In addition, in practical systems, the addition of noise and parameter variation make closed-loop control more desirable. For the solution of the matrix Riccati equation, those models which have no zero are more desirable since the feedback coefficients can be directly calculated using the formulae of Appendix C.2. However, those models which contain zeroes, approximate the optimal control more closely, even when they are in state variable rather than phase variable form, as can be seen from Table 4. Since in this representation the formulae can also be used,

computationally, a reduced model without a zero is no more desirable.

Considering models containing zeroes, if the cost function for the system is written in terms of the output and its derivatives, the model should be represented in phase variable form. For the test system considered, this resulted in most models having a negative feedback coefficient, because of one of the elements of the control vector is also negative. The Pattern Search (minimax$\perp$) which was fourth best for all other cases considered, was the only model having positive feedback coefficients. However, it approximated the optimal control most closely, for the second cost function, when used in phase variable form.

The pattern search method required more computer time than most other reduction methods, however, the resulting models are optimized with respect to a selected error criterion. They are also consistently the most suitable models for approximating the system optimal control. The increased computation required is therefore justified. The modal reduction methods are tedious and require the calculation of the eigenvalues, eigenvectors and modal matrix inverse. Even though the modes retained are those closest to the $j\omega$-axis in the s-plane, the resulting models are no more suitable than those derived using much simpler methods. As expected, the Optimal Projection Model was slightly better than the Davison Model.

To derive a first approximation of the optimal control, one of the simpler reduction methods such as the Iterative or Step Response Methods should be used.

Although the second cost function is more difficult to apply, there is a better separation of sub-optimal costs and responses as can be seen from Table 4 and Figures 5.5 and 5.6. This is due to the $.1u^2$ term which allows larger feedback coefficients and hence, a larger control input. Hence, variation in model parameters result in greater differences in the approximation of the optimal control. The suitability of different models is much more apparent than for the first cost function.

5.3 Comparison of Model Suitability.

The results summarized in Tables 2,3 and 4 are very consistent. The order of model suitability is basically unchanged, between the different cases of sub-optimal control. In addition, the most suitable models result in near-optimal control of the system. For both cost functions, the system sub-optimal cost approaches the true optimum even for the second cost function, where the model and system costs are not truly identical.

For the first cost function, the suitability of the different models may be more easily perceived if one compares the feedback coefficients of Table 3 with the first two optimal feedback coefficients for the system.

Those reduction methods which optimize the model with respect to some error criterion, result in models which are most suitable for sub-optimal control applications. The most desirable criterions are those which stress the steady-state part of the system response. The Pattern Search $\Sigma|e|$ criterion and Σe^2 with steady-state constraint criterion, both stress the steady-state part, the former more than the latter. In addition, the Chen & Shieh reduction method stresses the steady-state part of the system response, because of the expansion around $s = 0$.

The reduction methods which minimize the mean squared error, are not as suitable because this criterion inherently stresses the transient portion of the system response. As can be seen from Tables 2,3 and 4, the three models resulting from this criterion are equally suitable. Since the step response reduction method allows one to stress selected system response features, the resulting model may be made more suitable by stressing steady-state response.

The modal reduction methods result in models as suitable as the mean squared error models. A better model may have been derived if the pole closest to the origin had not been neglected.

In summary,the most suitable reduction methods are the Pattern Search Method with error criterion which stress steady-state and the Chen & Shieh method. The resulting sub-optimal controls and trajectories are very near optimum as can be seen from Figures 5.1 to 5.6. The next chapter discusses the conclusions to be drawn from this work.

CHAPTER 6

C O N C L U S I O N S

Ten different reduced models have been applied, to provide sub-optimal control of a test system. The models were used to provide both open-loop and feedback sub-optimal control for one cost function, and only feedback sub-optimal control for another. Since each model was derived using a different reduction method, the objective was to determine which reduced models were most suitable for providing an approximation of the optimal control of a system. Since the results obtained are quite consistent, some conclusions may be drawn from this work.

For all models used, the feedback method of sub-optimal control produced better results than the open-loop method. In addition the feedback method was much easier to apply. For this method, for both cost functions, the sub-optimal cost for the system was very near to the system optimum for some models used. This method of using the model optimal feedback coefficients as an approximation of the system optimal coefficients is useful and for some reduced models results in a sufficiently close approximation of the system optimal control for most practical purposes. Those models which are computationally easier to handle, that is, those models having no zero,

may be used to derive a first approximation of the optimal control, the true optimum being found by some other method.

In general, it is usually not possible to find an explicit representation of the closed-loop optimal control for a non-linear system. If a linear low-order model for the system is found, the closed-loop method of sub-optimal control may be used to approximate the optimal control for the system. If necessary, a better approximation may then be found using a search routine.

The desirability of using reduced models to approximate the optimal control of a complex system is also verified. Although the test system was only seventh order with one zero, if the model optimum was used instead of the system, the reduction in computer time required to solve the matrix Riccati equation was enormous. Even for those models which most closely approximated the system optimal response, the computer time was one fortieth of that required for the system solution. In addition, the system parameters must be known in order to derive the optimal feedback coefficients. Since some reduction methods require only the system response for a particular input, the system may be controlled near-optimally without any knowledge of the system parameters.

If only a first approximation of the optimal control is required, it would be more desirable to use a reduced model having no zero.

The model optimal feedback coefficients can be directly calculated using the formulae previously mentioned. For those models with a zero, the matrix Riccati equation must be integrated backward in time until a steady-state solution is reached. This fact seems to make reduced models having no zero, more suitable for on-line applications. However, even if the reduced model is represented in state variable form, rather than phase variable, the approximate optimal control is still better than that derived for those models without a zero. Since in this representation the formulae can also be used, no added value may be given to a model because it does not contain a zero.

The order of model suitability is not basically different for both cost functions and both methods of sub-optimal control. The reduction methods which stress the steady-state or settling time part of the system response result in reduced models which are most suitable for sub-optimal control. These methods include the pattern search method which minimizes the sum of the absolute errors between the system and model response, and the Chen & Shieh method. One weakness of the Chen and Shieh method is that the system transfer function must be known and only single input-output systems can be considered. To a lesser extent the pattern search method which minimizes the sum of the squares of the errors with steady-state constraint, also results in a

suitable model. As expected, the methods which minimize the mean squared error, all produce models which are equally good. Also, they are not as suitable as those mentioned above, since a mean squared error objective inherently stresses the transient part of the system response. The modal reduction methods, although they require a knowledge of the system eigenvalues and eigenvectors, and are computationally tedious, produce models which are no better than those derived using simpler reduction techniques.

From these results, it may be stated that those reduction methods which optimize the low-order model with respect to some objective error function, produce models which are most suitable for optimal control. If this objective function stresses the steady-state portion of the model and system responses, the order of suitability is increased. Of the reduction techniques considered in this work, the pattern search method results in models which are most suitable. Although computationally long, it requires no knowledge of the system parameters and the objective error function may be chosen to stress any part or aspect of the system response. For a first approximation, the step response method may be selected since it requires no knowledge of the system parameters, is computationally the easiest method to apply and stresses selected system response characteristics.

By choosing the appropriate reduction method, a second order reduced model can be derived which can provide a very good approximation

of the optimal control for the system. If higher order reduced models are derived, this approximate control may be made to approach the true optimum.

APPENDIX A

THE TEST SYSTEM

A.1 The Test System Transfer Function

For the sake of reality and validity, it was decided that an authentic system model containing a reasonable number of complex poles, fairly distributed in the s-plane, was required.

Reduction techniques invariably neglect (to some extent) the poles of the system which are located far from the origin since these poles represent small time constants and their effects are relatively short-lived. Some authors have taken examples in which there are two sets of poles, one near the $j\omega$-axis, and the other far away from it. This restriction, besides being unrealistic, favors the modal reduction methods which select certain modes to be retained and neglect the rest. To avoid biased results, then, a transfer function with poles distributed over the entire left-half plane was preferred.

The system chosen[14] was one of the designs studied for the current super-sonic transport aircraft. Figure A.1 shows the block diagram of the system with variable parameters K1, K2, τ,ζ ω_n. To obtain reasonable pole-zero locations, the following parameter values were selected by the authors[7] (consistent with design

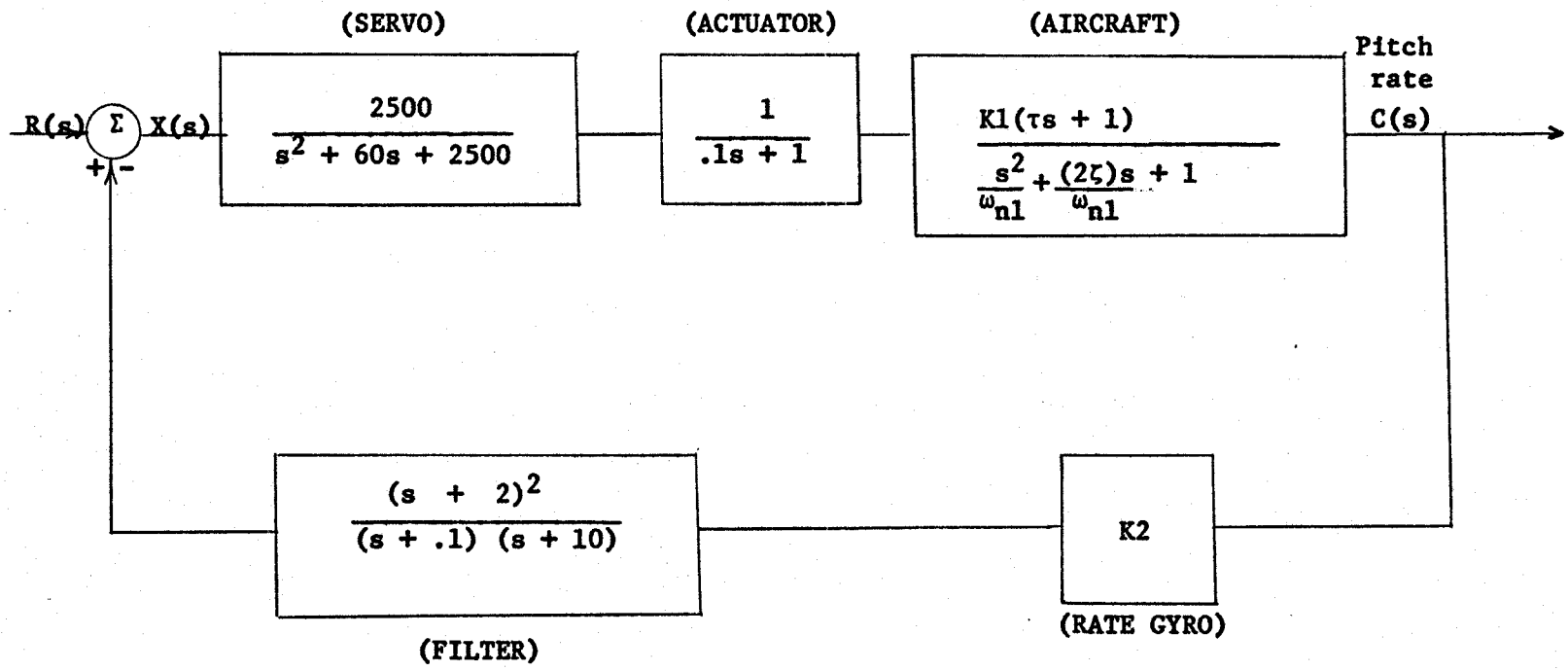


Figure A.1 Block Diagram for a Super Sonic Transport Airplane Design.

description and maintaining stability)

$K1 = 0.2$

$K2 = 1.0$

$\omega_{n1} = 2.5[r/s]$

$\tau_1 = 12$

$\zeta = 0.707$

With these parameter values, the transfer function of the test system becomes

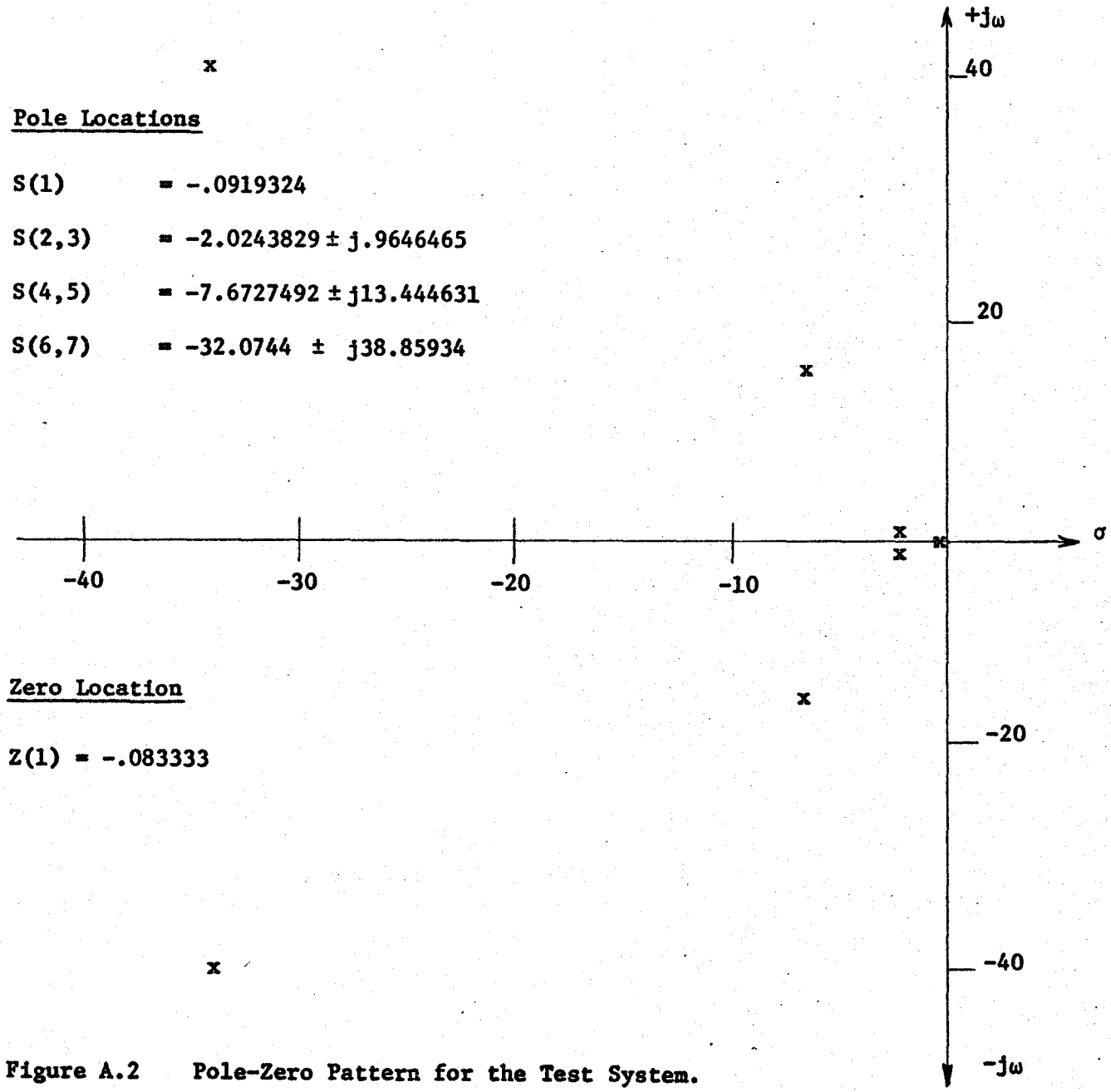
$$\frac{C(s)}{R(s)} = \frac{375000(s + 0.083\dot{3})}{s^7 + 83.64s^6 + 4097.40s^5 + 70341.9s^4}$$

$$+ \frac{}{853703s^3 + 2814271s^2 + 3310875s + 281250}$$

The characteristic equation of the system is given by F(s) = Denominator of Transfer Function.

Since the roots of the characteristic equation are the poles of the system and also the eigenvalues of the A matrix if the system is represented in state space form, these are only mentioned once in Figure A.2.

Pole-Zero Map for the Test System



Pole Locations

S(1) = -.0919324

S(2,3) = -2.0243829 ± j.9646465

S(4,5) = -7.6727492 ± j13.444631

S(6,7) = -32.0744 ± j38.85934

Zero Location

Z(1) = -.083333

Figure A.2 Pole-Zero Pattern for the Test System.

A.2 State Space System Representation

From the transfer function the state space equations are developed. These are necessary both for calculating the system time response and the calculation and implementation of optimal, or sub-optimal control.

The state space equations, written in the general form

$$\dot{\underline{x}} = A\,\underline{x} + E\,\underline{u}$$

$$y = C\,\underline{x}$$

are not desirable. As stated in Chapter 4, it is necessary for the state space representation to be in phase variable form. This means specifically that

$$x_1 = y \qquad \text{and} \qquad x_2 = \dot{y}$$

The state space equations for a single input-output system can then be simply written

$$\dot{\underline{x}} = A\,\underline{x} + B\,\underline{u}$$

$$y = x_1$$

For this representation, the effects of the system zeroes are compensated for by modifying the control vector. One of the techniques [12] of accomplishing this is to proceed as follows:

Letting $x_1 = y(t) + b_0\, u(t)$

$x_2 = \dot{y}(t) + b_0\, \dot{u}(t) + b_1\, u(t)$

etc.

and applying the restriction that no derivatives of u(t) must appear in the vector differential equation, the elements of the control vector $b_1, b_2, \ldots b_7$ can be solved for in terms of the transfer function parameters. The resulting vector differential equation, can be written

as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -a_1 & -a_2 & -a_3 & -a_4 & -a_5 & -a_6 & -a_7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ b_6 \\ b_7 \end{bmatrix}$$

where $a_1 = 281250$

$a_2 = 3310875$

$a_3 = 2814271$

$a_4 = 853703$

$a_5 = 70342$

$a_6 = 4097.4$

$a_7 = 83.64$

and $b_6 = 375000$

$b_7 = -31333751$

now $y = x_1$

$\dot{y} = x_2$

This representation has been used in all calculations involving the test system. In the above discussion $\underline{x}$, y, and u are functions of time, but for simplicity the (t) have been omitted.

APPENDIX B

SOLUTION OF THE STATE SPACE EQUATIONS USING A DIGITAL COMPUTER

B.1 Methods of Solution

The solution of the equations

$$\dot{\underline{x}}(t) = A\underline{x}(t) + Bu(t)$$

is

$$\underline{x}(t) = e^{A(t-t_0)} \cdot \underline{x}(t_0) + e^{At} \int_{t_0}^{t} e^{-A\tau} Bu(\tau)d\tau$$

This equation can be solved by direct integration and then substitution of the required values for t. However, this method cannot be used for high-order systems because of computational difficulty.

If t_0 is taken to be zero, which is not unduly restrictive, the solution becomes

$$\underline{x}(t) = e^{At}[\underline{x}(0) + \int_{0}^{t} e^{-A\tau} Bu(\tau)d\tau]$$

In order to use a digital computer to solve the state space equations, the solution must first be expressed in discrete form. The solution becomes easier if the driving function, or input, is held constant between sampling intervals, or can be suitably approximated by a series of step functions which can be considered constant between the

sampling intervals. This condition is met when the system is driven sub-optimally in an open-loop sense since the optimal control for the model is given as a series of values at the sampling instants.

The solution can then be given by

$$\underline{x}(k+1)T = e^{AT}\,\underline{x}(kT) + \int_0^T e^{-At}\underline{B}\underline{u}(kT)dt. \qquad \ldots.(B.1\text{-}1)$$

where T = sampling interval

and k = sampling instant

If e^{AT} and e^{-AT} are represented in infinite series form as

$$e^{AT} = (I+AT + \frac{(AT)^2}{2!} + \frac{(AT)^3}{3!} + ...) \qquad \ldots.(B.1\text{-}2)$$

and $$e^{-AT} = (I-AT+\frac{(AT)^2}{2!} - \frac{(AT)^3}{3!} + ...)$$

and equation B.1-1 integrated between the limits shown, the solution becomes

$$\underline{x}\,(k+1)T = e^{AT}\,\underline{x}(kT) + [I-\frac{AT}{2!} + \frac{(AT)^2}{3!} - \ldots.]\,Bu(kT)T \qquad \ldots(B.1\text{-}3)$$

where e^{AT} is given by equation B.1-2.

This form of solution (later referred to as the series expansion method), lends itself to machine computation. If the two infinite series are first computed using a suitable truncation criterion, equation B.1-3 can be solved iteratively by simply up-

dating the initial conditions and control input.

However, this form of solution is not suitable for feedback control where $u(kT) = -K^T\underline{x}(kT)$. For this form of control the responses may be best calculated using a numerical method of integrating the ordinary differential equations. Although round-off errors may become significant using these methods, these can be minimized by selecting a suitably small time interval. The numerical method selected is a fourth order Runge-Kutta method.[15] It obtains an approximate solution of the vector differential equation with given initial conditions.

To obtain the value of the cost function for the total time interval of interest, the value of the integrand of

$J = \int_0^\infty (x^TQx + u^TPu)dt$ is obtained for each sample interval.

A Simpson's rule integration technique is then employed to evaluate the integral of this equally spaced data. There is a subroutine resident in the CDC 6400 computer library which combines Simpson's rule and Newton's 3/8 rule to perform the necessary integration.

A general program is included in Appendix B.2 which calculates the response of a system defined by the equations

$$\dot{\underline{x}} = A\underline{x} + Bu$$

$$y = C\underline{x}$$

$$u = N[r(t)-K^T\underline{x}(t)]$$

The program also evaluates the cost function for the total time interval of control and plots the output, control and desired state variables using the line printer.

This program was used to compute all model and system responses for the cases of model and system optimal control and open-loop or closed-loop sub-optimal control of the system. The sub-optimal costs and responses of the system were also computed for the case of open-loop sub-optimal control of the system, using the series expansion method previously mentioned. There is good agreement between the results of both methods of solution if a sampling interval of .04 seconds is used. Since an 80 second control time is required to drive the system to zero, the system response is computed for 2000 samples.

A smaller sample time produced results which agreed closely with those produced by a .04 second interval (five figure accuracy). For all calculations therefore, it was decided that a .04 second time interval provided sufficient accuracy.

The program can be easily modified to provide a punched data deck of response or control values. However, these additions are not included in this listing.

APPENDIX B.2

COMPUTER PROGRAM FOR SOLUTION OF

STATE SPACE EQUATIONS USING A DIGITAL COMPUTER

```
C         GRAPHICAL TIME RESPONSE (GTRESP)
C       SUBPROGRAMS USED- CALCU, RUNGE,TRESP, YDOT
C     THIS PROGRAM CALCULATES THE RESPONSE , STATE VECTOR, AND
C     CONTROL TRAJECTORIES GIVEN SOME INITIAL CONDITION AND
C     EITHER OPEN-LOOP OR CLOSED-LOOP CONTROL
      INTEGER CHAR(15)
      COMMON IPLOT,IVAR(10)
      DIMENSION A(10,10),C(10),B(10),AK(10),X(10),NAME(5)
      DATA CHAR(1),CHAR(2),CHAR(3),CHAR(4),CHAR(5),
     8CHAR(6),CHAR(7),CHAR(8),CHAR(9),CHAR(10),CHAR(11),
     7CHAR(12),CHAR(13),CHAR(14),CHAR(15)/2H 1,2H 2,2H 3,2H 4,
     4 2H 5,2H 6,2H 7,2H 8,2H 9,2H10,2H E,2H O,2H Y,2H R,2H  /
C
C     INPUT AND OUTPUT FORMAT STATEMENT
C
    3 FORMAT(7F11.2)
 1000 FORMAT (1H0,10X,8HTZERO = ,F10.6,10X,5HTF = ,F10.6/
     811X,5HDT = ,F10.6,13X,7HFREQ = ,I5)
 1001 FORMAT (1H0,10X,13H THE A MATRIX   /)
 1002 FORMAT (6(E20.8))
 1003 FORMAT (1H0,10X,19H INITIAL CONDITIONS   /)
 1004 FORMAT (1H0,10X,13H THE B MATRIX  /)
 1005 FORMAT (1H0,10X,16H FEEDBACK COEFF.    /)
 1006 FORMAT (1H0,10X,8H GAIN = ,E20.8)
 1007 FORMAT (1H0,10X,13H THE C MATRIX  /)
 1008 FORMAT (8A2)
 1009 FORMAT (5X,25HPROBLEM IDENTIFICATION - ,5A4)
 1010 FORMAT(1H1,4X,23HGRAPHICAL TIME RESPONSE)
 1011 FORMAT(/5X,45(1H*))
 1012 FORMAT(8F10.7)
C
C
      IPROG = 0
   10 READ(5,1) (NAME(I),I=1,5),N
    1 FORMAT (5A4,I2)
      DO 60 I=1,8
   60 IVAR(I) = CHAR(15)
      PRINT 1010
      PRINT 1009,(NAME(I),I=1,5)
      PRINT 1011
C
C     THE A MATRIX IS READ IN AND OUTPUTTED
      PRINT 1001
      DO 2 I=1,N
```

```
      READ 3, (A(I,J),J=1,N)
      PRINT 1002, (A(I,J), J=1,N)
    2 CONTINUE
C
C     THE B VECTOR ELEMENTS FOR THE SYSTEM ARE GIVEN
C
      B(1) = 0.0
      B(2) = 0.0
      B(3) = 0.0
      B(4) = 0.0
      B(5) = 0.0
      B(6) = 375000.
      B(7) = 375000.*0.08333-375000.*83.64
      PRINT 1004

      PRINT 1002, (B(I),I=1,N)
C
C     THE C VECTOR ELEMENTS ARE READ IN AND OUTPUTTED
      READ 3, (C(I),I=1,N)
      PRINT 1007
      PRINT 1002, (C(I),I=1,N)
C
C     THE FEEDBACK COEFFICIENTS ARE READ IN AND WRITTEN
      READ 1012, (AK(I),I=1,N)
      PRINT 1005
      PRINT 1002, (AK(I),I=1,N)
C
      READ 3, GAIN
      PRINT 1006, GAIN
C
C     THE INITIAL CONDITIONS ARE READ IN AND WRITTEN
      READ 3, (X(I),I=1,N)
      PRINT 1003
      PRINT 1002, (X(I),I=1,N)
C
C     THE PROGRAM TIME PARAMETER ARE READ IN AND WRITTEN
      READ 3, TZERO,TF,DT,FREQ
      IFQ = FREQ
      PRINT 1000, TZERO,TF,DT,IFQ
      PRINT 1011
C
C     THIS PART OF THE PROGRAM DETERMINES THE TRAJECTORIES
C     THAT ARE TO BE GRAPHED AND PREPARED THE PLOT SUBROUTINE
C
C
      READ 1008, (IVAR(I),I=1,8)
      DO 40 I=1,8
      DO 30 J=1,15
      IF(IVAR(I)-CHAR(J)) 30,25,30
   25 IVAR(I) = J
```

```
      GO TO 40
   30 CONTINUE
   40 CONTINUE
      MIN = 1
      MAX = 8
      M = 8
  419 DO 42 I=MIN,MAX
      IF(IVAR(I).NE.15) GO TO 42
      M = MAX-1
      IF(I.GT.M) GO TO 42
      DO 43 J=I,M
   43 IVAR(J) = IVAR(J+1)
      GO TO 431
   42 CONTINUE
      GO TO 432
  431 MIN = I
      MAX = M
      GO TO 419
  432 IPLOT = M
      IF(IPLOT.LT.2) GO TO 50
      LIM = IPLOT-1

      DO 44 I=1,LIM
      MIN = I+1
      DO 44 J=MIN,IPLOT
      IF(IVAR(I)-IVAR(J)) 44,44,45
   45 IHOLD = IVAR(I)
      IVAR(I) = IVAR(J)
      IVAR(J) = IHOLD
   44 CONTINUE
C
C     THE INTEGRATION PROCEDURE IS INITIATED BY CALLING SUBROUTINE
C     TRESP
C
   50 CALL TRESP(A,X,B,AK,TZERO,TF,DT,IFQ,N,GAIN,C)
      IPROG = IPROG+1
      IF(IPROG-6)10,20,20
   20 STOP
      END
```

```
      SUBROUTINE CALCU(X,U,T,N,AK,GAIN,R)
C    THIS SUBROUTINE COMPUTES THE REFERENCE AND CONTROL INPUTS
      DIMENSION X(10), AK(10)
C     BEGINNING OF ROUTINE TO DEFINE R THE OPEN-LOOP CONTROL INPUT
 1001 R = 0.0
 1002 CONTINUE
C     END OF ROUTINE TO DEFINE R(T)
      U = R
      DO 1 I=1,N
    1 U = U-AK(I)*X(I)
      U = U*GAIN
      RETURN
      END
```

```
      SUBROUTINE RUNGE (N,FN,H,X,Y,L,I)
C    FOURTH ORDER RUNGE KUTTA INTEGRATION ROUTINE
C     THIS SUBROUTINE PERFORMS THE ACTUAL INTEGRATION AND IS CALLED
C     FOUR TIMES FOR EACH TIME INTERVAL
      DIMENSION Y(600),PHI(600),SAVEY(600),FN(8)
      I = I+1
      GO TO (1,2,3,4,5) ,I
    1 L = 1
      RETURN
    2 DO 600 J=1,N
      SAVEY(J) = Y(J)
      PHI(J) = FN(J)
  600 Y(J) = SAVEY(J)+.5*H*FN(J)
      X = X+.5*H
      L = 1
      RETURN
    3 DO 700 J=1,N
      PHI(J) = PHI(J)+2.*FN(J)
  700 Y(J) = SAVEY(J)+.5*H*FN(J)
      L = 1
      RETURN
    4 DO 800 J=1,N
      PHI(J) = PHI(J)+2.*FN(J)
  800 Y(J) = SAVEY(J) +H*FN(J)
      X = X+.5*H
      L = 1
```

```
      RETURN
    5 DO 900 J=1,N
  900 Y(J) = SAVEY(J)+(H/6.)*(PHI(J)+FN(J))
      L = 2
      I = 0
      RETURN
      END



      SUBROUTINE TRESP(A,Y,B,AK,X,XMAX,H,IFREQ,N,GAIN,C)
C     THIS SUBROUTINE IS THE MASTER SUBROUTINE WHICH CALLS THE OTHER THREE
C    SUBROUTINES IN THE INTEGRATION PROCEDURE
C     IT USES A FOURTH ORDER RUNGE-KUTTA ALGORITHM TO INTEGRATE THE
C     LINEAR SYSTEM
C        THIS SUBROUTINE COMPUTES AND PLOTS TIME RESPONSE
C     USING CALCU,RUNGE,YDOT AND Y8VSX
      INTEGER CHAR(15)
      COMMON IPLOT,IVAR(10)
      DIMENSION SKJ(101,9),C(10),SCOST(2001),SINT(2001)
      DIMENSION FN(10),Y(10),A(10,10),B(10),AK(10)
      DATA CHAR(1),CHAR(2),CHAR(3),CHAR(4),CHAR(5),
     8CHAR(6),CHAR(7),CHAR(8),CHAR(9),CHAR(10),CHAR(11),
     4CHAR(12),CHAR(13),CHAR(14),CHAR(15)/1H1,1H2,1H3,1H4,
     81H5,1H6,1H7,1H8,1H9,1HA,1HE,1HU,1HY,1HR,1H /
   24 FORMAT(2F10.0,2I10)
   25 FORMAT(8F10.0)
   28 FORMAT(//,8X,1HT,12X,4HY(T),10X,4HU(T),4X,
     7    7(5X,1HX,I1,4H(T) ,3X))
   29 FORMAT(10(E14.6))
 1000 FORMAT(/,5X,33HMAXIMUM NUMBER OF POINTS EXCEEDED /)
      PRINT 28, (J,J=1,2)
C
C     THIS SECTION CALCULATES THE COST FOR EACH TIME INTERVAL AND
C     STORES IT IN SCOST
C
      ICOST = 1
      DO 4000 I=1,2001
 4000 SCOST(I) = 0.0
      II = 0
```

```
      J = 0
      COST = 0.0
      KOUNT = IFREQ
  300 CALL CALCU(Y,U,X,N,AK,GAIN,R)
      SCOST(ICOST) = Y(1)**2+U**2
      ICOST = ICOST+1
C
 3000 KOUNT = KOUNT+1
      IF(KOUNT-IFREQ) 50,350,350
  350 KOUNT = 0
  450 P1 = 0.0
      DO 451 I=1,N
  451 P1 = P1+C(I)*Y(I)
      PRINT 29,  X,P1,U,(Y(M),M=1,2)
      IF(IPLOT.EQ.0) GO TO 21
      J = J+1
      IF(J.GT.101) GO TO 222
      SKJ(J,1) = X
      DO 40 I = 1,IPLOT
      MM = IVAR(I)
      IF(MM.EQ.0) GO TO 40
      IF(MM.GT.10) GO TO 35
      SKJ(J,I+1) = Y(MM)
      GO TO 40
   35 KNOW = MM-10
      GO TO (36,37,38,39), KNOW
   36 SKJ(J,I+1) = R-P1
      GO TO 40
   37 SKJ(J,I+1) = U
      GO TO 40
   38 SKJ(J,I+1) = P1
      GO TO 40
   39 SKJ(J,I+1) = R
   40 CONTINUE
   21 CONTINUE
   50 CALL RUNGE (N,FN,H,X,Y,L,II)
      IF(L-1) 100,200,100
  200 CALL CALCU(Y,U,X,N,AK,GAIN,R)
      CALL YDOT(A,Y,FN,B,U,N)
  550 GO TO 50
  222 PRINT 1000
      GO TO 400
  100 IF(X-XMAX) 300,300,400
  400 IF(IPLOT.EQ.0) GO TO 403
      PRINT 600
  600 FORMAT (1H1,50X,15HSYSTEM RESPONSE//)
      PRINT 601
  601 FORMAT (48X,8HVARIABLE,8X,6HSYMBOL//)
      DO 608 I = 1,IPLOT
      MM = IVAR(I)
      IF(MM.GT.10) GO TO 603
```

```
      PRINT 602,IVAR(I),CHAR(MM)
  602 FORMAT(51X,1HX,I2,13X,A1)
      GO TO 608
  603 KK = IVAR(I)-10
      GO TO (604,605,606,607), KK
  604 PRINT 610
      GO TO 608
  605 PRINT 611
      GO TO 608
  606 PRINT 612
      GO TO 608
  607 PRINT 613
  608 CONTINUE
  610 FORMAT (50X,5HERROR,12X,1HE)
  611 FORMAT(49X,7HCONTROL,11X,1HU)
  612 FORMAT(50X,6HOUTPUT,11X,1HY)
  613 FORMAT(50X,5HINPUT,12X,1HR)
      CALL Y8VSX(SKJ,J,IPLOT,10)
C
C     THE COST FUNCTION  IS WRITTEN FOR THE TIME INTERVAL
C
      CALL QSF(H,SCOST,SINT,2001)
      COST = SINT(2001)
  403 PRINT 8000, COST

 8000 FORMAT(1H0,10X,*JCOST = *,E14.6)
C
      RETURN
      END
```

```
      SUBROUTINE YDOT(A,Y,XDOT,B,U,N)
C     THIS SUBROUTINE IS USED TO COMPUTE THE DERIVATIVES OF X WITH RESPECT
C     TO TIME FOR EACH TIME INTERVAL
      DIMENSION Y(10),A(10,10),B(10),XDOT(10)
      DO 2 I = 1,N
      XDOT(I) = 0.
      DO 1 J=1,N
      XDOT(I) = XDOT(I)+A(I,J)*Y(J)
    1 CONTINUE
      XDOT(I) = XDOT(I)+B(I)*U
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE Y8VSX(A,N,M,NGRID)
C     THIS SUBROUTINE PLOTS UP TO 8 VARIABLES VERSUS TIME
C     THIS IS BOTH A X-T AND X-Y PLOT ROUTINE
C      IT AUTOMATICALLY SELECTS LIMITS AND DIMENSIONS FOR THE GRAPH
      COMMON IPLOT,IVAR(10)
      INTEGER CHAR(15)
      DIMENSION A(101,9),ABSCA(11),KAXIS(101),ORDIN(11),TEMPY(9)
      DATA CHAR(1),CHAR(2),CHAR(3),CHAR(4),CHAR(5),
     7CHAR(6),CHAR(7),CHAR(8),CHAR(9),CHAR(10),CHAR(11),
     5CHAR(12),CHAR(13),CHAR(14),CHAR(15)/1H1,1H2,1H3,1H4,1H5,
     21H6,1H7,1H8,1H9,1HA,1HE,1HU,1HY,1HR,1H /
      DATA ISTAR,II,IPER,IDASH,IBLANK/1H*,1HI,1H.,1H-,1H /
  100 FORMAT (////,9X,11(E10.2))
  101 FORMAT(E13.2,2X,101A1)
  102 FORMAT(15X,101A1)
      YMAX = A(1,2)
      YMIN = A(1,2)
      MP1 = M+1
      DO 4 J=2,MP1
      DO 4 I=1,N
      IF(YMAX-A(I,J)) 1,2,2
    1 YMAX = A(I,J)
    2 IF(YMIN-A(I,J)) 4,4,3
    3 YMIN = A(I,J)
    4 CONTINUE
      YSHFT = 0.0
      IF(YMIN) 5,6,6
    5 YSHFT = YMIN*100.0/(YMAX-YMIN)
    6 NM1 = N-1
      DO 8 I=1,NM1
      IP1 = I+1
      DO 8 K=IP1,N
      IF(A(K,1)-A(I,1)) 7,8,8
    7 DO 88 J=1,MP1
      ATEMP = A(I,J)
      A(I,J) = A(K,J)
      A(K,J) = ATEMP
   88 CONTINUE
    8 CONTINUE

      XMIN = A(1,1)
      XMAX = A(N,1)
      ABSCA(1) = XMIN
      ABSCA(11) = XMAX
      ORDIN(1) = YMIN
      ORDIN(11) = YMAX
      DO 9 I=2,10
      Z = I-1
      ABSCA(I) =(XMAX-XMIN)*Z/10.0+XMIN
    9 ORDIN(I) = (YMAX-YMIN)*Z/10.0+YMIN
      PRINT 100,(ORDIN(J),J=1,11)
```

```
      STEPX = (XMAX-XMIN)/100.0
      KDELX = 1
      KLINE = 1
      LINE = 1
      DO 26 IND=1,N
      IF(NGRID.EQ.0) GO TO 200
      KSTEP = LINE
      GO TO 201
  200 KSTEP = (A(IND,1)-XMIN)/STEPX+1.5
  201 DO 10 J=2,MP1
   10 TEMPY(J) = A(IND,J)*100.0/(YMAX-YMIN)-YSHFT
   11 IF(KLINE-LINE) 12,12,18
   12 DO 13 I=2,100
   13 KAXIS(I) = IDASH
      DO 14 I=1,101,10
   14 KAXIS(I) = ISTAR
      IF(KSTEP-LINE) 15,15,17
   15 DO 16 I=2,MP1
      K = TEMPY(I)+1.5
      MM = IVAR(I-1)
   16 KAXIS(K) = CHAR(MM)
   17 PRINT 101,ABSCA(KDELX),(KAXIS(J),J=1,101)
      IF(NGRID.EQ.0) GO TO 202
      KLINE = KLINE+NGRID
      ABSCA(KDELX) = A(KLINE,1)
      GO TO 24
  202 KLINE = KLINE+10
      KDELX = KDELX+1
      GO TO 24
   18 DO 19 I=2,100
   19 KAXIS(I) = IBLANK
      DO 20 I=1,101,10
   20 KAXIS(I) = IPER
      IF(KSTEP-LINE) 21,21,23
   21 DO 22 I=2,MP1
      K = TEMPY(I)+1.5
      MM = IVAR(I-1)
   22 KAXIS(K) = CHAR(MM)
   23 PRINT 102,(KAXIS(J),J=1,101)
   24 LINE = LINE+1
      IF(LINE-102) 25,25,27
   25 IF(KSTEP-LINE) 26,11,11
   26 CONTINUE
   27 RETURN
      END
```

APPENDIX C

SOLUTION OF THE MATRIX RICCATI EQUATION

C.1 Solution of the Matrix Riccati Equation for a Second Order Model having no zero.

Solution of the matrix Riccati equation requires that the reduced model be represented in state space form. To satisfy the requirements imposed in Chapter 4, the state variables must be in phase variable form.

If the model has a transfer function given by

$$G(s) = \frac{a}{s^2 + cs + b}$$

and

$$x_1 = y$$

$$x_2 = \dot{x}_1 = \dot{y}$$

the vector differential equation becomes

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -b & -c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ a \end{bmatrix} u$$

$$y = x_1$$

Although x, y, and u are functions of time the (t) have been omitted for simplicity.

If the cost function is given by equation 3.2-2,

Q is given by

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12} & Q_{22} \end{bmatrix}$$

and $$R = \begin{bmatrix} R_{11} & R_{12} \\ R_{12} & R_{22} \end{bmatrix}$$

the matrix Riccati equation can be solved very simply.

It becomes only necessary to solve the degenerate Riccati equation given by

$$A^T R + R A - R B P^{-1} B^T R + Q = 0$$

The resulting algebraic equations are

$$-2bR_{12} - P^{-1}a^2 R_{12}^2 + Q_{11} = 0$$

$$R_{11} - cR_{12} - bR_{22} - P^{-1}a^2 R_{12} R_{22} + Q_{12} = 0$$

$$2(R_{12} - cR_{22}) - P^{-1}a^2 R_{22}^2 + Q_{22} = 0$$

Since only one of the above equations is non-linear, and one equation has only one element, the

R matrix elements can be calculated directly

$$R_{12} = \frac{-b + \sqrt{b^2 + P^{-1}a^2 Q_{11}}}{P^{-1}a^2}$$

$$R_{22} = \frac{-c + \sqrt{c^2 + p^{-1}a^2(2R_{12} + Q_{22})}}{p^{-1}a^2}$$

$$R_{11} = cR_{12} + bR_{22} + p^{-1}a^2R_{12}R_{22} - Q_{12}$$

The feedback coefficients can be calculated from

$$K_r^T = p^{-1}B_r^TR$$

This procedure is computationally very simple and allows one to represent the optimal feedback coefficients directly in terms of the model and cost function parameters. It may therefore be useful in on-line adaptive control applications, and the optimization of a reduced model with respect to a system cost function.

C.2 SOLUTION OF THE MATRIX RICCATI EQUATION FOR THE SYSTEM AND A SECOND ORDER MODEL CONTAINING A ZERO.

As stated in Appendix C.1, the reduced model must first be represented in state space in a phase variable form. If the model transfer function is given by

$$G(s) = \frac{a\,s + b}{s^2 + ds + c}$$

and $x_1 = y(t) + b_0u(t)$

$x_2 = \dot{y}(t) + b_0\dot{u}(t) + b_1u(t)$

By imposing the condition that the coefficient of any derivative of u(t) must equal zero in the vector differential equation, the phase variable vector differential euqation becomes

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -c & -d \end{bmatrix} + \begin{bmatrix} a \\ b- ad \end{bmatrix} u$$

$$y = x_1$$

Again for $\underline{x}$, y and u the (t) have been omitted.

The degenerate matrix Riccati equation results in equations which cannot be easily solved. It is therefore necessary to solve the matrix Riccati equation given by

$$\dot{R} = RBP^{-1}B^TR - RA - A^TR - Q \qquad \ldots.(C.2\text{-}1)$$

This can be done easily on a digital computer since the above equation represents a set of first-order ordinary differential equations. The terminal conditbn is known and is

$$R(t_f) = 0$$

The matrix Riccati equation can then be integrated backwards in time from the known terminal condition. Since the control interval for the cost functions considered in this work is infinite, the R matrix is constant and the integration can be continued until the solution con-

verges to a steady-state value, or a desired degree of accuracy. Only the steady-state values are required and hence the integration step-size may be as large as desired, without introducing any errors in the final solution. This technique is also used to solve equation C.2-1 for the test system.

A program is included in Appendix C.2 which solves the matrix Riccati equationfor the seventh order test system. This program uses a subroutine which is resident in the CDC 6400 computer library, which uses a fourth-order Runge-Kutta method to perform the actual integration. Although the Runge-Kutta method is inefficient compared to predictor-corrector methods, it is stable and self-starting, and the integration step-size may be easily altered at any time during the calculations.

For the seventh-order system, a step size of .01 seconds has to be used to keep the R and $\dot{R}$ values within the computer maximum number bounds. Because of the magnitude of the test system parameters, choosing a larger step size causes the matrix values to exceed these limits.

In the program, Subroutine FCT contains the 28 differential equations, rather than the general matrix notation of the right hand side of equation C.2-1. Since only one test system is being considered, it is felt that this representation is more desirable,

as it minimizes the already long computer time required for the program. This program can be used for any order model just by changing the dimension variable.

The feedback coefficients can be calculated once the R matrix is known, using the equation

$$K^T = P^{-1}B^TR$$

Although this method is computationally longer than the formulae solution given in Appendix C.1, it is more general and may be used for any order system and cost function. Systems with multiple inputs and outputs may also be handled with equal facility.

APPENDIX C.2

COMPUTER PROGRAM FOR SOLUTION OF

THE MATRIX RICCATI EQUATION FOR THE TEST SYSTEM

```
C     THIS PROGRAM IS USED TO COMPUTE THE R MATRIX FOR THE SEVENTH ORDER SYSTEM
C     IT CALLS THE SUB ROUTINE RKGS WHICH IS RESIDENT IN THE LIBRARY
C     THE PRMT ELEMENTS DEFINE THE PARAMETERS OF INTEGRATION
C     THE Y ELEMENTS GIVE THE INITIAL VALUES FOR THE VARIABLES
      DIMENSION Y(28),PRMT(5),AUX(8,28),DERY(28)
      EXTERNAL FCT,OUTP
      PRINT 1
    1 FORMAT(5X,*SOLUTION OF MATRIX RICCATI EQN*/)
      PRINT 2
    2 FORMAT(1H0,10X,*MATRIX COEFFICIENTS*)
      NDIM = 28
      PRMT(1) = 80.0
      PRMT(2) = 0.0
      PRMT(3) = -.01
      PRMT(4) = 100.
      DO 10 I=1,28
      DERY(I) = 1.0/28.0
   10 Y(I) = 0.0
      CALL RKGS(PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)
      PRINT 3, IHLF
    3 FORMAT(1H0,10X,*ERROR MESSAGE  *,I3)
      STOP
      END


      SUBROUTINE OUTP(T,R,RDOT,IHLF,NDIM,PRMT)
C     THIS SUBROUTINE CONTROLS THE OUTPUT OF THE PROGRAM
C     IT SELECTS THOSE VALUES OF R AND R. WHICH ONE WANTS TO RETAIN
      DIMENSION R(28),RDOT(28),PRMT(5)
      IF(T-80.0) 21,20,20
   20 TCOUN = 80.0
      GO TO 6
   21 IF((TCOUN-T)-.20) 22,7,7
   22 CONTINUE
      RETURN
    7 TCOUN = T
    6 PRINT 1, T,(R(I),I=1,28)
    1 FORMAT(1X,F6.3,3X,10(E11.4,1X)/10X,10(E11.4,1X)/10X,10(E11.4,1X))
      PRINT 1,IHLF,(RDOT(I),I=1,28)
      RETURN
      END
```

```
      SUBROUTINE FCT(X,R,RDOT)
C     THIS SUBROUTINE CALCULATES THE DERIVATIVE OF R WITH RESPECT
C     TO TIME FOR EACH TIME INCREMENT
      DIMENSION R(28),RDOT(28)
C
C     THE FOLLOWING ARE THE PARAMETERS OF THE SEVENTH ORDER SYSTEM
C
      A = 375000.0
      B = 375000.*0.08333
      D1 = 281250.0
      D2 = 3310875.0
      D3 = 2814271.0
      D4 = 853703.0
      D5 = 70342.0
      D6 = 4097.0
      D7 = 83.64
      T = B-A*D7
C
      UA = SQRT(10.0)*(A*R(6)+T*R(7))
      UB = SQRT(10.)*(A*R(12)+T*R(13))
      UC = SQRT(10.)*(A*R(17)+T*R(18))
      UD = SQRT(10.)*(A*R(21)+T*R(22))
      UE = SQRT(10.)*(A*R(24)+T*R(25))
      UF = SQRT(10.)*(A*R(26)+T*R(27))
      UG = SQRT(10.)*(A*R(27)+T*R(28))
C
C     THE FOLLOWING ARE THE 28 NONLINEAR EQUATIONS DEFINING THE
C     THE DERIVATIVE OF R WITH RESPECT TO TIME
C
      RDOT(1) = UA**2+2.0*D1*R(7)-1.0
      RDOT(2) = UA*UB+D1*R(13)-(R(1)-D2*R(7))
      RDOT(3) = UA*UC+D1*R(18)-(R(2)-D3*R(7))
      RDOT(4)=UA*UD+D1*R(22)-(R(3)-D4*R(7))
      RDOT(5)=UA*UE+D1*R(25)-(R(4)-D5*R(7))
      RDOT(6)=UA*UF+D1*R(27)-(R(5)-D6*R(7))
      RDOT(7)=UA*UG+D1*R(28)-(R(6)-D7*R(7))
      RDOT(8)=UB**2-(R(2)-D2*R(13))-(R(2)-D2*R(13))-10.
      RDOT(9)=UB*UC-(R(8)-D3*R(13))-(R(3)-D2*R(18))
      RDOT(10)=UB*UD-(R(9)-D4*R(13))-(R(4)-D2*R(22))
      RDOT(11)=UB*UE-(R(10)-D5*R(13))-(R(5)-D2*R(25))
      RDOT(12)=UB*UF-(R(11)-D6*R(13))-(R(6)-D2*R(27))
      RDOT(13)=UB*UG-(R(12)-D7*R(13))-(R(7)-D2*R(28))
      RDOT(14)=UC**2-2.0*(R(9)-D3*R(18))
      RDOT(15)=UC*UD-(R(14)-D4*R(18))-(R(10)-D3*R(22))
      RDOT(16)=UC*UE-(R(15)-D5*R(18))-(R(11)-D3*R(25))
      RDOT(17)=UC*UF-(R(16)-D6*R(18))-(R(12)-D3*R(27))
      RDOT(18)=UC*UG-(R(17)-D7*R(18))-(R(13)-D3*R(28))
      RDOT(19)=UD**2-2.0*(R(15)-D4*R(22))
      RDOT(20)=UD*UE-(R(19)-D5*R(22))-(R(16)-D4*R(25))
      RDOT(21)=UD*UF-(R(20)-D6*R(22))-(R(17)-D4*R(27))
      RDOT(22)=UD*UG-(R(21)-D7*R(22))-(R(18)-D4*R(28))
```

```
RDOT(23)=UE**2-2.0*(R(20)-D5*R(25))
RDOT(24)=UE*UF-(R(23)-D6*R(25))-(R(21)-D5*R(27))
RDOT(25)=UE*UG-(R(24)-D7*R(25))-(R(22)-D5*R(28))
RDOT(26)=UF**2-2.0*(R(24)-D6*R(27))
RDOT(27)=UF*UG-(R(26)-D7*R(27))-(R(25)-D6*R(28))
RDOT(28)=UG**2-2.0*(R(27)-D7*R(28))
RETURN
END
```

REFERENCES

1. E. J. DAVISON: " A Method for Simplifying Linear Dynamic Systems ", IEEEE Trans. Automatic Control, vol. AC-11, 1966, pp. 93 - 101.

2. E. J. DAVISON: "A New Method for Simplifying Large Linear Dynamic Systems ", IEEE Trans. Automatic Control, Vol.AC -13, 1968, pp. 214-215.

3. J. H. ANDERSON: "Geometrical Approach to the Reduction of Dynamical Systems", Proc. IEE Vol. 114, 1967, pp.1014-1018.

4. D. MITRA: " On the Reduction of Complexity of Linear Dynamic Models", Technical Report AEEW - R 520, United Kingdom Atomic Energy Authority, 1967.

5. N. K. SINHA & W. PILLE : " A New Method for Reduction of Dynamic Systems", International Journal of Control (in press).

6. N. K. SINHA & G. T. BEREZNAI : "Optimum Approximation of High-Order Systems by Low-Order Models " , Int. Journal of Control, (accepted for publication).

7. D. M. FELLOWS, N.K. SINHA, & J.C. WISMATH : "Reduction of the Order of Dynamic Models", National Conf. on Automatic Control, Waterloo, Canada 1970.

8. C. F. SHEN & L. S. SHIEH : " A Novel Approach to Linear Model Simplification ", Proc. Joint Automatic Control Conference, 1968, pp. 454-461.

9. S. C. CHUANG : "Application of Continued -Fraction Method for Modelling Transfer Functions to Give More Accurate Initial Transient Response ", Electronic Letters, 1970, pp. 861-863.

10. D. MITRA : "Analytical Results on the Use of Reduced Models in the Control of Linear Dynamical Systems", Proc. IEE, Vol. 116, 1969, pp. 1439 - 1443.

11. D. MITRA : " W matrix and the Geometry of Model Equivalence and Reduction" , Proc. IEE, Vol. 116, 1969, pp. 1101-1106.

12. SCHULTZ & MELSA : "State Functions and Linear Control Systems", McGraw-Hill, 1967.

13. Stephen J. CITRON : "Elements of Optimal Control", Holt, Rinehart, and Winston Inc., 1969.

14. Richard C. DORF : " Modern Control Systems", Addison Wesley Inc., 1967.

15. R. W. HAMMING : "Numerical Methods for Scientists and Engineers", McGraw-Hill, 1962.

BIBLIOGRAPHY

1. SCHULTZ and MELSA: "State Functions and Linear Control Systems", McGraw-Hill Inc., 1967

2. R. W. HAMMING : "Numerical Methods for Scientists and Engineers", McGraw-Hill Inc., 1962

3. Stephen J. CITRON : "Elements of Optimal Control", Holt, Rinehart and Winston Inc., 1969

4. Andrew P. SAGE : "Optimum Systems Control", Prentice-Hall Inc., 1968

5. James L. MELSA : "Computer Programs for Computational Assistance", McGraw-Hill Inc., 1970