(NASA-CR-175831) ALTERNATIVES FOR JET ENGINE CONTROL Technical Progress Report, 1 Nov. 1983 - 31 Oct. 1984 (Notre Dame Univ.) 266 p HC A12/MF A01 CSCL 21E

N85-26713

Unclas
G3/07 21362

Department of

ELECTRICAL ENGINEERING

UNIVERSITY OF NOTRE DAME, NOTRE DAME, INDIANA

Twelve Month Technical Progress Report

to the

National Aeronautics and Space Administration

on

NASA Grant NSG-3048*

ALTERNATIVES FOR JET ENGINE CONTROL

November 1, 1983 - October 31, 1984

*Under the Direction of

Dr. Michael K. Sain
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana
USA 46556

ABSTRACT

This report documents the technical progress of researches under National Aeronautics and Space Administration Grant NSG-3048, entitled "Alternatives for Jet Engine Control", during the twelve-month period from November 1, 1983 to October 31, 1984. NASA Technical Officer for the work was Dr. Bruce Lehtinen, at Lewis Research Center. Dr. Michael K. Sain was director of the investigation at the University of Notre Dame.

In the period since the last report, Mr. Joseph A. O'Sullivan has completed a numerical study employing feedback tensors for optimal control of nonlinear systems. It is believed that these studies are the first of their kind. Results demonstrate improvement in state regulation, with a decrease in control power. A detailed treatment follows; it is based upon an extended version of Mr. O'Sullivan's M.S. Thesis.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

This paper deals with the nonlinear optimal control problem. For a system given by

$$\dot{x}(t) = f(x,u,t),\ t \in [t_0,t_1],\ x \in R^n,\ u \in R^m$$

a control, $u(t) = U(x(t),t)$, which is a function of the state vector, is sought to minimize the performance index

$$J(x_0,u,t_0) = M(x(t_1)) + \int_{t_0}^{t_1} L(x,u,t)dt,$$

where $M(x(t_1))$ and $L(x,u,t)$ are positive convex functionals [1]. All partial derivatives of $f(x,u,t)$ in some neighborhood of the origin of $R^n \times R^m$ with respect to x and u are assumed to exist so that

$$\begin{aligned} f(x,u,t) &= A(t) \circledcirc x + B(t) \circledcirc u + A_{0,[2]}(t) \overset{2}{\circledcirc} x^{[2]} + A_{[2],0}(t) \overset{2}{\circledcirc} u^{[2]} \\ &\quad + A_{1,1}(t) \overset{2}{\circledcirc} u \otimes x + \ldots \\ &= \sum_{p+q\geq 1} A_{[p],[q]}(t) \overset{p+q}{\circledcirc} u^{[p]} \otimes x^{[q]} \end{aligned}$$

where $A_{[p],[q]}(t)$ is a tensor which is continuous and bounded, $\overset{p+q}{\circledcirc}$ is the contraction operator described by Buric [1], and $u^{[p]} \otimes x^{[q]}$ stands for the tensor product of u with itself p times and x with itself q times. The sum is assumed to have either a finite number of terms or to be a convergent power series in some open neighborhood of $(x,u) = (0,0)$. The pair $(A,B) = (A_{0,1}, A_{1,0})$ is completely controllable. Similarly,

$$M(x) = \sum_k M_{[k]} \overset{k}{\circledcirc} x_f^{[k]},\ \text{where } x(t_1) = x_f,$$

$$L(x,u,t) = \sum_{j+k\geq 2} Q_{[j],[k]}(t) \overset{j+k}{\circledcirc} u^{[j]} \otimes x^{[k]},$$

where $Q_{[j],[k]}(t)$ are tensors which are piecewise continuous bounded functions of $t \in [t_0,t_1]$. The set of acceptable controls is assumed to be $\Omega = \{u(t) :$

$u(t) = \sum_j \overset{1}{K}_{[j]}(t) \overset{j}{\odot} x^{[j]}$, where the $\overset{1}{K}_{[j]}(t)$ are assumed to be bounded piecewise continuous functions of $t \in [t_0, t_1]$. Buric shows that under these assumptions the control tensor coefficients $\overset{1}{K}_{[j]}(t)$ are unique and are calculated recursively using $\overset{1}{K}_{1}(t), \ldots, \overset{1}{K}_{[j-1]}(t)$ and the optimal performance function $V(x,t)$ which is the solution to the Hamilton-Jacobi-Bellman equation for this problem. Buric's work is an extension of Lukes' [2] who demonstrated the existence and uniqueness of polynomial feedback in the case where

1) the $A_{[p],[q]}$ and the $Q_{[i],[j]}$ are not time-varying
2) $M(x(t_1)) = 0$
3) $t_1 = \infty$.

Lukes showed that the optimal value function, $V(x)$, is a power series

$$V(x) = \sum_{k=2} V_k(x), \text{ where } V_k(x) \text{ is a polynomial homogeneous of degree } k,$$

and that $V_k(x)$ can be computed as a function of $V_1(x), \ldots, V_{k-1}(x)$, and $\overset{1}{K}_{1}(x), \ldots, \overset{1}{K}_{k-2}(x)$. Then $\overset{1}{K}_{k-1}(x)$ is calculated as a function of $V_1(x), \ldots,$ $V_k(x)$ and $\overset{1}{K}_{1}(x), \ldots, \overset{1}{K}_{k-2}(x)$. Thus, he obtains the algorithm of calculating the $V_k(x)$ and $\overset{1}{K}_{k}(x)$ in the order: $V_2(x)$, $\overset{1}{K}_{1}(x)$, $V_3(x)$, $\overset{1}{K}_{2}(x)$, $V_4(x)$, $\overset{1}{K}_{3}(x), \ldots$ His results, however, are not in a closed form. It is difficult to see in his presentation the exact method for calculating these terms. Buric introduces the time-varying coefficients and the tensor algebra to the problem. By using tensors he arrives at a precise method of calculation for each of the performance function tensors $V_{[k]}(t)$ and the feedback tensors $\overset{1}{K}_{[k]}(t)$.

Hill calculated $\overset{1}{K}_{1}(t)$ and $\overset{1}{K}_{2}(t)$ for a two state, two control example system [3]. That was the first time that these terms were calculated using Buric's algorithm to the best of our knowledge.

The contributions of this work are in the development of a software package to calculate the feedback tensors and to perform the required algebraic operations on tensors for an arbitrary problem. There is a library of tensor subroutines which form the basis for the calculations, a program which calculates the feedback tensors, and a program which uses these tensors as a control law to calculate the propagation of states in the system. All programs are written in FORTRAN IV Plus.

There are six chapters in the thesis. The second chapter of this thesis contains some tensor background material and describes how symmetric tensors provide a natural method for representing polynomials. The third chapter describes a few of the subroutines and how they accomplish the desired algebraic operations. A derivation of the equations for the feedback tensors is in the fourth chapter. An example and some results are in the fifth chapter, and the conclusions are in Chapter VI. The software is in the Appendix.

CHAPTER II

TENSOR ALGEBRA

This chapter deals with two subjects. The first is some tensor background material. The notation and conventions are derived almost exclusively from Buric [1]. The presentation is very condensed and is meant to clarify the equations in the remainder of the thesis and not to be a thorough treatment of the subject. The second part of this chapter deals with the representation of polynomials using symmetric tensors. Such a representation is shown to be natural in the sense that there is a one to one correspondence between polynomials and symmetric tensors.

TENSOR BACKGROUND

Let V be a vector space of dimension n with basis $\{e_1, e_2, \ldots, e_n\}$. A mapping is said to be q-linear if it is linear with respect to each of its q arguments when all of the other arguments are held constant. Define a q-linear mapping

$$\otimes: \underbrace{V \times V \times \ldots \times V}_{q \text{ times}} \to V \otimes V \otimes \ldots \otimes V$$

by

$$\otimes\,(e_{i_1}, e_{i_2}, \ldots, e_{i_q}) = e_{i_1} \otimes e_{i_2} \otimes \ldots \otimes e_{i_q}.$$

Note that $\otimes$ is q-linear because

$$\otimes\,(e_{i_1}, \ldots, \underbrace{ae_\alpha + be_\beta}_{j\text{th position}}, \ldots, e_{i_q}) = e_{i_1} \otimes \ldots \otimes \underbrace{(ae_\alpha + be_\beta)}_{j\text{th position}} \otimes \ldots \otimes e_{i_q}$$

$$= a(e_{i_1} \otimes \ldots \otimes e_\alpha \otimes \ldots \otimes e_{i_q}) + b(e_{i_1} \otimes \ldots \otimes e_\beta \otimes \ldots \otimes e_{i_q})$$

for any $1 \leq j \leq q$. Then, for any q-linear mapping $\mu = \underbrace{V \times V \times \ldots \times V}_{q \text{ times}} \to U$, where U is any other vector space, there exists a unique linear map λ such that the diagram below commutes.

$$\begin{array}{ccc} \underbrace{V \times V \times \dots \times V}_{q \text{ times}} & \xrightarrow{\otimes} & \underbrace{V \otimes V \otimes \dots \otimes V}_{q \text{ times}} \\ & \searrow^{\mu} & \downarrow \lambda \\ & & U \end{array}$$

λ is defined by its action on the basis for $V \otimes V \otimes \dots \otimes V$, namely $\{e_{i_1} \otimes e_{i_2} \otimes \dots \otimes e_{i_q}\}$, where $1 \leq i_j \leq n$ for all j. There are n^q such basis elements. Let

$$\lambda(e_{i_1} \otimes e_{i_2} \otimes \dots \otimes e_{i_q}) = \mu(e_{i_1}, e_{i_2}, \dots, e_{i_q}).$$

Then, $\lambda \cdot \otimes = \mu$. As an example, let $\dim V = 2$ and $U = R$. Then the basis for $V \otimes V$ is $\{e_1 \otimes e_1,\ e_1 \otimes e_2,\ e_2 \otimes e_1,\ e_2 \otimes e_2\}$. Let two elements of V be x and y,

$$x = x_1e_1 + x_2e_2\ ,\ y = y_1e_1 + y_2e_2,$$

then the bilinear mapping μ must be of the form

$$\mu(x,y) = c_1x_1y_1 + c_2x_1y_2 + c_3x_2y_1 + c_4x_2y_2.$$

The linear map λ is defined by

$$\lambda(e_1 \otimes e_1) = c_1 \qquad \lambda(e_1 \otimes e_2) = c_2$$

$$\lambda(e_2 \otimes e_1) = c_3 \qquad \lambda(e_2 \otimes e_2) = c_4.$$

Then

$$\begin{aligned}(\lambda \cdot \otimes)(x,y) &= \lambda(x \otimes y) = \lambda((x_1e_1 + x_2e_2) \otimes (y_1e_1 + y_2e_2)) \\ &= \lambda(x_1e_1 \otimes (y_1e_1 + y_2e_2) + x_2e_2 \otimes (y_1e_1 + y_2e_2)) \\ &= \lambda(x_1y_1e_1 \otimes e_1 + x_1y_2e_1 \otimes e_2 + x_2y_1e_2 \otimes e_1 + x_2y_2e_2 \otimes e_2) \\ &= x_1y_1\lambda(e_1 \otimes e_1) + x_1y_2\lambda(e_1 \otimes e_2) + x_2y_1\lambda(e_2 \otimes e_1) \\ &\quad + x_2y_2\lambda(e_2 \otimes e_2) \\ &= x_1y_1c_1 + x_1y_2c_2 + x_2y_1c_3 + x_2y_2c_4 \\ &= \mu(x,y).\end{aligned}$$

Lines 2 and 3 are a result of the bilinearity of $\otimes$. Line 4 comes from the linearity of λ.

Let the dual space of V be V^*, the space of linear functionals on V. Find a basis of V^*, $\{e^1, e^2, \ldots, e^n\}$, such that the function e^i evaluated at e_j is one if $i = j$ and zero if $i \neq j$. That is,

$$e^i(e_j) = \delta^i_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}.$$

Now, define the bilinear map $\langle\cdot,\cdot\rangle$: $V^* \times V \to R$ to be

$$\langle e^i, e_j\rangle = e^i(e_j).$$

$\langle\cdot,\cdot\rangle$ is called a scalar product and is bilinear because

$$(e^{i_1} + e^{i_2})(e_j) = e^{i_1}(e_j) + e^{i_2}(e_j),$$

$$(ae^i)(e_j) = a(e^i(e_j)),$$

and

$$e^i(ae_{j_1} + be_{j_2}) = a(e^i(e_{j_1})) + b(e^i(e_{j_2})).$$

Notice that V is reflexive so that $V^{**} = V$ [4].

<u>Definition</u>: Given an n-dimensional vector space V and its dual space V^*, a tensor of degree (p,q) is a (p+q)-linear mapping

$$t^p_q : \underbrace{V^* \times \ldots \times V^*}_{p \text{ times}} \times \underbrace{V \times \ldots \times V}_{q \text{ times}} \to R.$$

This tensor is said to be contravariant of degree p and covariant of degree q [1]. This tensor is a member of the vector space of (p,q) tensors, called $T^p_q(V)$. The dimension of this tensor space is n^{p+q}. Corresponding to each tensor is a unique linear map λ which makes the following diagram commute.

$$\begin{tikzcd} \underbrace{V^* \times V^* \dots \times V^*}_{p \text{ times}} \times \underbrace{V \times \dots \times V}_{q \text{ times}} \arrow[r, "\otimes"] \arrow[dr, "t^p_q"'] & \underbrace{V^* \otimes V^* \otimes \dots V^*}_{p \text{ times}} \otimes \underbrace{V \otimes \dots \otimes V}_{q \text{ times}} \arrow[d, "\lambda"] \\ & R . \end{tikzcd}$$

To get a clearer view of λ, assume $p = 0$, $q = 2$; then the diagram is

$$\begin{tikzcd} V \times V \arrow[r, "\otimes"] \arrow[dr, "t^o_2"'] & V \otimes V \arrow[d, "\lambda"] \\ & R . \end{tikzcd}$$

Since λ is a linear map it is an element of the dual space of $V \otimes V$. This space is just $V^* \otimes V^*$. This can be seen by defining a scalar product

$$\langle e^i \otimes e^j,\ e_k \otimes e_\ell \rangle = \langle e^i, e_k \rangle \langle e^j, e_\ell \rangle .$$

This scalar product is again bilinear and all linear maps from $V \quad V$ to R are elements of $V^* \otimes V^*$. To see this, let

$$\lambda : V \otimes V \to R,$$

then find out the action of λ on each member of the basis of $V \otimes V$,

$$\lambda(e_i \otimes e_j) = \alpha_{ij} \in R.$$

Then, if $L = \sum_{k=1}^{n} \sum_{\ell=1}^{n} \alpha_{k\ell}\, e^k \otimes e^\ell$, we have $\langle L, e_i \otimes e_j \rangle$ equal to

$$\langle \sum_{k=1}^{n} \sum_{\ell=1}^{n} \alpha_{k\ell}\, e^k \otimes e^\ell,\ e_i \otimes e_j \rangle =$$

$$\sum_{k=1}^{n} \sum_{\ell=1}^{n} \alpha_{k\ell} \langle e^k \otimes e^\ell,\ e_i \otimes e_j \rangle =$$

$$\sum_{k=1}^{n} \sum_{\ell=1}^{n} \alpha_{k\ell} \langle e^k, e_i \rangle \langle e^\ell, e_j \rangle =$$

$$\sum_{k=1}^{n} \sum_{\ell=1}^{n} \alpha_{k\ell}\, \delta_i^k\, \delta_j^\ell = \alpha_{ij} = \lambda(e_i \otimes e_j).$$

So $V^* \otimes V^*$ can be seen as the dual of $V \otimes V$. By reversing the roles of V and V^*, $V \otimes V$ is the dual of $V^* \otimes V^*$. Extending this idea, the dual space of $\underbrace{V^* \otimes V^* \otimes \ldots \otimes V^*}_{p \text{ times}} \otimes \underbrace{V \otimes V \otimes \ldots \otimes V}_{q \text{ times}}$ is $\underbrace{V \otimes V \otimes \ldots \otimes V}_{p \text{ times}} \otimes \underbrace{V^* \otimes V^* \otimes \ldots \otimes V^*}_{q \text{ times}}$.

The scalar product associated with these spaces is

$$\langle e_{i_1} \otimes e_{i_2} \otimes \ldots e_{i_p} \otimes e^{j_1} \otimes e^{j_2} \otimes \ldots e^{j_q}, e^{k_1} \otimes e^{k_2} \otimes \ldots e^{k_p} \otimes e_{\ell_1} \otimes e_{\ell_2} \otimes \ldots \otimes e_{\ell_q} \rangle = \langle e^{k_1}, e_{i_1}\rangle\langle e^{k_2}, e_{i_2}\rangle \ldots \langle e^{j_1}, e_{\ell_1}\rangle\langle e^{j_2}, e_{\ell_2}\rangle \ldots \langle e^{j_q}, e_{\ell_q}\rangle.$$

This product is still bilinear. Since for every element of $T^p_q(V)$ there corresponds a unique element of this dual space, the two spaces are essentially the same. Thus,

$$t^p_q \in T^p_q(V) = \underbrace{V \otimes V \otimes \ldots \otimes V}_{p \text{ times}} \otimes \underbrace{V^* \otimes V^* \otimes \ldots \otimes V^*}_{q \text{ times}} = (\overset{p}{\otimes} V) \otimes (\overset{q}{\otimes} V^*).$$

Written in terms of the basis for $T^p_q(V)$,

$$t^p_q = \sum_{i_1,\ldots i_p, j_1, \ldots j_q} a^{i_1 i_2 \ldots i_p}_{j_1 j_2 \ldots j_q}\, e_{i_1} \otimes e_{i_2} \otimes \ldots e_{i_p} \otimes e^{j_1} \otimes e^{j_2} \otimes \ldots \otimes e^{j_q}.$$

Now define the tensor product of two tensors, t^p_q and τ^r_s, where

$$\tau^r_s = \sum b^{k_1 k_2 \ldots k_r}_{\ell_1 \ell_2 \ldots \ell_s}\, e_{k_1} \otimes e_{k_2} \otimes \ldots e_{k_r} \otimes e^{\ell_1} \otimes e^{\ell_2} \ldots \otimes e^{\ell_s},$$

to be

$$t^p_q \otimes \tau^r_s = \sum c^{i_1 i_2 \ldots i_p k_1 k_2 \ldots k_r}_{j_1 j_2 \ldots j_q \ell_1 \ell_2 \ldots \ell_s}\, e_{i_1} \otimes \ldots e_{i_p} \otimes e_{k_1} \otimes \ldots e_{k_r} \otimes e^{j_1} \otimes \ldots e^{j_q} \otimes e^{\ell_1} \ldots \otimes e^{\ell_s},$$

where

$$c^{i_1 i_2 \cdots i_p k_1 k_2 \cdots k_r}_{j_1 j_2 \cdots j_q \ell_1 \ell_2 \cdots \ell_s} = a^{i_1 i_2 \cdots i_p}_{j_1 j_2 \cdots j_q} \, b^{k_1 k_2 \cdots k_r}_{\ell_1 \ell_2 \cdots \ell_s} .$$

So $t^p_q \otimes \tau^r_s \in T^{p+r}_{q+s}(V)$ and $T^p_q(V) \otimes T^r_s(V) = T^{p+r}_{q+s}(V)$. As an example, let $p=q=r=s=1$, then

$$t^1_1 = \sum_{i=1}^{n} \sum_{j=1}^{n} a^i_j \, e_i \otimes e^j \quad , \quad \tau^1_1 = \sum_{k=1}^{n} \sum_{\ell=1}^{n} b^k_\ell \, e_k \otimes e^\ell ,$$

and

$$t^1_1 \otimes \tau^1_1 = \sum_{i=1}^{n} \sum_{k=1}^{n} \sum_{j=1}^{n} \sum_{\ell=1}^{n} c^{ik}_{j\ell} \, e_i \otimes e_k \otimes e^j \otimes e^\ell$$

where

$$c^{ik}_{j\ell} = a^i_j b^k_\ell .$$

This multiplication is associative and is distributive over addition. It should be noticed that the result of this tensor product has a basis which consists of a tensor product of basis elements from the primary space followed by a tensor product of basis elements from the dual space (that is, in the basis, the elements with the subscripts are written before those with superscripts, as in $e_i \otimes e_k \otimes e^j \otimes e^\ell$). This will be the convention used throughout this paper and it implies the commutativity of the following diagram

$$\begin{tikzcd}
(\overset{p}{\otimes} V) \otimes (\overset{q}{\otimes} V^*) \times (\overset{r}{\otimes} V) \otimes (\overset{s}{\otimes} V^*) \arrow[r, "\otimes"] \arrow[dr, "b"'] & (\overset{p+r}{\otimes} V) \otimes (\overset{q+s}{\otimes} V^*) \arrow[d, "\lambda"] \\
 & U
\end{tikzcd}$$

Here, b is a bilinear function and λ is the corresponding unique linear function.

One can have tensor products of more than one space, such as $U \otimes V$. If there are copies of the dual and primary space of each, one can write

$$t^{p,r}_{q,s} \in T^{p,r}_{q,s}(U,V) = (\overset{p}{\otimes}U) \otimes (\overset{r}{\otimes}V) \otimes (\overset{q}{\otimes}U^*) \otimes (\overset{s}{\otimes}V^*)$$

for

$$t^{p,r}_{q,s} = a^{i_1 \ldots i_p, k_1 \ldots k_r}_{j_1 \ldots j_q, \ell_1 \ldots \ell_s} w_{i_1} \otimes \ldots \otimes w_{i_p} \otimes e_{k_1} \otimes \ldots e_{k_r} \otimes w^{j_1} \otimes \ldots w^{j_q} \otimes e^{\ell_1} \otimes \ldots e^{\ell_s}$$

where $\{w_i\}$ is a basis for U and $\{w^i\}$ is a basis for U^*.

So far, the space $(\overset{p}{\otimes}V) \otimes (\overset{q}{\otimes}V^*)$ has only been referred to as the dual of $(\overset{p}{\otimes}V^*) \otimes (\overset{q}{\otimes}V)$. This space can also be seen as the space of linear functions f^p_q,

$$f^p_q : \overset{q}{\otimes}V \to \overset{p}{\otimes}V$$

where

$$f^p_q = \sum a^{i_1 i_2 \ldots i_p}_{j_1 j_2 \ldots j_q} e_{i_1} \otimes \ldots e_{i_p} \otimes e^{j_1} \otimes e^{j_2} \otimes \ldots \otimes e^{j_q}$$

and

$$b^q = \sum c^{k_1 k_2 \ldots k_q} e_{k_1} \otimes e_{k_2} \otimes \ldots \otimes e_{k_q}$$

and

$$f^p_q(b^q) = \sum a^{i_1 i_2 \ldots i_p}_{j_1 j_2 \ldots j_q} c^{k_1 k_2 \ldots k_q} e_{i_1} \otimes \ldots \otimes e_{i_p} \langle e^{j_1}, e_{k_1}\rangle \langle e^{j_2}, e_{k_2}\rangle \ldots \langle e^{j_q}, e_{k_q}\rangle$$

$$= \sum a^{i_1 i_2 \ldots i_p}_{j_1 j_2 \ldots j_q} c^{j_1 j_2 \ldots j_q} e_{i_1} \otimes e_{i_2} \ldots \otimes e_{i_p}.$$

This is a linear mapping, and any linear map from $\overset{q}{\otimes}V$ to $\overset{p}{\otimes}V$ can be written in this form. Similarly, the space $(\overset{p}{\otimes}V) \otimes (\overset{q}{\otimes}V^*)$ can be seen as the space of linear maps f^p_q,

$$f^p_q : (\overset{r}{\otimes}V^*) \otimes (\overset{q}{\otimes}V) \to \overset{p-r}{\otimes} V.$$

In order to incorporate all of the interpretations of this type into one theory, define the contraction operator $\overset{r}{\odot}$ in the following manner:

$$\overset{r}{\odot} : (U_1 \otimes V_1^* \otimes V_2^* \otimes \ldots \otimes V_r^*,\ V_1 \otimes V_2 \otimes \ldots \otimes V_r \otimes U_2) \to U_1 \otimes U_2$$

where

$$\overset{r}{\odot} (w_i \otimes e^{i_1} \otimes e^{i_2} \otimes \ldots \otimes e^{i_r},\ e_{j_1} \otimes e_{j_2} \otimes \ldots \otimes e_{j_r} \otimes v_j) =$$

$$\langle e^{i_1} \otimes e^{i_2} \otimes \ldots \otimes e^{i_r},\ e_{j_1} \otimes e_{j_2} \otimes \ldots \otimes e_{j_r} \rangle\, w_i \otimes v_j$$

where

e^{i_k} is a basis element in V_k^*

e_{j_k} is a basis element in V_k

w_i is a basis element in U_1

v_j is a basis element in U_2.

This operator maps two tensors to a third tensor. This map is bilinear. The two tensors must have the basis elements in the order above for the scalar products $\langle e^{i_k}, e_{j_k} \rangle$ to be well defined. That is, e^{i_k} must be a basis element in the dual space to the space containing e_{j_k}. The spaces U_1 and U_2 are optional and can refer to any spaces. For example, if U_1 and U_2 are not present, then $U_1 \otimes U_2 = R$ and the contraction operator maps $V^* \times V \to R$, where $V = V_1 \otimes V_2 \otimes \ldots \otimes V_r$. In this case, the operation is just taking the scalar product. If U_2 is not present, $V_i = V$, and $U_1 = \overset{p}{\otimes} V$, then the contraction operation accomplishes the map seen earlier,

$$\overset{q}{\odot} (f^p_q, b^q) =$$

$$\overset{q}{\odot} \left(\sum a^{i_1 i_2 \ldots i_p}_{j_1 j_2 \ldots j_q}\, e_{i_1} \otimes \ldots \otimes e^{j_q},\ \sum c^{k_1 k_2 \ldots k_q}\, e_{k_1} \otimes \ldots \otimes e_{k_q}\right) =$$

$$\sum a^{i_1 i_2 \cdots i_p}_{j_1 j_2 \cdots j_q} c^{k_1 k_2 \cdots k_q} e_{i_1} \otimes \cdots \otimes e_{i_p} \langle e^{j_1}, e_{k_1}\rangle\langle e^{j_2}, e_{k_2}\rangle \cdots \langle e^{j_q}, e_{k_q}\rangle =$$

$$\sum a^{i_1 i_2 \cdots i_p}_{j_1 j_2 \cdots j_q} c^{j_1 j_2 \cdots j_q} e_{i_1} \otimes e_{i_2} \cdots \otimes e_{i_p}.$$

Usually, this contraction is written as

$$f^{p}_{q} \overset{q}{\odot} b^q.$$

The contractions throughout the remainder of this paper will be written in this manner. As an example, let $p = 1$, $q = 2$, $n = 2$.

$$f^{p}_{q} = f^{1}_{2} = a^{1}_{11} e_1 \otimes e^1 \otimes e^1 + a^{1}_{12} e_1 \otimes e^1 \otimes e^2 + a^{1}_{21} e_1 \otimes e^2 \otimes e^1 + a^{1}_{22} e_1 \otimes e^2 \otimes$$
$$e^2 + a^{2}_{11} e_2 \otimes e^1 \otimes e^1 + a^{2}_{12} e_2 \otimes e^1 \otimes e^2 + a^{2}_{21} e_2 \otimes e^2 \otimes e^1 + a^{2}_{22} e_2 \otimes e^2 \otimes e^2$$

$$b^q = b^2 = c^{11} e_1 \otimes e_1 + c^{12} e_1 \otimes e_2 + c^{21} e_2 \otimes e_1 + c^{22} e_2 \otimes e_2.$$

Then

$$f^{p}_{q} \overset{q}{\odot} b^q = f^{1}_{2} \overset{2}{\odot} b^2 = \sum_{i,j,k,\ell,m=1}^{2} a^{i}_{jk} c^{\ell m} e_i \langle e^j, e_\ell\rangle\langle e^k, e_m\rangle$$

$$= (a^{1}_{11} c^{11} + a^{1}_{12} c^{12} + a^{1}_{21} c^{21} + a^{1}_{22} c^{22})\, e_1$$

$$+ (a^{2}_{11} c^{11} + a^{2}_{12} c^{12} + a^{2}_{21} c^{21} + a^{2}_{22} c^{22})\, e_2 .$$

For another example, let $A^{1}_{1,1} \in V \otimes U^* \otimes V^*$ and let $u \otimes x \in U \otimes V$, where $\dim U = m = 2$ and $\dim V = n = 2$, and $\{w_1, w_2\}$ is a basis for U, then

$$A^{1}_{1,1} = \sum_{i,j,k=1}^{2} a^{i}_{j,k} e_i \otimes w^j \otimes e^k$$

and

$$u \otimes x = \sum_{\ell,m=1}^{2} u^\ell x^m w_\ell \otimes e_m$$

and

$$A^{1}_{1,1} \overset{2}{\odot} u \otimes x = \Big(\sum_{i,j,k=1}^{2} a^{i}_{j,k} \; e_i \otimes w^j \otimes e^k \Big) \overset{2}{\odot} \Big(\sum_{\ell,m=1}^{2} u^{\ell} x^{m} w_{\ell} \otimes e_m \Big)$$

$$= \sum_{\ell,m,i,j,k=1}^{2} a^{i}_{j,k} \; u^{\ell} \, x^{m} \, e_i \; \langle w^j, w_{\ell} \rangle \langle e^k, e_m \rangle$$

$$= (a^{1}_{11} u^1 x^1 + a^{1}_{12} u^1 x^2 + a^{1}_{21} u^2 x^1 + a^{1}_{22} u^2 x^2) \; e_1$$

$$+ \; (a^{2}_{11} u^1 x^1 + a^{2}_{12} u^1 x^2 + a^{2}_{21} u^2 x^1 + a^{2}_{22} u^2 x^2) \; e_2 .$$

SYMMETRIC TENSORS

It is natural to talk about symmetric tensors when one is working in a polynomial space. Polynomials, by their very nature, do not need the full "unsymmetric" tensor basis for their representation. The symmetric tensor basis, which has considerably lower dimension than the unsymmetric tensor basis, is perfect for describing them. The symmetric tensor views a tensor product of basis elements which comes from a permutation of another tensor product of basis elements as redundant and thus does not include them. The method of implementing the symmetric tensor computations and transformations on the computer is nontrivial and is thus included. First, however, we must look at polynomials and see where the need for a symmetric tensor arises.

Let's assume we have a polynomial in $m + n$ variables denoted $\{u_1, u_2, \ldots, u_m, x_1, x_2, \ldots x_n\}$, The polynomial is then of the form

$$p(x,u) = \sum_{i=1}^{k} a_i u_1^{i_1} \, u_2^{i_2} \cdots u_m^{i_m} \, x_1^{i_{m+1}} \, x_2^{i_{m+2}} \cdots x_n^{i_{m+n}} \; .$$

There are k terms in this polynomial and the power to which each u_j or x_j is raised in each term is denoted i_j or i_{m+j} respectively. The a_i are the coefficients of the terms of the polynomial.

<u>Definition</u>: The <u>degree of p(x,u)</u> is the greatest sum of the powers of u_i and x_j, that is, the degree of $p(x,u) = d(p) = \max\limits_{1<i<k} \left(\sum_{j=1}^{m+n} i_j\right)$, where i_j is the power of u_j or x_{j-m} as above.

The <u>degree of a term</u> in p(x,u) is just the sum of the powers to which u_i and x_j are raised:

$$d(a_i u_i^{i_1} \cdots x_n^{i_{m+n}}) = \sum_{j=1}^{m+n} i_j \ .$$

The <u>degree of u</u> in a term is $d_u(a_i u_1^{i_1} u_2^{i_2} \cdots x_n^{i_{m+1}} \cdots x_n^{i_{m+n}}) = \sum_{j=1}^{m} i_j$.

Similarly, the <u>degree of x</u> in a term is $d_x(a_i u_1^{i_1} u_2^{i_2} \cdots x_1^{i_{m+1}} \cdots x_n^{i_{m+n}}) = \sum_{j=m+1}^{m+n} i_j$.

(Note: $d_u(a_i u_1^{i_1} \cdots x_n^{i_{m+n}}) + d_x(a_i u_1^{i_1} \cdots x_n^{i_{m+n}}) = d(a_i u_1^{i_1} \cdots x_n^{i_{m+n}})$.)

The terms of the polynomial are then separated by the degree of the terms. Thus, we now have sets of terms, each set having terms all of the same degree. Each set is now separated by the degree of u in each term. Thus, the set has subsets, each subset having terms with the same degree of u. Since each subset is contained in a larger set of terms of the same degree, the subsets will not only have the same degree of u, but also the same degree of x. An example is in order.

<u>Example 1</u>: Suppose m = 2, n = 3 and our polynomial is

$$p(x,u) = a_1u_1 + a_2u_2x_3 + a_3u_1u_2 + a_4u_1^2 + a_5x_2^2 + a_6u_1x_1$$
$$+ a_7u_1u_2x_1 + a_8u_1^2x_2 \ .$$

First we divide this polynomial into sets by the degree of the terms:

$\{a_1u_1\}$, $\{a_2u_2x_3,\ a_3u_1u_2,\ a_4u_1^2,\ a_5x_2^2,\ a_6u_1x_1\}$, $\{a_7u_1u_2x_1,\ a_8u_1^2x_2\}$.
Then we divide each set into subsets by the degree of u in each term,

$\{a_1u_1\}$, $\{\{a_5x_2^2\},\ \{a_2u_2x_3,\ a_6u_1x_1\},\ \{a_3u_1u_2,\ a_4u_1^2\}\}$, $\{a_7u_1u_2x_1,\ a_8u_1^2x_2\}$.

Note that the first and the third sets are unchanged by this subdivision since their elements are the same degree of u. The second set, however, has three subsets corresponding to terms with degree of u equal to zero, one, and two.

Now that we have a method of separating polynomials into distinct sets we can start looking at a means of describing the polynomials using only the coefficients. A method of doing this will require a convention for knowing which coefficient multiplies which powers of u_i and x_j. This convention depends on the sets and subsets above and an ordering of the possible products u_i and x_j in those subsets. Also, we will not want to distinguish, for example, between an x_1x_2 term and an x_2x_1 term. The space we will work in will be commutative under multiplication so these two terms will be the same. In the case of a degree two term it will be easy to see which x_ix_j are redundant. In higher order terms it may not be as easy to spot the redundant terms. Also, for each set of redundant terms we will want to pick one and only one representative for a basis.

Redundant terms arise from a permutation of the x_i in the terms. Thus x_ix_j arises from x_jx_i by the permutation

$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}.$$

Here the first term becomes the second and the second term is moved over to the first. In general, we will want to choose a basis such that any other term in the space which is not in the basis is just a term arising from a permutation on the elements of one of the basis terms.

For now we will look only at terms of the form

$$\prod_{j=1}^{p} x_{i_j} \text{ where } 1 \le i_j \le n,\ j = 1,\dots,p.$$

Permutations on these terms are described by

$$S_p = \begin{bmatrix} 1 & 2 \dots & p \\ \ell_1 & \ell_2 \dots & \ell_p \end{bmatrix}$$

where $1 \le \ell_i \le p$ and $\ell_i \ne \ell_j$ for $i \ne j$, ℓ_i integers. If $\sigma \in S_p$ is the permutation, $\sigma(i) = \ell_i$. When this permutation is applied to a term, the result is

$$\prod_{j=1}^{p} x_{i(\sigma(j))}.$$

Definition: $G_n^p \overset{\Delta}{=} [i_1 i_2 \dots i_p]$, $i_j = 1,\dots,n$, the sequences of p integers each integer ranging from 1 to n. All possible terms of degree p are of the form

$$\prod_{j=1}^{p} x_{i_j},\ [i_1 i_2 i_3 \dots i_p] \in G_n^p.$$

As was mentioned before, we want terms which arise from permutations of other terms to be regarded as the same, thus we group the terms into "orbits".

Definition: An orbit is a subset of G_n^p whose elements are obtained from an "orbit representative" by a permutation on the representative. The orbit representative is a sequence of integers $[i_1 i_2, \dots i_p]$ such that $i_1 \le i_2 \le \dots \le i_p$.

Each orbit contains one representative. To prove this, suppose $i_1 i_2 \dots i_p$ and $j_1 j_2 \dots j_p$ are distinct orbit representatives in the same orbit. Then there exists a maximum k such that $i_k \ne j_k$. Say $j_k > i_k$. Since $j_1 j_2 \dots j_p$ is obtained from $i_1 i_2 \dots i_p$ by a permutation, there is an $\ell < k$ such that $i_\ell = j_k$.

This contradicts $i_1 < i_2 < \ldots < i_\ell < \ldots < i_k < \ldots < i_p$, because $i_\ell > i_k$. Therefore $i_k = j_k$ for all $1 \leq k \leq p$. Therefore each orbit contains one orbit representative.

The orbits can be viewed as equivalence classes where the equivalence relation is the permutation of the integers. The orbit representatives then stand for the equivalence classes, and S_p the set of permutations. In the following, the term $\prod_{j=1}^{p} x_{i_j}$ and the sequence of integers $i_1 i_2 \ldots i_p$ will be used interchangeably.

The set of orbit representatives, $G_{p,n}$, has $\binom{p+n-1}{p}$ elements. Each orbit has $\left(\frac{p!}{m_1! m_2! \ldots m_n!} \right)$ elements, where m_i is the number of times $j_k = i$ in the term $\left(\prod_{k=1}^{p} x_{j_k} \right)$, $i = 1,\ldots,n$.

When talking about terms in the polynomial space of degree p, an orbit representative will be

$$x_1^{j_1} x_2^{j_2} \ldots x_n^{j_n} \quad , \quad j_1 + j_2 + \ldots j_n = p.$$

Note that, in this term, x_1 comes first then x_2 then x_3.... So this term corresponds to the sequence of integers

$$g = [\underbrace{11\ldots1}_{j_1 \text{ times}}\underbrace{22\ldots2}_{j_2 \text{ times}}3\ldots\underbrace{nn\ldots n}_{j_n \text{ times}}] \in G_{p,n} \subset G_n^p.$$

Any term

$$\prod_{j=1}^{p} x_{i_j},$$

such that

$$[i_1 i_2 \ldots i_p]$$

is in the same orbit as g will be considered the same as g.

We can now consider any polynomial in $\{x_1, x_2, \ldots, x_n\}$ to be the sum of orbit representatives times coefficients. Let us order the orbit representatives in a unique manner. The ordering will be lexicographic. Thus if $i_1i_2 \ldots i_p$ precedes $j_1j_2 \ldots j_p$, then for the minimum k for which $i_k \neq j_k$, then $i_k < j_k$. Assuming that the polynomial has terms of different degrees, we will put the orbit representatives from the lower degree terms first. For example, if n = 3 the ordering of orbit representatives will be (in vector form)

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_2 \\ x_2x_3 \\ x_3x_3 \\ x_1x_1x_1 \\ x_1x_1x_2 \\ x_1x_1x_3 \\ x_1x_2x_2 \\ x_1x_2x_3 \\ x_1x_3x_3 \\ x_2x_2x_2 \\ x_2x_2x_3 \\ x_3x_3x_3 \\ x_1x_1x_1x_1 \\ \vdots \end{bmatrix}$$

If the coefficients for the terms are ordered in the same manner (i.e., determined by the term they multiply), and loaded into a vector, then the polynomial would be the inner product of the two vectors. (Inner product of $[y_i]$ and $[z_i]$ is $\sum_i y_i z_i$.)

Getting back to our original problem, where the polynomial was a function of $\{u_1,u_2,\ldots u_m\}$ as well as $\{x_1,x_2,\ldots,x_n\}$, we must come up with a similar ordering scheme so the loading of the coefficients into a vector will be unique.

Take all possible polynomial terms (ignoring coefficients) and divide them into the sets and subsets determined by their degrees as described above. For each set of terms of degree p, order the subsets by the degree of u in the terms (the subset of terms whose degree of u is zero come first, the subset with degree of u equal to p come last). In the ordering of a subset whose degree of u is k, take the first element of the ordered set of orbit representatives, $G_{k,m}$, and multiply it by each element of the ordered set of orbit representatives of x_j, i.e., $G_{(p-k),n}$. Take in order the rest of the elements of $G_{k,m}$ and multiply each by the elements of $G_{(p-k),n}$, in order. This orders the subsets. The sets having been ordered by the degree of the terms, all elements are ordered. Another example is in order. Taking m = 2 and n = 3 and loading a vector of representatives in the order above we have Figure 2.1. Now loading a vector of coefficients in the same manner as before, the polynomial is again the inner product of the vectors.

The methods used so far have not used tensors at all, only the characteristics of polynomials. The vectors of ordered orbit representatives, however, are very similar to symmetric tensors. The set of symmetric tensor basis elements (over the same sets of variables $\{u_i\}$ and $\{x_i\}$) is bijectively related to the elements of the vector above.

$$
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ u_1 \\ u_2 \\ x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_2 \\ x_2x_3 \\ x_3x_3 \\ u_1x_1 \\ u_1x_2 \\ u_1x_3 \\ u_2x_1 \\ u_2x_2 \\ u_2x_3 \end{bmatrix}
\quad
\begin{vmatrix} u_1u_1 \\ u_1u_2 \\ u_2u_2 \\ x_1x_1x_1 \\ x_1x_1x_2 \\ x_1x_1x_3 \\ x_1x_2x_2 \\ \vdots \\ x_3x_3x_3 \\ u_1x_1x_1 \\ u_1x_1x_2 \\ u_1x_1x_3 \\ \vdots \\ u_1x_3x_3 \\ u_2x_1x_1 \\ u_2x_1x_2 \\ \vdots \end{vmatrix}
\quad
\begin{bmatrix} u_3x_2x_3 \\ u_3x_3x_3 \\ u_1u_1x_1 \\ u_1u_1x_2 \\ u_1u_1x_3 \\ u_1u_2x_1 \\ u_1u_2x_2 \\ u_1u_2x_3 \\ u_2u_2x_1 \\ u_2u_2x_2 \\ u_2u_2x_3 \\ u_1u_1u_1 \\ u_1u_1u_2 \\ u_1u_2u_2 \\ u_2u_2u_2 \\ x_1x_1x_1x_1 \\ \vdots \end{bmatrix}
$$

continued continued

Figure 2.1 Example Vector Loaded with All Possible Polynomial Combinations of the set $\{x_1,x_2,x_3,u_1,u_2\}$

To show this, first we must look at the generation of symmetric tensors. If $\{u_1, u_2, \ldots, u_m\}$ and $\{x_1, x_2, \ldots, x_n\}$ are viewed as elements of vector spaces of dimension m and n respectively, then they can be seen as $\{u_1w_1, u_2w_2, \ldots, u_mw_m\}$ and $\{x_1e_1, x_2e_2, \ldots, x_ne_m\}$ where $\{w_i\}$ is a basis for the space U of dimension m and $\{e_i\}$ is a basis for the space X of dimension n. Furthermore, the bases $\{w_i\}$ and $\{e_i\}$ are ordered so that $\{u_i\}$ and $\{x_i\}$ can be seen as vectors

$$\begin{bmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ u_m \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

where an element in the k^{th} position is understood to multiply the k^{th} basis element.

Having defined bases, a multilinear mapping μ

$$\mu : \underbrace{U \times U \times \ldots \times U}_{k \text{ times}} \times \underbrace{X \times X \times \ldots \times X}_{p-k \text{ times}} \rightarrow V,$$

where the dimension of V is ℓ, will be clearly defined if we say what it does to the basis elements. Let us further suppose μ is p-linear. Then we can form the tensor product

$$\otimes : \underbrace{U \times U \times \ldots \times U}_{k \text{ times}} \times \underbrace{X \times X \times \ldots \times X}_{p-k \text{ times}} \rightarrow \underbrace{U \otimes U \otimes \ldots \otimes U}_{k \text{ times}} \otimes \underbrace{X \otimes X \otimes \ldots \otimes X}_{p-k \text{ times}}$$

where $\otimes (w_{i_1}, w_{i_2}, \ldots, w_{i_k}, e_{j_1}, e_{j_2}, \ldots, e_{j_{p-k}}) = w_{i_1} \otimes w_{i_2} \otimes \ldots \otimes w_{i_k} \otimes e_{j_1}$

$$\otimes \ldots \otimes e_{j_{p-k}}.$$

Basis elements will be ordered lexicographically. Again, we will refer to sequences of integers interchangeably with the basis elements. That is, we will assume for $[i_1 i_2,\ldots,i_k]$ the basis element $[w_{i_1} \otimes w_{i_2} \otimes \ldots \otimes w_{i_k}]$. For $i = i_1 i_2 \ldots i_k$ and $j = j_1 j_2 \ldots j_k$, $1 \leq i_\gamma, j_\gamma \leq m$ order the elements of G_m^k such that if i precedes j, then for the smallest α with $i_\alpha \neq j_\alpha$ we have $i_\alpha < j_\alpha$. Similarly order G_n^{p-k}. Then the ordered basis for $\underbrace{U \otimes U \otimes \ldots \otimes U}_{k \text{ times}} \otimes \underbrace{X \otimes X \otimes \ldots \otimes X}_{p-k \text{ times}}$ is obtained by taking the tensor product of each element of the ordered group G_m^k and every element of G_n^{p-k} in order. Thus, if $k = m = n = 2$ and $p = 4$,

$$G_2^2 = G_m^k = \{11,\ 12,\ 21,\ 22\} = G_n^{p-k} .$$

The ordered basis is (using $\{w_i\}$ and $\{e_i\}$)

$$\begin{bmatrix} w_1 \otimes w_1 \otimes e_1 \otimes e_1 \\ w_1 \otimes w_1 \otimes e_1 \otimes e_2 \\ w_1 \otimes w_1 \otimes e_2 \otimes e_1 \\ w_1 \otimes w_1 \otimes e_2 \otimes e_2 \\ w_1 \otimes w_2 \otimes e_1 \otimes e_1 \\ w_1 \otimes w_2 \otimes e_1 \otimes e_2 \\ w_1 \otimes w_2 \otimes e_2 \otimes e_1 \\ w_1 \otimes w_2 \otimes e_2 \otimes e_2 \\ w_2 \otimes w_1 \otimes e_1 \otimes e_1 \\ w_2 \otimes w_1 \otimes e_1 \otimes e_2 \\ w_2 \otimes w_1 \otimes e_2 \otimes e_1 \\ w_2 \otimes w_1 \otimes e_2 \otimes e_2 \\ w_2 \otimes w_2 \otimes e_1 \otimes e_1 \\ w_2 \otimes w_2 \otimes e_1 \otimes e_2 \\ w_2 \otimes w_2 \otimes e_2 \otimes e_1 \\ w_2 \otimes w_2 \otimes e_2 \otimes e_2 \end{bmatrix}$$

Define the mapping

$$\pi_s = \overset{p}{\otimes} X \to \overset{p}{\otimes} X/AS^p$$

by

$$\pi_s(e_{i_1} \otimes e_{i_2} \otimes \ldots \otimes e_{i_p}) = \alpha^{j_1 j_2 \cdots j_p} e_{j_1} \otimes e_{j_2} \otimes \ldots \otimes e_{j_p}, \tag{1}$$

where $j = [j_1 j_2 \cdots j_p]$ is the representative of the orbit to which $i = [i_1 i_2 \ldots i_p]$ belongs, and

$$\alpha^{j_1 j_2 \cdots j_p} = \frac{m_1! m_2! \ldots m_n!}{p!},$$

where m_ℓ is the number of occurrences of the integer ℓ in the sequence of integers i, and $0! \overset{\Delta}{=} 1$. Equation (1) can be rewritten as

$$\pi_s(e_i) = \alpha^j e_j,$$

and e_j is referred to as the orbit representative for e_i. Defining

$$T^*(X) = R \oplus X \oplus X^2 \oplus X^3 \oplus \ldots, \; X^i = \overset{i}{\otimes} X,$$

then π_s can be extended in a unique way to

$$\pi_s = T^*(X) \to T^*(X)/AS^*,$$

where π_s is still defined by equation (1). Note

$$\pi_s(e_i) = \;_i, \; 1 \leqslant i \leqslant n. \tag{2}$$

The space $\overset{p}{\otimes} X/AS^p$ is the space generated by the orbit representatives e_j, $j \in G_{p,n}$. Extending this idea,

$$T^*(X)/AS^* = R \oplus X/AS^1 \oplus X^2/AS^2 \oplus X^3/AS^3 \oplus \ldots .$$

By equation (2) above, $X/AS^1 = X$. Each AS^p is the kernel of π_s applied to $\overset{p}{\otimes} X$. So if A is in AS^p, then $\pi_s(A) = 0$. If

$$A = \sum_{i \in G^p_n} a^i e_i,$$

then

$$\pi_s(A) = \sum_{j \in G_{p,n}} \alpha^j \Big(\sum_{i \in \Delta_j} a^i \Big) e_j, \tag{3}$$

where Δ_j is the orbit to which j belongs. Now, $\pi_s(A) = 0$ implies

$$\sum_{i\varepsilon\Delta_j} a^i = 0, \text{ for all } j \in G_{p,n}. \tag{4}$$

On the other hand, if the components of A satisfy (4), then by (3) $\pi_s(A) = 0$. Tensors whose components satisfy Equation (4) are called antisymmetric and are elements of AS^p. AS^* is an ideal in $T^*(X)$, where

$$AS^* = AS^1 \oplus AS^2 \oplus AS^3 \oplus \ldots$$

To see this, let B be in $\overset{q}{\otimes} X$; if

$$B = \sum_j b^j e_j$$

then

$$\begin{aligned}\pi_s(A \otimes B) &= \pi_s(\sum a^i b^j e_i \otimes e_j)\\ &= \sum_{\ell\varepsilon G_{p+q,n}} \alpha^\ell \Big(\sum_{k\varepsilon\Delta_\ell} a^{k_1 k_2 \ldots k_p} b^{k_{p+1}\ldots k_{p+q}} \Big) e_\ell,\end{aligned}$$

where Δ_ℓ is the orbit in G_n^{p+q} to which ℓ belongs. But

$$\sum_{k\varepsilon\Delta_\ell} a^{k_1\ldots k_p} b^{k_{p+1}\ldots k_{p+q}} = \sum_{\ell'} \sum_{k''\varepsilon\Delta_{\ell'}} \Big(\sum_{m\varepsilon\Delta_{k'}} a^m \Big) b^{k_{p+1}\ldots k_{p+q}},$$

where ℓ' is the orbit representative of any subset of q integers from

$\{\ell_1, \ell_2, \ldots, \ell_{p+q}\}$,

$\Delta_{\ell'}$ is the orbit to which ℓ' belongs,

$k'' = [k_{p+1} k_{p+2} \ldots k_{p+q}]$,

k' is the orbit representative of the set $\{\{\ell_1, \ell_2 \ldots, \ell_{p+q}\} - \{k_{p+1}, k_{p+2}, \ldots, k_{p+q}\}\}$,

$\Delta_{k'}$ is the orbit to which k' belongs.

And from Equation (4),

$$\sum_{m\varepsilon\Delta_{k'}} a^m = 0,$$

so

$$\pi_s(A \otimes B) = 0.$$

Similarly,

$$\pi_S(B \otimes A) = 0,$$

and AS^* is an ideal. The space $\overset{p}{\otimes}X/AS^p$ is the pth symmetric tensor space referred to earlier.

The development of the basis ordering for symmetric tensors is similar to that used earlier for the polynomials. The difference used here will be that each of the subsets earlier will correspond to different tensor spaces. Taking the direct sum of all of the tensor spaces, generate a new space whose basis is bijectively related to that of the polynomial space. For example, m = 2, n = 3 (as before for the polynomials) gives a basis

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ w_1 \\ w_2 \\ e_1 \otimes e_1 \\ e_1 \otimes e_2 \\ e_1 \otimes e_3 \\ e_2 \otimes e_2 \\ \vdots \\ e_3 \otimes e_3 \\ w_1 \otimes e_1 \\ \vdots \\ w_1 \otimes w_1 \\ w_1 \otimes w_2 \\ w_2 \otimes w_2 \\ \vdots \end{bmatrix}$$

Note: We use the same symbol $\otimes$ in the quotient. Other possible symbols, such as $\vee$, could also be used.

By taking tensor products of the vectors

$$u_1w_1 + \ldots + u_mw_m \quad \text{and} \quad x_1e_1 + x_2e_2 + \ldots + x_ne_n$$

terms of the form

$$u_ix_jw_i \otimes e_j$$

result. If we continue taking higher order tensor products, the result is polynomial coefficients of the basis elements. If we then apply π_s to the resulting tensors, then only the orbit representatives of polynomials and basis elements appear. Thus this symmetric tensor product is the same dimension as the polynomial vector space.

In second order tensors, looking at only x terms, $x = [x_1x_2]'$, the symmetric basis is $\{e_1 \otimes e_1, e_1 \otimes e_2, e_2 \otimes e_2\}$. All second degree polynomial terms are elements of (x_1^2, x_1x_2, and x_2^2)

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \otimes \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x2 & e_1 \otimes e_1 \\ x_1x_2 & e_1 \otimes e_2 \\ x_2x_1 & e_2 \otimes e_1 \\ x_2^2 & e_2 \otimes e_2 \end{bmatrix}$$

$$= \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2x_1 \\ x_2^2 \end{bmatrix}, \text{ (assuming the basis elements).}$$

Then, as described above, we don't want to distinguish between elements which arise from permutations of other elements, so we apply π_s. Here in matrix form,

$$\pi_S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ;$$

$$\pi_S(x \otimes x) = \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2 \end{bmatrix} .$$

All second degree polynomial terms can then be expressed as a unique linear operator acting on this tensor. (This linear operator can be viewed as a tensor and the polynomial generated by a contraction.) Let us look at a polynomial

$$5\,x_1^2 + 7\,x_1x_2 + 4\,x_2^2.$$

Using the first notation this is expressed as the inner product

$$[5\ 7\ 4] \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix} .$$

Using tensor notation this is

(unsymmetric)
$$\begin{bmatrix} 5 \\ a \\ 7-a \\ 4 \end{bmatrix} \overset{2}{\odot} \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2x_1 \\ x_2^2 \end{bmatrix}$$

where the operator

$$\begin{bmatrix} 5 \\ a \\ 7-a \\ 4 \end{bmatrix} \overset{2}{\odot}$$

takes the first element (x_1^2) and multiplies it by 5 and takes the second element (x_1x_2) and multiplies it by a, and so forth. Note that this is not unique. In symmetric notation this is:

$$\begin{bmatrix} 5 \\ 7 \\ 4 \end{bmatrix} \overset{(2)}{\odot} \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix} .$$

In symmetric notation this is unique. This carries through to higher order tensor products as well: every polynomial has a unique representation using symmetric tensors. This symmetric contraction has not been defined yet but its action here is obvious. Parentheses around the number above the contraction symbol indicate that it is a symmetric contraction.

When representing a polynomial in tensor form, the polynomial is expressed as a sum of coefficient tensors contracted with a tensor product of state and control vectors. Each tensor contraction is associated with one of the subsets described earlier. That is, the terms of the polynomial which have the same degree and have the same degree of u are isolated and will be involved in the same tensor contraction. Using the example on page 14, the terms of degree two with degree of u equal to two are $a_3u_1u_2$ and $a_4u_1^2$. The sum of these terms is represented by the tensor contraction

$$\begin{bmatrix} a_4 \\ a_3 \\ 0 \end{bmatrix} \overset{(2)}{\odot} \begin{bmatrix} u_1^2 \\ u_1u_2 \\ u_2^2 \end{bmatrix} = a_4u_1^2 + a_3u_1u_2.$$

As with the contraction above, each of the subsets can be associated with a unique tensor contraction. The ordering of components in coefficient tensors is determined from the pertinent lexicographic ordering.

Let the symmetric contraction be defined in a manner analogous to normal contraction. That is, for $V^{(r)} = V^r/AS^r$,

$$\overset{(r)}{\circledcirc} : U_1 \otimes V^{*(r)} \times V^{(r)} \otimes U_2 \to U_1 \otimes U_2$$

where

$$\overset{(r)}{\circledcirc} (w_i \otimes e^{i_1} \otimes e^{i_2} \otimes \ldots \otimes e^{i_r}, e_{j_1} \otimes e_{j_2} \otimes \ldots \otimes e_{j_r} \otimes v_j) =$$
$$\langle e^{i_1}, e_{j_1}\rangle\langle e^{i_2}, e_{j_2}\rangle \ldots \langle e^{i_r}, e_{j_r}\rangle\, w_i \otimes v_j$$

where

$\{e^i\}$ is a basis for V^*

$\{e_j\}$ is a basis for V

$\{w_i\}$ is a basis for U_1

$\{v_j\}$ is a basis for U_2

$$i_1 < i_2 < \ldots < i_r \quad \text{and} \quad j_1 < j_2 < \ldots < j_r.$$

This map is bilinear. The spaces U_1 and U_2 are optional and can be any linear spaces. For the polynomial representations above, the coefficient tensors are assumed to be in $V^{*(r)}$ and their basis elements were assumed relative to their positions in the tensors. Thus,

$$\begin{bmatrix} a_4 \\ a_3 \\ 0 \end{bmatrix} = a_4\, w^1 \otimes w^1 + a_3\, w^1 \otimes w^2 + 0\, w^2 \otimes w^2$$

and

$$\begin{bmatrix} a_4 \\ a_3 \\ 0 \end{bmatrix} \overset{(2)}{\circledcirc} \begin{bmatrix} u_1^2 \\ u_1 u_2 \\ u_2^2 \end{bmatrix} = (a_4\, w^1 \otimes w^1 + a_3\, w^1 \otimes w^2) \overset{(2)}{\circledcirc} (u_1^2 w_1 \otimes w_1 + u_1 u_2 w_1 \otimes w_2$$
$$+ u_2^2 w_2 \otimes w_2)$$
$$= a_4 u_1^2 + a_3 u_1 u_2 \,.$$

It has now been shown that every polynomial has a unique representation in terms of symmetric tensors. Furthermore, it has been shown that this representation is natural in the sense that each symmetric tensor is associated with a subset of terms of the polynomial obtained by separation by degrees. This is not the only tensor representation possible. As was discussed earlier, it is possible to use the unsymmetric tensor. This representation is not unique for nonlinear polynomials but is necessary for almost all of the computations used to calculate the feedback tensors. Since it is natural to write polynomials in terms of symmetric tensors, yet unsymmetric tensors are used for the calculations, a way of changing tensors from one type to the other is necessary. In the programs, this is accomplished using the subroutine SYM.

There are five ways subroutine SYM can be used, corresponding to the option argument, IOPT, of SYM equalling 1, 2, 3, 4, or 5. There are 2 separate options each for symmetric to unsymmetric and unsymmetric to symmetric transformations. One option is for a covariant and one is for a contravariant transformation. The fifth option is for an unsymmetric tensor to be transformed to a symmetric tensor then back to an unsymmetric tensor. This last option is useful because the resulting tensor is symmetric in another way. When a symmetric tensor is transformed into a larger dimension "unsymmetric" tensor, the new tensor will still have certain symmetric properties. The components of this tensor which are in the same orbit (the orbit corresponding to the basis elements involved) will all be equal. They will each be equal to the component of the reduced basis symmetric tensor corresponding to the orbit representative divided by the order of the orbit, if the tensor is contravariant in the basis elements involved (if the tensor is covariant in those basis elements, then each component will be equal to the corresponding component of the symmetric tensor). This new tensor, call it $A_{[p],[q]}$, will thus have the property that:

$$\text{if} \quad A_{[p],[q]} = a^{i}_{j,k} e_i \otimes w^{j_1} \otimes w^{j_2} \otimes \cdots \otimes w^{j_p} \otimes e^{k_1} \otimes e^{k_2} \otimes \cdots \otimes e^{k_q},$$

$$j = [j_1 j_2 \cdots j_p] \in G^{p}_{m},$$

$$k = [k_1 k_2 \cdots k_q] \in G^{q}_{n},$$

then $a^{i}_{\pi_1(j),\pi_2(k)} = a^{i}_{j,k}$, for all $\pi_1 \in S_p$, $\pi_2 \in S_q$. Square brackets around the indices will indicate this type of symmetry.

As noted above, the symmetrization of tensors is different for covariant and contravariant tensors. For contravariant tensors the method is analogous to the choice of polynomial representatives. For example, when choosing the quadratic representatives when $x \in V$, dim $V = 2$, it was seen that the proper symmetrization (or projection onto the symmetric space) was

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2 x_1 \\ x_2^2 \end{bmatrix} = \begin{bmatrix} x_1^2 \\ x_2 x_2 \\ x_2^2 \end{bmatrix}.$$

If one wishes to symmetrize a covariant tensor, such as the coefficients corresponding to the quadratic polynomials above, then the proper symmetrization is (assuming all coefficients are 1)

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

For the symmetrization to be valid, the result of contractions in the symmetric tensor space must be the same as contractions in the original tensor space. This is seen by

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \overset{2}{\odot} \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2x_1 \\ x_2^2 \end{bmatrix} = x_1^2 + x_1x_2 + x_2x_1 + x_2^2$$

and

$$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \overset{(2)}{\odot} \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix} = x_1^2 + 2x_1x_2 + x_2^2 .$$

The results are clearly equal since $x_1x_2 = x_2x_1$. The reverse process, going from the symmetric to the normal tensor space, also involves two types of symmetrization. For the example above, the contravariant process is

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix} = \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_1x_2 \\ x_2^2 \end{bmatrix} ,$$

while the covariant process is

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

For this symmetrization to be valid the result of the contraction in the tensor space must be the same as it is in the symmetric tensor space. For this example,

$$\begin{bmatrix}1\\1\\1\\1\end{bmatrix} \overset{2}{\odot} \begin{bmatrix}x_1^2\\x_1x_2\\x_1x_2\\x_2^2\end{bmatrix} = \begin{bmatrix}1&0&0\\0&1/2&0\\0&1/2&0\\0&0&1\end{bmatrix}\begin{bmatrix}1\\2\\1\end{bmatrix} \overset{2}{\odot} \begin{bmatrix}1&0&0\\0&1&0\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}x_1^2\\x_1x_2\\x_2^2\end{bmatrix}$$

$$= [1 \quad 2 \quad 1]\begin{bmatrix}1&0&0&0\\0&1/2&1/2&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0\\0&1&0\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}x_1^2\\x_1x_2\\x_2^2\end{bmatrix}$$

$$= [1 \quad 2 \quad 1]\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}x_1^2\\x_1x_2\\x_2^2\end{bmatrix} = \begin{bmatrix}1\\2\\1\end{bmatrix} \overset{(2)}{\odot} \begin{bmatrix}x_1^2\\x_1x_2\\x_2^2\end{bmatrix} .$$

In the general case, this is made more formal as follows. Select a tensor, $f_q^p \in T_q^p(V)$, dim $V = n$, and

$$f_q^p = \sum a^{i_1i_2\cdots i_p}_{j_1j_2\cdots j_q} e_{i_1} \otimes \cdots e_{i_p} \otimes e^{j_1} \otimes \cdots \otimes e^{j_q} .$$

Then define $f_q^{(p)}$ by

$$f_q^{(p)} = \sum_{(i),j} a_j^{(i)} e_{(i)} \otimes e^j, \text{ where } (i) \in G_{p,n},\ j \in G_n^q ,$$

and where

$$a_j^{(i)} = \sum_{\ell\in\Delta} \frac{1}{|\Delta|} a_j^{\ell} ,$$

$$j \in G_n^q ,\ \ell \in \Delta \subset G_n^p ,$$

Δ is the orbit in G_n^p to which i belongs,

$|\Delta|$ is the number of elements in Δ,

$$e(i) = e_{i_1} \otimes e_{i_2} \otimes \ldots \otimes e_{i_p},$$

and

$$e^j = e^{j_1} \otimes e^{j_2} \otimes \ldots \otimes e^{j_q}.$$

Similarly, $f^p_{(q)}$ is defined by

$$f^p_{(q)} = \sum_{i,(j)} a^i_{(j)}\, e_i \otimes e^{(j)}, \text{ where } i \in G^p_n,\ (j) \in G_{q,n},$$

and where

$$a^i_{(j)} = \sum_{k\varepsilon\Delta} a^i_k,$$

$$i \in G^p_n,\ k \in \Delta \subset G^q_n,$$

and Δ is the orbit in G^q_n to which j belongs.

Then, $f^{(p)}_{(q)}$ is defined by

$$f^{(p)}_{(q)} = \sum_{(i),(j)} a^{(i)}_{(j)}\, e(i) \quad e^{(j)}, \text{ where } (i) \in G_{p,n},\ (j) \in G_{q,n},$$

and where

$$a^{(i)}_{(j)} = \sum_{\ell\varepsilon\Delta_1, k\varepsilon\Delta_2} \frac{1}{|\Delta_1|}\, a^\ell_k,$$

$\ell \in \Delta_1$ the orbit in G^p_n to which i belongs,

and $k \in \Delta_2$ the orbit in G^q_n to which j belongs.

The difference between covariant and contravariant symmetrization is seen to be that one divides the sum by the order of the orbit for covariant symmetrization. The contravariant symmetrization is obtained by just summing over all elements in the orbit. In transforming f^p_q to $f^{(p)}_{(q)}$, one can take three approaches. The covariant and contravariant symmetrizations can be accomplished simultaneously or one can be done before the other; the result is the same for each. For notational purposes, $f^{(p)}_{q} \varepsilon T^{(p)}_{q}(V)$, $f^{p}_{(q)} \varepsilon T^{q}_{(q)}(V)$, and $f^{(p)}_{(q)} \varepsilon T^{(p)}_{(q)}(V)$.

The other type of symmetric tensor is a tensor in the normal tensor space obtained from a tensor in the symmetric tensor space. To obtain this tensor, the transformation from $T^{(p)}_{(q)}(V)$ to $T^{p}_{q}(V)$ must be defined. Let $f^{(p)}_{(q)} \varepsilon T^{(p)}_{(q)}(V)$, then $f^{[p]}_{[q]} \varepsilon T^{p}_{q}(V)$ is found in the following manner. If

$$f^{(p)}_{(q)} = \sum_{(\ell),(k)} a^{(\ell)}_{(k)} e_{(\ell)} \otimes e^{(k)},$$

where

$$(\ell) \varepsilon G_{p,n},\ (k) \varepsilon G_{q,n},$$

then

$$f^{[p]}_{[q]} = \sum_{i,j} a^{i}_{j} e_{i} \otimes e^{j},$$

where

$$i \varepsilon G^{p}_{n},\ j \varepsilon G^{q}_{n},$$

and where

$$a^{i}_{j} = \frac{1}{|\Delta_2|} a^{(\ell)}_{(k)},$$

Δ_2 is the orbit in G^{q}_{n} to which j belongs,

(k) is the orbit representative for Δ_2,

and (ℓ) is the orbit representative for the orbit in G^{p}_{n} to which i belongs. $f^{[p]}_{[q]}$ will have the property that any two components $a^{i_1}_{j_1}$ and $a^{i_2}_{j_2}$ will be equal if i_1 and i_2 come from the same orbit in G^{p}_{n} and j_1 and j_2 come from the same orbit in G^{q}_{n}. A tensor f^{p}_{q} is said to have a covariant part and a contravariant part, where "part" refers to the corresponding indices and basis elements. A basic fact about symmetrization is that symmetrization commutes with contraction in two cases. The first case is one where the symmetrization is performed over the part of the tensor which is not being contracted. The second case is any symmetrization when one of the tensors is symmetric (in the sense corresponding to the square brackets above) with respect to the part of the tensor being contracted. An example of the first case is shown with the contraction $f_q^p \overset{q}{\odot} b^q$. Let

$$f_q^p = \sum a^{i_1 i_2 \cdots i_p}_{j_1 j_2 \cdots j_q} e_{i_1} \otimes \cdots \otimes e_{i_p} \otimes e^{j_1} \otimes \cdots \otimes e^{j_q},$$

and
$$b^q = \sum c^{k_1 k_2 \cdots k_q} e_{k_1} \otimes \cdots \otimes e_{k_q}.$$

Then,

$$f_q^p \overset{q}{\odot} b^q = \sum a^{i_1 i_2 \cdots i_p}_{j_1 j_2 \cdots j_q} c^{j_1 j_2 \cdots j_q} e_{i_1} \otimes \cdots \otimes e_{i_p}.$$

The symmetrized version of the result is

$$\pi_{\mathfrak{S}}(f_q^p \overset{q}{\odot} b^q) = \sum_{(i),j} a_j^{(i)} c^j e_{(i)},$$

where

$$(i) \in G_{p,n},\ j \in G_n^q,$$

and where

$$a_j^{(i)} c^j = \sum_{\ell \in \Delta} \frac{1}{|\Delta|} a_j^{\ell} c^j,$$

$$j \in G_n^q,\ \ell \in G_n^p,$$

and Δ is the orbit in G_n^p to which i belongs.

If f_q^p had been symmetrized first, then

$$f_q^{(p)} \overset{q}{\odot} b^q = \Big(\sum_{(i),j} a_j^{(i)} e_{(i)} \otimes e^j\Big) \overset{q}{\odot} \Big(\sum c^k e_k\Big)$$

$$= \sum_{(i),j} a_j^{(i)} c^j e_{(i)}$$

where

$$a_j^{(i)} = \sum_{\ell \varepsilon \Delta} \frac{1}{|\Delta|} a_j^{\ell}.$$

This equals $\pi_S(f_q^p \overset{q}{\circledcirc} b^q)$ if $a_j^{(i)} c^j = (a_j^{(i)})c^j$. This equality holds since the sum is over ℓ not j. An example of the second case can be shown using the same contraction. Assume that $f_q^p = f_{[q]}^p$; that is $a_{j}^{i} = a_{\sigma(j)}^{i}$, for all $\sigma \varepsilon S_q$. Then $f_{[q]}^p \overset{q}{\circledcirc} b^q$ will be the same as before

$$f_{[q]}^p \overset{q}{\circledcirc} b^q = \sum a_{j_1 j_2 \cdots j_q}^{i_1 i_2 \cdots i_p} c^{j_1 j_2 \cdots j_q} e_{i_1} \otimes \ldots \otimes e_{i_p}$$

$$= \sum_{i,(\ell)} (a_\ell^i (\sum_{j \varepsilon \Delta} c^j)) e_i,$$

where

$$i \varepsilon G_n^p, (\ell) \varepsilon G_{q,n}, \ j \varepsilon G_n^q, \ \ell \text{ is any element from } \Delta,$$

and where

$j \varepsilon \Delta$, the orbit for which (ℓ) is the orbit representative,

the first sum runs over all $i \varepsilon G_n^p$ and $(\ell) \varepsilon G_{q,n}$,

the second sum runs over all $j \varepsilon \Delta$, and

$$e_i = e_{i_1} \otimes \ldots \otimes e_{i_p}.$$

The equality holds because the elements in each orbit Δ are equal. Now,

$$f_{[q]}^p \overset{q}{\circledcirc} b^q = \sum_{i,(\ell)} (a_\ell^i (\sum_{j \varepsilon \Delta} \frac{|\Delta|}{|\Delta|} c^j)) e_i$$

$$= \sum_{i,(\ell)} (|\Delta| \, a_\ell^i (\sum_{j \varepsilon \Delta} \frac{1}{|\Delta|} c^j)) e_i$$

$$= \sum_{i,(\ell)} d^{i}_{(\ell)} c^{(\ell)} e_i, \ (\ell) \in G_{q,n},$$

where

$$d^{i}_{(\ell)} = |\Delta| \, a^{i}_{\ell} = \sum_{j \in \Delta} a^{i}_{j}$$

and

$$c^{(\ell)} = \frac{1}{|\Delta|} \sum_{j \in \Delta} c^{j}.$$

But

$$f^{p}_{(q)} = \sum d^{i}_{(\ell)} \, e_i \otimes e^{(\ell)}$$

and

$$b(q) = \sum_{(\ell)} c^{(\ell)} \, e_{(\ell)},$$

thus

$$f^{p}_{(q)} \overset{q}{\circledast} b(q) = \sum_{i,(\ell)} d^{i}_{(\ell)} c^{(\ell)} e_i = f^{p}_{[q]} \overset{q}{\circledast} b^{q},$$

and the contraction operation commutes with symmetrization. The exact same result is easily established when $b^q = b^{[q]}$ is symmetric over the indices contracted. Note that this result applies only when the symmetrization of the part of the tensor over which there is a contraction is performed on both tensors (otherwise the contraction is not defined).

As stated above, the program SYM accomplishes the symmetrizations in the computations. SYM first checks to see if the input tensors and their associated dimensions make sense with the option requested. If these acceptable input conditions are met, SYM proceeds with the calculations. There are three main sections of the subroutine. The first two sections set up a map between components in the larger dimensional tensor from the normal tensor space and the smaller dimensional tensor in the symmetric tensor space. The third section carries out the map so defined and calculates the components of the desired tensor.

The first section of the subroutine associates with each tensor component in the larger tensor a base ten number. Tensors are stored as vectors and the base ten number is the location in the vector of the orbit representative corresponding to the component in question. First, the orbit representative for each component in the part of the tensor to be symmetrized is found. This is done by taking the sequence of integers corresponding to the basis element and reordering this sequence so that it is an orbit representative. This is done for the covariant and contravariant parts of the tensor, indices which depend both upon the control variable and state variable. From these sequences of integers, and the sequences corresponding to the parts of the tensor which are not symmetrized, the position of the orbit representative in the original large dimension tensor can be calculated. In general, if $i = [i_1 i_2 \ldots i_p] \in G_n^p$ is the sequence of integers, then this sequence is mapped to the number

$$\alpha_i = \sum_{j=1}^{p} (i_j - 1) n^{p-j} + 1.$$

This number can be viewed as the base ten number corresponding to the base n number

$$(i_1-1)(i_2-1)\ldots(i_{p-1}-1)(i_p-1) + 1$$

where the parentheses separate digits in a number and do not stand for multiplication. The "1" is added in each case so that the lowest integer sequence [11...1] corresponds to "1" and not "0". In other words, it corresponds to the first element of the tensor which in this FORTRAN is the "1" entry. If there are more than one parts of the tensor, then more than one number has been calculated, each standing for a part of the tensor. The number which shall be used to correspond to this tensor component will be the sum of each of these numbers times the highest number possible for the next lower part of the tensor. For instance, if $A^{r}_{p,q} \in (\overset{r}{\otimes}V) \otimes (\overset{p}{\otimes}U^{*}) \otimes (\overset{q}{\otimes}V^{*})$ and dim V = n, dim U = m, and one wishes to calculate the desired number for the component

$$a^{i_1 i_2 \ldots i_r}_{j_1 j_2 \ldots j_p, k_1 k_2 \ldots k_q},$$

then first the three numbers for the sequences of integers individually are calculated:

$$\alpha_i = (i_1-1)n^{r-1} + (i_2-1)n^{r-2} + \ldots + (i_{n-1}-1)n + (i_r-1) + 1,$$

$$\alpha_j = (j_1-1)m^{p-1} + (j_2-1)m^{p-2} + \ldots + j_p,$$

$$\alpha_k = (k_1-1)n^{q-1} + (k_2-1)n^{q-2} + \ldots + k_q.$$

The desired number is

$$\alpha = (\alpha_i-1)m^p n^q + (\alpha_j-1)n^q + \alpha_k.$$

(Note that one had to be subtracted from α_i and α_j prior to the multiplications. This is because the first element in the tensor must have a value of 1. Thus an alternative definition for the α_i, for $i \in G^p_n$, could be

$$\alpha_i = \sum_{j=1}^{p} (i_j-1)n^{p-j}.$$

Then the definition for α above would be

$$\alpha = \alpha_i \; m^p n^q + \alpha_j \; n^q + \alpha_k + 1.)$$

A vector of the same dimension as the large dimension tensor is used to store the number corresponding to each tensor component. Call this vector the large map vector.

The second section of the subroutine puts the distinct numerical values of the large map vector into another vector. Call this vector the small map vector. The small map vector will be of the same dimension as the smaller dimension symmetric tensor since each distinct numerical value in the large map vector corresponds to a different orbit. A third map vector, called the count vector, is used to count the number of repetitions of each distinct numerical value of the large map vector. Thus, each component of the count vector is the number of elements in a distinct orbit. The count vector is the same size as the small map vector.

The third section of the subroutine actually calculates the components of the new tensor. Say the desired tensor is the symmetric one. Each component of the symmetric tensor is set to zero. Then, the ith component of the original tensor is added to a particular component of the symmetric tensor. The particular component is found as follows: find the component of the small map vector which has the same value as the ith component of the large map vector. If this is the jth component, then the particular component of the symmetric tensor is the jth component. After these maps are complete the sums are divided by the corresponding components of the count vector for a contravariant symmetrization. For a covariant symmetrization the sums are the desired components. If the desired tensor is the larger dimension normal tensor, the process is similar. Each component of the larger tensor is set equal to a particular component of the symmetric tensor for contravariant symmetrization or equal to that component divided by the corresponding component of the count vector for covariant symmetrization. The "particular component" referred to here is the same as the "particular component" referred to above.

This chapter has presented two main ideas. First some basic tensor concepts were discussed. Secondly, the symmetric tensor was shown to be a natural representation for polynomials. This latter part went on to describe the relationship between symmetric and unsymmetric tensors and to describe the subroutine SYM. The next chapter will describe a few more subroutines and the functions they perform. A good understanding of these subroutines is helpful in following the somewhat involved derivations in Chapter IV.

CHAPTER III

PARTIAL SOFTWARE DESCRIPTION

This chapter describes several subroutines used to solve the problem. The functions they perform are described in general and are shown by way of example. Also, the actual FORTRAN statements used to carry out the subroutines are described.

PERM

The subroutine PERM is used to reorder the basis elements in a tensor. It changes the normal pattern of first incrementing the covariant x powers and then the covariant u to first incrementing the covariant u then the covariant x. Its purpose is to ease the computations and to make sense of some terms in the equations. Its need can be demonstrated using a typical term in which it appears:

$$(1,2)\ Q_{1,2} \odot K_2^1 \ .$$

This term arises from setting the coefficients of x^4 to zero, and thus originally comes from

$$Q_{1,2} \overset{3}{\odot} [(K_2^1 \overset{2}{\odot} x^2) \otimes x^2].$$

In this term, $Q_{1,2}$ is contracted three times with the expression in brackets. The expression in brackets has three contravariant powers: one contravariant u power coming from

$$K_2^1 \overset{2}{\odot} x^2$$

and two contravariant x powers from x^2. The result of the contraction operation is a scalar which is a fourth order polynomial. Thus, in setting the coefficients of x^4 to zero one must get a closed form expression for the coefficient of x^4. If one were to change the above original term to

$$(1,2)Q_{1,2} \overset{3}{\odot} [x^2 \odot (K_2^1 \overset{2}{\odot} x^2)]$$

where (1,2) indicates a reordering of the basis, then the result is the same as before. The next step is to change this to

$$[(1,2)\ Q_{1,2} \overset{1}{\odot} K_2^1] \overset{4}{\odot} x^4 .$$

The result is then the same, but the coefficient of x^4 has been isolated.

In the computations, this permutation must be carried out by the computer. Since the computer stores all tensors in vector form, the permutation must be described in terms of what it actually does to a vector. If $x \varepsilon R^2$ and $u \varepsilon R^2$, then Figure 3.1 shows how the permutation (1,2) $Q_{1,2}$ is accomplished. In the figure, part A shows what the actual scalar components of (1,2) $Q_{1,2}$ will be in terms of the scalar components of $Q_{1,2}$. This corresponds to the basis reordering shown in part B. Notice how the basis elements from the control space are after those from the state space, and therefore permuted first. If the tensor which is being permuted has contravariant powers, then, when computing the permuted tensor, the reordering of the covariant basis elements must occur each time the contravariant indices are incremented.

Let the original tensor be separated into three parts: the contravariant part, the covariant part which depends on the control variable, and the covariant part which depends on the state variable. Let the dimensions of these parts be I1, I2, and I3 respectively (e.g. if $x \varepsilon R^n$, $u \varepsilon R^m$, and the tensor has q covariant powers of the state variable, p covariant powers of the control variable, and r contravariant powers of the state variable, then $I1 = n^r$, A.

$$
\begin{aligned}
(1,2)\ Q_{1,2}\ (1) &= Q_{1,2}\ (1)\\
(1,2)\ Q_{1,2}\ (2) &= Q_{1,2}\ (5)\\
(1,2)\ Q_{1,2}\ (3) &= Q_{1,2}\ (2)\\
(1,2)\ Q_{1,2}\ (4) &= Q_{1,2}\ (6)\\
(1,2)\ Q_{1,2}\ (5) &= Q_{1,2}\ (3)\\
(1,2)\ Q_{1,2}\ (6) &= Q_{1,2}\ (7)\\
(1,2)\ Q_{1,2}\ (7) &= Q_{1,2}\ (4)\\
(1,2)\ Q_{1,2}\ (8) &= Q_{1,2}\ (8)
\end{aligned}
$$

B.

original ordered basis　　ordered basis after permutation

$$
\begin{bmatrix}
w^1 \otimes e^1 \otimes e^1 \\
w^1 \otimes e^1 \otimes e^2 \\
w^1 \otimes e^2 \otimes e^1 \\
w^1 \otimes e^2 \otimes e^2 \\
w^2 \otimes e^1 \otimes e^1 \\
w^2 \otimes e^1 \otimes e^2 \\
w^2 \otimes e^2 \otimes e^1 \\
w^2 \otimes e^2 \otimes e^2
\end{bmatrix}
\longrightarrow
\begin{bmatrix}
e^1 \otimes e^1 \otimes w^1 \\
e^1 \otimes e^1 \otimes w^2 \\
e^1 \otimes e^2 \otimes w^1 \\
e^1 \otimes e^2 \otimes w^2 \\
e^2 \otimes e^1 \otimes w^1 \\
e^2 \otimes e^1 \otimes w^2 \\
e^2 \otimes e^2 \otimes w^1 \\
e^2 \otimes e^2 \otimes w^2
\end{bmatrix}
$$

C.

```
      JA = 0
      DO   10 J1 = 1, I1
      DO   10 J2 = 1, I2
      DO   10 J3 = 1, I3
      JA = JA + 1
      JB = (J1-1) * I2 * I3 + (J3-1) * I2 + J2
   10 B(JB) = A(JA)
```

D.

```
      JA = 0
      DO 10 J2 = 1,2
      DO 10 J3 = 1,4
      JA = JA + 1
      JB = (J3-1) * 2 + J2
   10 B(JB) = A(JA)
```

```
CALL PERM (Q12,Q12P,0,1,2,I12,IDIMU,IDIMX,3,3)
```

Figure 3.1　The action of subroutine PERM is shown in A,B, and D, for the tensor $Q_{1,2}$. Part C shows the FORTRAN statements which accomplish a general permutation.

$I2 = m^p$, and $I3 = n^q$). Note that the dimension of the tensor is then the product of I1, I2, and I3. The lexicographic ordering in the tensor is then equivalent to ordering three indices, say J1, J2, and J3, lexicographically, where $J1 \in \{1,2,\ldots,I1\}$, $J2 \in \{1,2,\ldots I2\}$, and $J3 \in \{1,2,\ldots I3\}$. The permuted tensor would then have basis element ordering equivalent to the lexicographic ordering of J1, J3, J2, where these indices are as above. FORTRAN statements which fill a general permuted tensor B from the original tensor A are listed in part C of the figure. Notice that the three DO loops step through tensor A while the correct position of tensor B must be calculated at each step. The FORTRAN statements specifically for the case in parts A and B look like part D of the figure. Here $B = (1,2)\, Q_{1,2}$ and $A = Q_{1,2}$. Also, $I1 = 1$, $I2 = 2$, $I3 = 4$. In the original ordered basis, $w^{k_1} \otimes e^{k_2} \otimes e^{k_3}$,

J2 = 1	corresponds to	$k_1 = 1$		
J2 = 2	corresponds to	$k_1 = 2$		
J3 = 1	corresponds to	$k_2 = 1$	and	$k_3 = 1$
J3 = 2	corresponds to	$k_2 = 1$	and	$k_3 = 2$
J3 = 3	corresponds to	$k_2 = 2$	and	$k_3 = 1$
J3 = 4	corresponds to	$k_2 = 2$	and	$k_3 = 2$.

The ordered basis after permutation will be $e^{k_2} \otimes e^{k_3} \otimes w^{k_1}$, with k_1, k_2, and k_3 as above. It is thus seen, that incrementing J3 before J2, as the DO loops will, corresponds to the ordering of the original ordered basis, while incrementing J2 before J3 corresponds to the ordering of the basis after permutation. Thus, JA increments each time through the DO loops because it corresponds to the original basis, while JB is calculated as if J2 were incremented before J3.

Subroutine PERM does the permutations. The actual statement which calls PERM to execute the permutation talked about is also in Figure 3.1. What each argument stands for is discussed in the software. PERM is fully described in the software. It does have one option which was not described here. Occasionally a tensor will have two covariant parts which depend on the same variable (state or control). PERM can permute these two parts. A discussion of this would duplicate the discussion above, and is thus not included.

RAISE

The raising and lowering of powers becomes necessary when one wants to equate expressions like the following:

$$V_2 \overset{2}{\odot} x^2 \qquad x^* \odot V_1^1 \odot x \qquad x_2 \overset{2}{\odot} V^2 .$$

V_1^1 is obtained from V_2 by raising one covariant power (or equivalently V_2 is obtained from V_1^1 by lowering one contravariant power). V^2 is obtained from V_2 by raising two covariant powers. Raising and lowering powers change the type of the tensor. They do not, however, change the scalar components of the tensor. The result of raising or lowering powers is thus a tensor of exactly the same dimension and with exactly the same components as the original tensor. Often on the computer, the components are also stored in the same order as the original tensor. This is the case with V_2, V_1^1, and V^2; on the computer they are identical. The differences between them come in their uses. As shown above, they are used differently in contractions. Also, V_2 and V^2 are symmetrized differently. The difference in their symmetrization is described in Chapter II.

In some tensors, the raising or lowering of powers does change the order in which the components are stored. As an example, assume $x \in R^2$, $u \in R^2$ and it is desired to raise two powers of the state variable in $Q_{1,2}$. If the resultant tensor is Q_1^2, then Figure 3.2 shows what the components of Q_1^2 will be

$$Q_1^2\,(1) = Q_{1,2}\,(1)$$
$$Q_1^2\,(2) = Q_{1,2}\,(5)$$
$$Q_1^2\,(3) = Q_{1,2}\,(2)$$
$$Q_1^2\,(4) = Q_{1,2}\,(6)$$
$$Q_1^2\,(5) = Q_{1,2}\,(3)$$
$$Q_1^2\,(6) = Q_{1,2}\,(7)$$
$$Q_1^2\,(7) = Q_{1,2}\,(4)$$
$$Q_1^2\,(8) = Q_{1,2}\,(8)$$

original ordered basis — ordered basis after raising two powers

$$\begin{bmatrix} w^1 \otimes e^1 \otimes e^1 \\ w^1 \otimes e^1 \otimes e^2 \\ w^1 \otimes e^2 \otimes e^1 \\ w^1 \otimes e^2 \otimes e^2 \\ w^2 \otimes e^1 \otimes e^1 \\ w^2 \otimes e^1 \otimes e^2 \\ w^2 \otimes e^2 \otimes e^1 \\ w^2 \otimes e^2 \otimes e^2 \end{bmatrix} \longrightarrow \begin{bmatrix} e_1 \otimes e_1 \otimes w^1 \\ e_1 \otimes e_1 \otimes w^2 \\ e_1 \otimes e_2 \otimes w^1 \\ e_1 \otimes e_2 \otimes w^2 \\ e_2 \otimes e_1 \otimes w^1 \\ e_2 \otimes e_1 \otimes w^2 \\ e_2 \otimes e_2 \otimes w^1 \\ e_2 \otimes e_2 \otimes w^2 \end{bmatrix}$$

```
CALL RAISE(Q12,Q12R,0,1,2,2,1,0,IDIMU,IDIMX,I12,2)
```

Figure 3.2 The action of subroutine RAISE is shown for $Q_{1,2}$. The two covariant powers of the state variable are raised to get Q_1^2.

in terms of the components of $Q_{1,2}$. The reordered basis is shown as well as what a call of subroutine RAISE would look like to accomplish the raising of powers in Q12. The description of the arguments of RAISE is included in the comments in the software.

TRANS

Subroutine TRANS transposes tensors. A transposition is the simultaneous raising of the first covariant power and lowering of the first contravariant power. It is usually performed on tensors with only one contravariant power such as $A_{p,q}$. Thus if

$$A_{p,q} = \sum a^{i}_{j_1 j_2 \ldots j_p,\, k_1 k_2 \ldots k_q}\, e_i \otimes w^{j_1} \otimes w^{j_2} \otimes \ldots w^{j_p} \otimes e^{k_1} \otimes e^{k_2} \otimes \ldots \otimes e^{k_q},$$

then

$$A^{T}_{p,q} = \sum b^{j_1}_{i j_2 \ldots j_p, k_1 k_2 \ldots k_q}\, w_{j_1} \otimes e^{i} \otimes w^{j_2} \otimes \ldots w^{j_p} \otimes e^{k_1} \otimes e^{k_2} \otimes \ldots \otimes e^{k_q}$$

where

$$b^{j_1}_{i j_2 \ldots j_p, k_1 k_2 \ldots k_q} = a^{i}_{j_1 j_2 \ldots j_p, k_1 k_2 \ldots k_q}\ .$$

As an example, take $A_{1,1}$ and transpose it. For $u \in R^2$ and $x \in R^2$, Figure 3.3 shows what the components of $A^{T}_{1,1}$ are in terms of the components of $A_{1,1}$. It also shows the ordered bases before and after transposition. The actual FORTRAN statement which accomplishes the transposition is also included. The arguments of this statement are talked about in the software.

Transposition is used to factor out powers of x in terms like

$$(V_{[k]} \overset{k-1}{\odot} x^{[k-1]}) \odot (A_{[p][q]} \overset{p+q-1}{\odot} (K_{[n]} \overset{p-1}{\odot} \overset{n}{\odot}\, x^{[n]}) \otimes x^{[q]})\ .$$

This is discussed in greater detail in Chapter IV.

$$A^T_{1,1}\ (1) = A_{1,1}\ (1)$$
$$A^T_{1,1}\ (2) = A_{1,1}\ (2)$$
$$A^T_{1,1}\ (3) = A_{1,1}\ (5)$$
$$A^T_{1,1}\ (4) = A_{1,1}\ (6)$$
$$A^T_{1,1}\ (5) = A_{1,1}\ (3)$$
$$A^T_{1,1}\ (6) = A_{1,1}\ (4)$$
$$A^T_{1,1}\ (7) = A_{1,1}\ (7)$$
$$A^T_{1,1}\ (8) = A_{1,1}\ (8)$$

original ordered basis — new ordered basis

$$\begin{bmatrix} e_1 \otimes w^1 \otimes e^1 \\ e_1 \otimes w^1 \otimes e^2 \\ e_1 \otimes w^2 \otimes e^1 \\ e_1 \otimes w^2 \otimes e^2 \\ e_2 \otimes w^1 \otimes e^1 \\ e_2 \otimes w^1 \otimes e^2 \\ e_2 \otimes w^2 \otimes e^1 \\ e_2 \otimes w^2 \otimes e^2 \end{bmatrix} \longrightarrow \begin{bmatrix} w_1 \otimes e^1 \otimes e^1 \\ w_1 \otimes e^1 \otimes e^2 \\ w_1 \otimes e^2 \otimes e^1 \\ w_1 \otimes e^2 \otimes e^2 \\ w_2 \otimes e^1 \otimes e^1 \\ w_2 \otimes e^1 \otimes e^2 \\ w_2 \otimes e^2 \otimes e^1 \\ w_2 \otimes e^2 \otimes e^2 \end{bmatrix}$$

```
CALL TRANS(A11,A11T,1,1,1,I12,IDIMU,IDIMX,1)
```

Figure 3.3 The transpose of $A_{1,1}$ is shown here, and is calculated by subroutine TRANS.

TMULT

The subroutine TMULT performs tensor multiplications. It forms the tensor product of two input tensors. The output tensor's dimension will be the product of the dimensions of the input tensors. Each component of the output tensor is a product of components from the input tensors. Every possible product of one component from the first input tensor and one from the second is a component of the output tensor. The order in which these products are placed in the tensor depends upon the types of the generating tensors. There are two main types of tensor products used in the solution of the problem. They are

$$K^{i}_{j} \otimes K^{m}_{n}$$

and

$$V^{1}_{\ell} \otimes K^{i}_{j}.$$

This first tensor product is calculated as follows: if

$$K^{i}_{j} = \sum a^{\alpha_1\alpha_2\ldots\alpha_i}_{\gamma_1\gamma_2\ldots\gamma_j} w_{\alpha_1} \otimes w_{\alpha_2} \otimes \ldots w_{\alpha_i} \otimes e^{\gamma_1} \otimes e^{\gamma_2} \ldots \otimes e^{\gamma_j}$$

and

$$K^{m}_{n} = \sum b^{k_1k_2\ldots k_m}_{\ell_1\ell_2\ldots\ell_n} w_{k_1} \otimes w_{k_2} \otimes \ldots w_{k_m} \otimes e^{\ell_1} \otimes e^{\ell_2} \ldots \otimes e^{\ell_n}$$

then

$$K^{i}_{j} \otimes K^{m}_{n} = \sum c^{\alpha_1\alpha_2\ldots\alpha_i k_1k_2\ldots k_m}_{\gamma_1\gamma_2\ldots\gamma_j\ell_1\ell_2\ldots\ell_n} w_{\alpha_1} \otimes \ldots \otimes w_{\alpha_i} \otimes w_{k_1} \ldots \otimes w_{k_m} \otimes$$

$$e^{\gamma_1} \otimes \ldots e^{\gamma_j} \otimes e^{\ell_1} \ldots \otimes e^{\ell_n}$$

where

$$c^{\alpha_1\alpha_2\ldots\alpha_i k_1k_2\ldots k_m}_{\gamma_1\gamma_2\ldots\gamma_j\ell_1\ell_2\ldots\ell_n} = a^{\alpha_1\alpha_2\ldots\alpha_i}_{\gamma_1\gamma_2\ldots\gamma_j} b^{k_1k_2\ldots k_m}_{\ell_1\ell_2\ldots\ell_n}.$$

For the second tensor product, if

$$V^1_\ell = \sum v^t_{s_1 s_2 \ldots s_\ell} e_t \otimes e^{s_1} \otimes e^{s_2} \ldots \otimes e^{s_\ell}$$

then

$$V^1_\ell \otimes K^i_j = \sum c^{t\,\alpha_1\alpha_2\ldots\alpha_i}_{s_1 s_2 \ldots s_\ell \gamma_1 \gamma_2 \ldots \gamma_j} e_t \otimes w_{\alpha_1} \otimes \ldots w_{\alpha_i} \otimes e^{s_1} \otimes \ldots e^{s_\ell} \otimes e^{\gamma_1} \ldots e^{\gamma_j}$$

where

$$c^{t\,\alpha_1\alpha_2\ldots\alpha_i}_{s_1 s_2 \ldots s_\ell \gamma_1 \gamma_2 \ldots \gamma_j} = v^t_{s_1 s_2 \ldots s_\ell} a^{\alpha_1\alpha_2\ldots\alpha_i}_{\gamma_1\gamma_2\ldots\gamma_j}.$$

An example of the first tensor product and the actual calling statement from the program is shown in Figure 3.4. The product calculated is $K^1_1 \otimes K^1_1$, where

$$K^1_1 = \sum_{i=1,j=1}^{i=m,j=n} a^i_j w_i \otimes e^j,$$

and m=n=2.

Let each input tensor be separated into three parts: the contravariant part which depends on the state variable, the part which depends on the control variable and the covariant part which depends on the state variable. The second part can be either contravariant or covariant, but if the first tensor has covariant powers of the control variable, then the second cannot have contravariant powers of the control variable. (This is just a constraint due to how TMULT was set up. If necessary this constraint can be removed by first using TMULT, then using an appropriate option of PERM.) Let the dimensions of the three parts of the first tensor be N1, N3, and N5, and of the second be N2, N4, and N6, respectively. Then the lexicographic ordering for the output tensor is equivalent to the ordering of the six indices ranging from one to N1, N2, N3, N4, N5, and N6, respectively. The FORTRAN statements which calculate a general tensor product are shown in Figure 3.5, part A. The statements

$$
\begin{aligned}
K^1_1 \otimes K^1_1 &= \left(\sum a^i_j \, w_i \otimes e^j\right) \otimes \left(\sum a^k_\ell \, w_k \otimes e^\ell\right) \\
&= a^1_1 a^1_1 \, w_1 \otimes w_1 \otimes e^1 \otimes e^1 + a^1_1 a^1_2 \, w_1 \otimes w_1 \otimes e^1 \otimes e^2 \\
&+ a^1_2 a^1_1 \, w_1 \otimes w_1 \otimes e^2 \otimes e^1 + a^1_2 a^1_2 \, w_1 \otimes w_1 \otimes e^2 \otimes e^2 \\
&+ a^1_1 a^2_1 \, w_1 \otimes w_2 \otimes e^1 \otimes e^1 + a^1_1 a^2_2 \, w_1 \otimes w_2 \otimes e^1 \otimes e^2 \\
&+ a^1_2 a^2_1 \, w_1 \otimes w_2 \otimes e^2 \otimes e^1 + a^1_2 a^2_2 \, w_1 \otimes w_2 \otimes e^2 \otimes e^2 \\
&+ a^2_1 a^1_1 \, w_2 \otimes w_1 \otimes e^1 \otimes e^1 + a^2_1 a^1_2 \, w_2 \otimes w_1 \otimes e^1 \otimes e^2 \\
&+ a^2_2 a^1_1 \, w_2 \otimes w_1 \otimes e^2 \otimes e^1 + a^2_2 a^1_2 \, w_2 \otimes w_1 \otimes e^2 \otimes e^2 \\
&+ a^2_1 a^2_1 \, w_2 \otimes w_2 \otimes e^1 \otimes e^1 + a^2_1 a^2_2 \, w_2 \otimes w_2 \otimes e^1 \otimes e^2 \\
&+ a^2_2 a^2_1 \, w_2 \otimes w_2 \otimes e^2 \otimes e^1 + a^2_2 a^2_2 \, w_2 \otimes w_2 \otimes e^2 \otimes e^2
\end{aligned}
$$

```
CALL TMULT(K1,K1,K22,0,1,1,0,1,1,I11,I11,I22,IDIMU,IDIMX,NCALL)
```

Figure 3.4 The tensor product of K^1_1 with itself is shown here along with the program statement which calls TMULT to accomplish it.

A.

```
      J=0
      DO 10 I1=1,N1
      DO 10 I2=1,N2
      DO 10 I3=1,N3
      DO 10 I4=1,N4
      DO 10 I5=1,N5
      JA = I5+(I3-1)*N5+(I1-1)*N5*N3
      JB = (I4-0)*N6+(I2-1)*N6*N4
      DO 10 I6=1,N6
      J=JB+1
      JB=JB+1
10    C(J)=A(JA)*B(JB)
```

B.

```
      J=0
      DO 10 I3=1,2
      DO 10 I4=1,2
      DO 10 I5=1,2
      JA=I5+(I3-1)*2
      JB=(I4-1)*2
      DO 10 I6=1,2
      J=J+1
      JB=JB+1
10    C(J)=A(JA)*B(JB)
```

Figure 3.5 These are the FORTRAN statements used to calculate the tensor product of A and B in the subroutine TMULT.

specifically shown for the case shown in Figure 3.4 look like the statements in Figure 3.5, part B. In Figure 3.5, the input tensors are A and B, while the output tensor is C. The description of the arguments in TMULT is in the software.

TCONT

Subroutine TCONT performs the tensor contractions necessary in the calculations. This subroutine is very similar to TMULT in the manner in which it calculates the output tensor. The difference between TCONT and TMULT is that in TCONT, each input tensor is separated into four parts instead of three. The first input tensor's parts are: the contravariant part which depends on either the control or state variable, the covariant part which depends on the control variable and is not being contracted over, the state variable covariant part which is not being contracted over, and the covariant part which is being contracted over. The second input tensor's parts are: the contravariant part which is being contracted over, the contravariant part not being contracted over, and the two covariant parts. Let the dimensions of these eight parts be I1, I3, I5, I7, I7, I2, I4, and I6, respectively. Note that I7 is the dimension of the part being contracted in each tensor. The lexicographic ordering for the output tensor is equivalent to the ordering of six indices ranging from one to I1, I2, I3, I4, I5, and I6, respectively. Each element of the output tensor will be the sum of I7 products of two elements, one from each input tensor. The FORTRAN statements which accomplish this are shown in Figure 3.6. The similarity to Figure 3.5 is apparent.

There is another subroutine, TCONT1, which is almost identical to TCONT. TCONT1, however, has two additions. First, it does not zero the output tensor before calculating the output tensor. Thus the values originally in this output tensor are just added to the new values and this tensor can act as a "running sum" of several contractions. Second, a scalar multiplication option is attached. This allows one to calculate the normal contraction and then multiply the result by a scalar.

The arguments for both TCONT and TCONT1 are described in the software. Each also has several options which put restrictions on the type of input tensors allowed. For instance, if the contraction is over powers of the control variable, then there can be no covariant powers of the state variable in the first input tensor. These options and restrictions are also discussed in the software.

```
      DO 5 J=1,DIMC
5     C(J)=0.
      J=0
      DO 10 J1=1,I1
      DO 10 J2=1,I2
      DO 10 J3=1,I3
      DO 10 J4=1,I4
      DO 10 J5=1,I5
      DO 10 J6=1,I6
      J=J+1
      DO 10 J7=1,I7
      JA=J7+I7*(J5-1+I5*(J3-1+I3*(J1-1)))
      JB=J6+I6*(J4-1+I4*(J2-1+I2*(J7-1)))
10    C(J)=C(J)+A(JA)*B(JB)
```

Figure 3.6 These are the FORTRAN statements used to perform tensor contractions in subroutine TCONT.

CHAPTER IV

NONLINEAR OPTIMAL CONTROL

This chapter contains two main parts. The first part uses optimization theory to obtain the Hamilton-Jacobi-Bellman (HJB) equation for the problem studied. The second part uses tensor theory to solve the HJB equation for the optimal feedback.

DERIVATION OF HJB EQUATION

The problem to be studied is that of minimizing a cost functional

$$J = M(x(t_1)) + \int_{t_o}^{t_1} L(x(t),u(t),t)dt$$

subject to the system equation

$$\dot{x}(t) = f(x(t),u(t),t), \quad x(t_o) = x_o,$$

where $x(\cdot) \in R^n$ is a state vector, $u(\cdot) \in R^m$ is a control vector, and $t \in [t_o,t_1]$. The mapping $f : R^n \times R^m \times [t_o,t_1] \to R^n$ is analytic in x and u and continuously differentiable in t. f satisfies $f(0,0,t) = 0$.

The functional $L : R^n \times R^m \times [t_o,t_1] \to R$ is assumed positive definite, and $M : R^n \to R$ is positive semidefinite. Both L and M are assumed analytic in x and u. Let Ω denote the set of admissible controllers u(t), and assume Ω is compact. Under these conditions an optimal control $u^*(x,t)$ exists and is in Ω.

Because $u^*(x,t)$ minimizes J, $u^*(x,\tau)$ minimizes

$$V(x,t) = M(x(t_1)) + \int_{t}^{t_1} L(x,u,\tau)d\tau \text{ for all } t \in [t_o,t_1]. \qquad (1)$$

This is the Dynamic Programming approach. V(x,t) is called the optimal value function and for any time t, it represents the "cost-to-go" for the problem. The function V(x,t) is a positive function whose derivative with respect to t is negative. The derivative of $V(x,t)$ with respect to t satisfies

$$\left.\frac{dV(x,t)}{dt}\right|_{u^*,t} + L(x,u^*(x,t),t) = 0$$

when the optimal control $u^*(x,t)$ is applied. $u^*(x,t)$ satisfies this equation and is in fact the $u \in \Omega$ which satisfies

$$\min_{u\in\Omega} \left[\left.\frac{dV(x,t)}{dt}\right|_{u,t} + L(x,u,t)\right] = 0. \tag{2}$$

This is the famous Hamilton-Jacobi-Bellman equation and is developed with more rigor by Lee and Markus [5] and Buric [1]. This minimum is achieved by a $u^*(x,t) \in \Omega$ so this function also satisfies

$$\frac{\partial}{\partial u} \left[\left.\frac{dV(x,t)}{dt}\right|_{u,t} + L(x,u,t)\right] = 0. \tag{3}$$

Equations 2 and 3 will be used in the next section to solve for $V(x,t)$ and $u^*(x,t)$. First, however, a few more things must be noted about the form these equations take.

The function f was assumed to be analytic in x and u. This allows a series expansion for f in polynomial terms in the state and controller. This expansion can contain either a finite number of terms or an infinite number and if there are an infinite number f must be absolutely convergent in some neighborhood of $(x,u) = (0,0)$. The expansion is written in the form

$$f(x,u,t) = Ax + Bu + \sum_{p+q\geq 2} A_{[p],[q]} \overset{p+q}{\circledast} u^{[p]} \otimes x^{[q]}.$$

Here the $A_{[p],[q]}$ tensors have one contravariant power in R^n (the superscript is dropped for simplicity's sake), p covariant powers in R^m and q covariant powers in R^n. The pair (A,B) is assumed controllable. The functions M and L also admit series expansions and these expansions are assumed to start with quadratic terms. Thus,

$$M(x(t_1)) = \sum_{k=2} M_{[k]} \overset{k}{\odot} x(t_1)^{[k]}$$

and

$$L(x,u,t) = \sum_{i+j\geq 2} Q_{[i],[j]} \overset{i+j}{\odot} u^{[i]} \otimes x^{[j]}$$

$$= x^T Q x + u^T R u + Q_{1,1} \overset{2}{\odot} u \otimes x$$

$$+ \sum_{i+j\geq 3} Q_{[i],[j]} \overset{i+j}{\odot} u^{[i]} \otimes x^{[j]}.$$

Usually, the matrix representations for the linear part of f and the quadratic part of L are used. A, B, Q, and R are the standard matrix representations in the literature for the terms involved.

The class of admissible controls is defined as

$$\Omega = \{u(x,t) = \sum_{m=1} K^{1}_{[m]} \overset{m}{\odot} x^{[m]}\},$$

where the tensors $K^{1}_{[m]}$ are bounded piecewise continuous functions of t for t ε $[t_0,t_1]$. For this set of controls and using the series expansions for f, L, and M, Buric proves that V(x,t) has a series expansion,

$$V(x,t) = \sum_{k=2} V_{[k]} \overset{k}{\odot} x^{[k]}$$

where $V_{[k]}$ are piecewise continuously differentiable functions of t. From Equation (1), it is seen that $V(x,t_1) = M(x(t_1))$, thus $V_{[k]}(t_1) = M_{[k]}$, $k = 2,3,\ldots$. V(x,t) is found using this final condition and integrating backwards using the equation for the derivative of V(x,t) with respect to t, Equation (2). This integration is carried out after solving Equations (2) and (3) explicitly for the terms $V_{[k]}$ and $K^{1}_{[m]}$.

DERIVATION OF SOLUTION EQUATIONS

The Hamilton-Jacobi-Bellman equation for the problem is Equation (2). In this expression, the first term can be rewritten as

$$\frac{\partial}{\partial t} V(x,t)\Big|_{u,t} + \frac{\partial V(x,t)}{\partial x} \odot \frac{dx}{dt}\Big|_{u,t} \quad .$$

Now, it is known that

$$\frac{dx}{dt} = f(x,u,t) = \sum_{p+q \geq 1} A_{[p],[q]} \overset{p+q}{\odot} u^{[p]} \otimes x^{[q]}$$

and

$$\frac{\partial V(x,t)}{\partial x} = \sum_{k=2} \frac{\partial}{\partial x} V_{[k]} \overset{k}{\odot} x^{[k]} = \sum_{k=2} k\, V_{[k]} \overset{k-1}{\odot} x^{[k-1]} \, .$$

Substituting back into Equation (2),

$$\min_{u \in \Omega} [\sum_{k=2} \dot{V}_{[k]} \overset{k}{\odot} x^{[k]} + (\sum_{k=2} k\, V_{[k]} \overset{k-1}{\odot} x^{[k-1]}) \odot (\sum_{p+q \geq 1} A_{[p],[q]} \overset{p+q}{\odot} u^{[p]} \otimes x^{[q]})$$

$$+ \sum_{i+j \geq 2} Q_{[i],[j]} \overset{i+j}{\odot} u^{[i]} \otimes x^{[j]}] = 0 \, . \tag{4}$$

To find where the minimum of this expression occurs, the partial derivative with respect to u is taken, and set to zero. There are three terms here. The partial derivative of the first term with respect to u is zero. The partial derivatives of the other two terms with respect to u are:

$$\frac{\partial}{\partial u} [(\sum_{k=2} k\, V_{[k]} \overset{k-1}{\odot} x^{[k-1]}) \odot (\sum_{p+q \geq 1} A_{[p],[q]} \overset{p+q}{\odot} u^{[p]} \otimes x^{[q]})] =$$

$$\sum_{k=2} \sum_{p+q \geq 1} (k\, V_{[k]} \overset{k-1}{\odot} x^{[k-1]}) \odot (\frac{\partial}{\partial u} A_{[p],[q]} \overset{p+q}{\odot} u^{[p]} \otimes x^{[q]}) =$$

$$\sum_{k=2} \sum_{p+q \geq 1} (k\, V^{[k]} \overset{k-1}{\odot} x^{[k-1]}) \odot (p\, A_{[p],[q]} \overset{p+q-1}{\odot} u^{[p-1]} \otimes x^{[q]})$$

and

$$\frac{\partial}{\partial u}\left(\sum_{i+j\geq 2} Q_{[i],[j]} \overset{i+j}{\odot} u^{[i]} \otimes x^{[j]}\right) =$$

$$\sum_{i+j\geq 2} i\, Q_{[i],[j]} \overset{i+j-1}{\odot} u^{[i-1]} \otimes x^{[j]} .$$

Thus the minimum of the left member of Equation (4) occurs when

$$\sum_{k=2} \sum_{p+q\geq 1} (k\, V_{[k]} \overset{k-1}{\odot} x^{[k-1]}) \odot (p\, A_{[p],[q]} \overset{p+q-1}{\odot} u^{[p-1]} \otimes x^{[q]})$$

$$+ \sum_{i+j\geq 2} i\, Q_{[i],[j]} \overset{i+j-1}{\odot} u^{[i-1]} \otimes x^{[j]} = 0 . \qquad (5)$$

Also, at this minimum, Equation (4) will be satisfied. The acceptable controls u can be expressed as

$$u = \sum_m K^1_{[m]} \overset{m}{\odot} x^{[m]} .$$

Replacing u by this expression in Equations (4) and (5) gives, respectively:

$$\sum_{k=2} \dot{V}_{[k]} \overset{k}{\odot} x^{[k]} + \left(\sum_{k=2} k\, V_{[k]} \overset{k-1}{\odot} x^{[k-1]}\right) \odot$$

$$\left(\sum_{p+q\geq 1} A_{[p],[q]} \overset{p+q}{\odot} \left[\sum_m K^1_{[m]} \overset{m}{\odot} x^{[m]}\right]^{[p]} \otimes x^{[q]}\right)$$

$$+ \sum_{i+j\geq 2} Q_{[i],[j]} \overset{i+j}{\odot} \left(\sum_m K^1_{[m]} \overset{m}{\odot} x^{[m]}\right)^{[i]} \otimes x^{[j]} = 0 ; \qquad (6)$$

$$\sum_{k=2} \sum_{p+q\geq 1} (k\, V_{[k]} \overset{k-1}{\odot} x^{[k-1]}) \odot (p\, A_{[p],[q]} \overset{p+q-1}{\odot} \left[\sum_m K^1_{[m]} \overset{m}{\odot} x^{[m]}\right]^{[p-1]} \otimes x^{[q]})$$

$$+ \sum_{i+j\geq 2} i\, Q_{[i],[j]} \overset{i+j-1}{\odot} \left(\sum_m K^1_{[m]} \overset{m}{\odot} x^{[m]}\right)^{[i-1]} \otimes x^{[j]} = 0 . \qquad (7)$$

These are the two equations which will be used to find the values of the tensors for the optimal performance function ($V_{[k]}$) and for the feedback control ($K^1_{[m]}$). To get equations that are easier to work with the following must be done:

1. express each of the terms in Equations (6) and (7) as coefficient tensors multiplying tensor powers of x.
2. express each of the equations as coefficient tensors multiplying tensor powers of x.
3. set each of the coefficient tensors equal to zero.

The first term in Equation (6) is already in the desired form. A look at all of the other terms makes one realize that the first step must be comprehending the meaning of

$$[\sum_m K^1_{[m]} \overset{m}{\odot} x[m]]^{[\ell]} .$$

A very compact form for this expression is actually relatively easy to derive:

$$(\sum_{m_1} K^1_{[m_1]} \overset{m_1}{\odot} x^{[m_1]} \otimes (\sum_{m_2} K^1_{[m_2]} \overset{m_2}{\odot} x^{[m_2]}) \otimes \ldots \otimes (\sum_{m_\ell} K^1_{[m_\ell]} \overset{m_\ell}{\odot} x^{[m_\ell]}) \quad (8a)$$

$$= \sum_{m_1} \sum_{m_2} (K^1_{[m_1]} \otimes K^1_{[m_2]}) \overset{m_1+m_2}{\odot} x^{[m_1+m_2]}) \otimes \ldots \otimes (\sum_{m_\ell} K^1_{[m_\ell]} \overset{m_\ell}{\odot} x^{[m_\ell]}) \quad (8b)$$

$$= \sum_{m_1,m_2,\ldots,m_\ell} (K^1_{[m_1]} \otimes K^1_{[m_2]} \otimes \ldots \otimes K^1_{[m_\ell]}) \overset{m_1+m_2+\ldots+m_\ell}{\odot} x^{[m_1+m_2+\ldots+m_\ell]} \quad (8c)$$

$$= \sum_{n=\ell} K^\ell_{[n]} \overset{n}{\odot} x[n] . \quad (8d)$$

The step from (8b) to (8c) is just a generalization of the step from (8a) to (8b) in this derivation. To see that the step from (8a) to (8b) is valid, one must look at the tensor elements to see that they are the same. Since tensor multiplication is associative it is sufficient to show that

$$(K^1_{[m_1]} \overset{m_1}{\odot} x^{[m_1]}) \otimes (K^1_{[m_2]} \overset{m_2}{\odot} x^{[m_2]}) = (K^1_{[m_1]} \otimes K^1_{[m2]}) \overset{m_1+m_2}{\odot} x^{[m_1+m_2]} . \quad (9)$$

To show this,

$$K^1_{[m_1]} = \sum a^{i_1}_{j_1 j_2 \ldots j_{m_1}} e_{i_1} \otimes e^{j_1} \otimes e^{j_2} \otimes \ldots \otimes e^{j_{m_1}} ,$$

$$K^1_{[m_2]} = \sum b^{\alpha}_{\gamma_1\gamma_2\cdots\gamma_{m_2}} e_\alpha \otimes e^{\gamma_1} \otimes e^{\gamma_2} \otimes \ldots \otimes e^{\gamma_{m_2}} ,$$

$$x = \sum y^{i_2} e_{i_2} ,$$

$$K^1_{[m_1]} \overset{m_1}{\odot} x^{[m_1]} = \sum a^{i_1}_{j_1 j_2\cdots j_{m_1}} e_{i_1} \otimes e^{j_1} \otimes e^{j_2} \otimes \ldots \otimes e^{j_{m_1}} \overset{m_1}{\odot} (y^{i_2} e_{i_2})^{m_1}$$

$$= \sum a^{i_1}_{j_1 j_2\cdots j_{m_1}} y^{j_1} y^{j_2} \cdots y^{j_{m_1}} e_{i_1} .$$

Similarly,

$$K^1_{[m_2]} \overset{m_2}{\odot} x^{[m_2]} = \sum b^{\alpha}_{\gamma_1\gamma_2\cdots\gamma_{m_2}} y^{\gamma_1} y^{\gamma_2} \cdots y^{\gamma_{m_2}} e_\alpha .$$

Thus, the left side of Equation (9) is

$$\sum a^{i_1}_{j_1 j_2\cdots j_{m_1}} b^{\alpha}_{\gamma_1\gamma_2\cdots\gamma_{m_2}} y^{j_1} y^{j_2} \cdots y^{j_{m_1}} y^{\gamma_1} y^{\gamma_2} \cdots y^{\gamma_{m_2}} e_{i_1} \otimes e_\alpha. \tag{10}$$

Evaluating the right side,

$$K^1_{[m_1]} \otimes K^1_{[m_2]} = (\sum a^{i_1}_{j_1 j_2\cdots j_{m_1}} e_{i_1} \otimes e^{j_1} \otimes e^{j_2} \otimes \ldots \otimes e^{j_{m_1}}) \otimes$$

$$(\sum b^{\alpha}_{\gamma_1\gamma_2\cdots\gamma_{m_2}} e_\alpha \otimes e^{\gamma_1} \otimes e^{\gamma_2} \otimes \ldots \otimes e^{\gamma_{m_2}}) =$$

$$\sum a^{i_1}_{j_1 j_2\cdots j_{m_1}} b^{\alpha}_{\gamma_1\gamma_2\cdots\gamma_{m_2}} e_{i_1} \otimes e_\alpha \otimes e^{j_1} \otimes e^{j_2} \otimes \ldots \otimes e^{j_{m_1}} \otimes e^{\gamma_1} \otimes e^{\gamma_2} \otimes \ldots \otimes e^{\gamma_{m_2}})$$

where the ordering of the basis elements in the tensor product is according to the convention that the basis elements of the primary space are before those of the dual space. The right side of Equation (9) now becomes

$$(K^1_{[m_1]} \otimes K^1_{[m_2]}) \overset{m_1+m_2}{\odot} x^{[m_1+m_2]} = (\sum a^{i_1}_{j_1 j_2\cdots j_{m_1}} b^{\alpha}_{\gamma_1\gamma_2\cdots\gamma_{m_2}} e_{i_1} \otimes e_\alpha \otimes e^{j_1} \otimes \ldots$$

$$e^{\gamma_{m_2}}) \overset{m_1+m_2}{\odot} (\sum y^{i_2} e_{i_2})^{m_1+m_2}$$

$$= \sum a^{i_1}_{j_1 j_2 \cdots j_{m_1}} b^{\alpha}_{\gamma_1 \gamma_2 \cdots \gamma_{m_2}} y^{j_1} y^{j_2} \cdots y^{j_{m_1}} y^{\gamma_1} y^{\gamma_2} \cdots y^{\gamma_{m_2}} e_{i_1} \otimes e_{\alpha} . \quad (11)$$

Now the right hand side of Equation (11) is exactly the same as Equation (10), establishing the equality in Equation (9), and verifying the steps from a to c in Equation (8). Returning now to Equation (8) the step from c to d must be verified. This is easy because all that is done is the grouping together of all terms which have the same power of x in them. The sum of all of the coefficients of each power of x is then called $\overset{\ell}{K}_{[n]}$. The first thing that should be noticed from Equation (8c) is that the lowest tensor power of x which can occur is ℓ, because each of $m_1, m_2, \ldots m_\ell$ is greater than or equal to one. Each $\overset{\ell}{K}_{[n]}$ is thus a sum of coefficients of $x^{[n]}$ where $n \geq 1$. Specifically,

$$\overset{\ell}{K}_{[n]} = \sum \overset{1}{K}_{[i_1]} \otimes \overset{1}{K}_{[i_2]} \otimes \cdots \otimes \overset{1}{K}_{[i_\ell]} ,$$

where the sum ranges over all combinations of $i_1, i_2, \ldots i_\ell$ such that $i_1 + i_2 + \cdots + i_\ell = n$.

Returning now to Equation (6), the second term can be rewritten as

$$(\sum_{k=2} k\, V_{[k]} \overset{k-1}{\odot} x^{[k-1]}) \odot (\sum_{p+q \geq 1} A_{[p],[q]} \overset{p+q}{\odot} [\sum_{n} \overset{p}{K}_{[n]} \overset{n}{\odot} x^{[n]}] \otimes x^{[q]}). \quad (12)$$

The goal is to write this term as a sum of coefficient tensors multiplying powers of x. After rearranging the tensors involved, it is desired that it be clear what the coefficient for any power of x is. Starting with the second set of parentheses, let the components of the tensors be:

$$A_{[p],[q]} = \sum a^{\alpha}_{i_1 i_2 \cdots i_p, j_1 j_2 \cdots j_q} e_\alpha \otimes w^{i_1} \otimes w^{i_2} \otimes \cdots \otimes w^{i_p} \otimes e^{j_1} \otimes e^{j_2} \otimes \cdots \otimes e^{i_q}$$

$$K^{p}_{[n]} = \sum b^{\ell_1 \ell_2 \ldots \ell_p}_{m_1 m_2 \ldots m_n} w_{\ell_1} \otimes w_{\ell_2} \otimes \cdots \otimes w_{\ell_p} \otimes e^{m_1} \otimes e^{m_2} \otimes \cdots \otimes e^{m_n}$$

$$x = \sum y^i e_i \ . \tag{13}$$

To accomplish the goal, use a (p,q) permutation on $A_{[p],[q]}$ to reorder the basis elements. Applying this permutation to $A_{[p],[q]}$ gives

$$(p,q)\ A_{[p],[q]} = (p,q)(\sum a^{\alpha}_{i_1 i_2 \ldots i_p, j_1 j_2 \ldots j_q}\ e_\alpha \otimes w^{i_1} \otimes \ldots \otimes e^{j_q})$$

$$= \sum a^{\alpha}_{i_1 i_2 \ldots i_p, j_1 j_2 \ldots j_q}\ e_\alpha \otimes e^{j_1} \otimes e^{j_2} \otimes \ldots \otimes e^{j_q} \otimes w^{i_1} \otimes w^{i_2} \otimes \ldots \otimes w^{i_p} \ . \tag{14}$$

It is now asserted that

$$[(p,q)A_{[p],[q]} \overset{p}{\odot} K^{p}_{[n]}] \overset{q+n}{\odot} x^{[q+n]} = A_{[p],[q]} \overset{p+q}{\odot} ([K^{p}_{[n]} \overset{n}{\odot} x^{[n]}] \otimes x^{[q]}). \tag{15}$$

To show this, first compute the left side:

$$(p,q)A_{[p],[q]} \overset{p}{\odot} K^{p}_{[n]} = \sum a^{\alpha}_{i_1 i_2 \ldots i_p, j_1 j_2 \ldots j_q}\ b^{i_1 i_2 \ldots i_p}_{m_1 m_2 \ldots m_n}$$

$$e_\alpha \otimes e^{j_1} \otimes e^{j_2} \otimes \ldots \otimes e^{j_q} \otimes e^{m_1} \otimes \ldots \otimes e^{m_n}$$

and

$$[(p,q)A_{[p],[q]} \overset{p}{\odot} K^{p}_{[n]}] \overset{q+n}{\odot} x^{[q+n]} = \sum a^{\alpha}_{i_1 i_2 \ldots i_p, j_1 j_2 \ldots j_q}\ b^{i_1 i_2 \ldots i_p}_{m_1 m_2 \ldots m_n}$$

$$y^{j_1} y^{j_2} \ldots y^{j_q}\ y^{m_1} y^{m_2} \ldots y^{m_n}\ e_\alpha \ . \tag{16}$$

The right side of Equation (15) is computed as follows:

$$K^{p}_{[n]} \overset{n}{\odot} x^{[n]} = \sum b^{\ell_1 \ell_2 \ldots \ell_p}_{m_1 m_2 \ldots m_n}\ y^{m_1} y^{m_2} \ldots y^{m_n}\ w_{\ell_1} \otimes w_{\ell_2} \otimes \ldots \otimes w_{\ell_p}$$

$$A_{[p],[q]} \overset{p+q}{\odot} ([K^{p}_{[n]} \overset{n}{\odot} x^{[n]}] \otimes x^{[q]}) = (\sum a^{\alpha}_{i_1 i_2 \ldots i_p, j_1 j_2 \ldots j_q}$$

$$e_\alpha \otimes w^{i_1} \otimes \ldots \otimes w^{i_p} \otimes e^{j_1} \otimes \ldots \otimes e^{j_q}) \overset{p+q}{\odot}$$

$$(b^{\ell_1 \ell_2 \ldots \ell_p}_{m_1 m_2 \ldots m_n} \; y^{m_1} y^{m_2} \ldots y^{m_n} \; y^{\gamma_1} y^{\gamma_2} \ldots y^{\gamma_q}$$

$$w_{\ell_1} \otimes w_{\ell_2} \otimes \ldots \otimes w_{\ell_p} \otimes e_{\gamma_1} \otimes e_{\gamma_2} \otimes \ldots \otimes e_{\gamma_q})$$

$$= \sum a^{\alpha}_{i_1 i_2 \ldots i_p, j_1 j_2 \ldots j_q} \; b^{i_1 i_2 \ldots i_p}_{m_1 m_2 \ldots m_n} \; y^{m_1} y^{m_2} \ldots y^{m_n} \; y^{j_1} y^{j_2} \ldots y^{j_q} \; e_\alpha \, . \quad (17)$$

It is easily seen that the right hand side of Equation (17) is equal to the right hand side of Equation (16), verifying the equality in Equation (15). Looking at expression (12), it is seen that the term in the second set of parentheses is a vector in the state space. Because $V_{[k]}$ is symmetric with respect to its k covariant indices,

$$(V_{[k]} \overset{k-1}{\odot} x^{[k-1]}) \odot y = (V_{[k]} \odot y) \overset{k-1}{\odot} x^{[k-1]} \quad (18)$$

where y is any vector in the state space. Thus, from Equations (15) and (18), expression (12) is equivalent to

$$\sum_{k=2} \sum_{p+q\geq 1} (k \, V_{[k]} \odot [(p,q) \, A_{[p],[q]} \overset{p}{\odot} K^{p}_{[n]}]) \overset{k-1+q+n}{\odot} x^{[k-1+q+n]} \, .$$

Now let $m = k - 1 + q + n$, then the expression becomes

$$\sum_{m=2} \sum_{k=2}^{m} \sum_{q=0}^{m+1-k} (k \, V_{[k]} \odot [\sum_{p=\max(1-q,\min(n,1))}^{n} (p,q) \, A_{[p],[q]} \overset{p}{\odot} K^{p}_{[n]}]) \overset{m}{\odot} x^{[m]},$$

(19)

where $n = m - k + 1 - q$.

Returning again to Equation (6), the third term is rewritten as

$$\sum_{i+j\geq 2} Q_{[i],[j]} \overset{i+j}{\odot} (\sum_{n} K^{i}_{[n]} \overset{n}{\odot} x^{[n]}) \odot x^{[j]} \, .$$

Using an (i,j) permutation on $Q_{[i],[j]}$, this can be further simplified to

$$\sum_{i+j>2} \sum_{n} [(i,j)\, Q_{[i],[j]} \overset{i}{\odot} K^{i}_{[n]}] \overset{j+n}{\odot} x^{[j+n]} .$$

Now, letting $m = j+n$, this becomes

$$\sum_{m=2} \sum_{j=0}^{m} \sum_{i=\max(2-j,\min(m-j,1))}^{m-j} [(i,j)\, Q_{[i],[j]} \overset{i}{\odot} K^{i}_{[m-j]}] \overset{m}{\odot} x^{[m]} . \qquad (20)$$

Equation (6) can now be rewritten using expressions (19) and (20) as

$$\begin{aligned}
&\sum_{k=2} \dot{V}_{[k]} \overset{k}{\odot} x^{[k]} + \sum_{m=2} \sum_{k=2}^{m} \sum_{q=0}^{m+1-k} \sum_{p=\max(1-q,\min(n,1))}^{n} \\
&\qquad (k\, V_{[k]} \odot [(p,q)\, A_{[p],[q]} \overset{p}{\odot} K^{p}_{[n]}]) \overset{m}{\odot} x^{[m]} \\
&+ \sum_{m=2} \sum_{j=0}^{m} \sum_{i=\max(2-j,\min(m-j,1))}^{m-j} [(i,j)Q_{[i],[j]} \overset{i}{\odot} K^{i}_{[m-j]}] \overset{m}{\odot} x^{[m]} = 0 ,
\end{aligned}$$

where $n = m - k + 1 - q$.

It is now obvious that the coefficient for $x^{[m]}$, $m \geq 2$ is:

$$\begin{aligned}
&\dot{V}_{[m]} + \sum_{k=2}^{m} \sum_{q=0}^{m+1-k} \sum_{p=\max(1-q,\min(n,1))}^{n} (k\, V_{[k]} \odot [(p,q)\, A_{[p],[q]} \overset{p}{\odot} K^{p}_{[n]}]) \\
&\qquad + \sum_{j=0}^{m} \sum_{i=\max(2-j,\min(m-j,1))}^{m-j} [(i,j)\, Q_{[i],[j]} \overset{i}{\odot} K^{i}_{[m-j]}] , \qquad (21)
\end{aligned}$$

where $n = m - k + 1 - q$.

For Equation (6) to vanish it is a necessary and sufficient condition that the symmetric version of the coefficient tensors for every power of x vanish [1]. Thus, the expression (21) vanishes for all $m \geq 2$. This is the first set of equations used to find $V_{[m]}$ and $K^{1}_{[\ell]}$ for $m \geq 2$ and $\ell \geq 1$. The second set of equations are derived in a similar manner from Equation (7).

Shifting attention to Equation (7), it is seen that there are two terms which must be rearranged to determine the factors of the tensor powers of x. The first term is

$$\sum_{k=2}\ \sum_{p+q\geq 1} (k\, V_{[k]} \overset{k-1}{\odot} x^{[k-1]}) \odot (p\, A_{[p],[q]} \overset{p+q-1}{\odot} [\sum_{n} K^{p-1}_{[n]} \overset{n}{\odot} x^{[n]}] \otimes x^{[q]}). \tag{22}$$

Looking at the second set of parentheses, it is seen that the manner in which we previously factored out the powers of x will not work here because a (p,q) permutation on $A_{[p],[q]}$ followed by a contraction (p-1) times will leave one basis element from the dual of the control space, making a contraction over the state space variable impossible.

The approach taken involves several steps. First raise one power of $V_{[k]}$ so that the first set of parentheses becomes

$$V^{1}_{[k-1]} \overset{k-1}{\odot} x^{[k-1]} .$$

Second, transpose the term in the second set of parentheses. If this term is viewed as B^1_1, where

$$B^1_1 = \sum b^i_j\, e_i \otimes w^j ,$$

then

$$B^{1T}_1 = \sum c^j_i\, w_j \otimes e^i$$

where

$$c^j_i = b^i_j .$$

It can be seen that

$$(A_{[p],[q]} \overset{p+q-1}{\odot} [\sum_{n} K^{p-1}_{[n]} \overset{n}{\odot} x^{[n]}] \otimes x^{[q]})^T =$$

$$A^T_{[p],[q]} \overset{p+q-1}{\odot} [\sum_{n} K^{p-1}_{[n]} \overset{n}{\odot} x^{[n]}] \otimes x^{[q]} .$$

The third step is a permutation on $A^T_{[p],[q]}$. If

$$\overset{T}{A}_{[p],[q]} = \sum a^{i_1}_{ji_2\ldots i_p,k_1k_2\ldots k_q}\; w_{i_1} \otimes e^{j} \otimes w^{i_2} \otimes \ldots \otimes w^{i_p} \otimes e^{k_1} \otimes e^{k_2} \otimes \ldots \otimes e^{k_q},$$

let the components of $(p,q)\ \overset{T}{A}_{[p],[q]}$ where

$$(p,q)\ \overset{T}{A}_{[p],[q]} = \sum b^{i_1}_{k_1k_2\ldots k_q,ji_2\ldots i_p}\; w_{i_1} \otimes e^{k_1} \otimes e^{k_2} \otimes \ldots\ e^{k_q} \otimes e^{j} \otimes w^{i_2} \ldots \otimes w^{i_p}$$

be defined by

$$b^{i_1}_{k_1k_2\ldots k_q,ji_2\ldots i_p} = a^{i_1}_{ji_2\ldots i_p,k_1k_2\ldots k_q}\ .$$

In the computations this permutation is accomplished using subroutine PERM, option (11). Expression (22) is then equivalent to

$$\sum_{k=2}\ \sum_{p+q\geq 1}\ \sum_{n} k\ p\ [(p,q)\ \overset{T}{A}_{[p],[q]} \overset{p}{\odot} (\overset{1}{V}_{[k-1]} \otimes \overset{p-1}{K}_{[n]})] \overset{q+k-1+n}{\odot} x^{[q+k-1+n]}\ .$$

Setting m = q+k-1+n and being more careful with the limits of the summations, this becomes

$$\sum_{m=1}\ \sum_{k=2}^{m+1}\ \sum_{n=0}^{m-k+1}\ \sum_{p=\min(2,n+1)}^{n+1} kp[(p,q)\overset{T}{A}_{[p],[q]} \overset{p}{\odot} (\overset{1}{V}_{[k-1]} \otimes \overset{p-1}{K}_{[n]})] \overset{m}{\odot} x^{[m]}, \qquad (23)$$

where " $\overset{p-1}{K}_{[n]}$" is omitted when p = 1 and where q = m - k + 1 - n.

The second term in Equation (7) which must be rearranged is

$$\sum_{i+j\geq 2} i\ Q_{[i],[j]} \overset{i+j-1}{\odot} (\sum_{n} \overset{i-1}{K}_{[n]} \overset{n}{\odot} x^{[n]}) \otimes x^{[j]}\ . \qquad (24)$$

Let $\overset{1}{Q}_{[i-1],[j]}$ stand for $Q_{[i],[j]}$ after one of the covariant powers of the control variable has been raised. Then, using the permutation (i-1,j), expression (24) can be rewritten as

$$\sum_{i+j\geq 2}\ \sum_{n} [i\ (i-1,j)\ \overset{1}{Q}_{[i-1],[j]} \overset{i-1}{\odot} \overset{i-1}{K}_{[n]}] \overset{j+n}{\odot} x^{[j+n]}\ ,$$

or, setting m = j+n,

$$\sum_{m=1}^{\infty} \sum_{j=0}^{m} \sum_{i=\min(2,m-j+1)}^{m-j+1} [i\,(i-1,j)\, Q^{1}_{[i-1],[j]} \overset{i-1}{\odot} K^{i-1}_{[m-j]}] \overset{m}{\odot} x^{[m]} \ , \qquad (25)$$

where $\overset{i-1}{\odot}\ K^{i-1}_{[m-j]}$ is omitted when $i = 1$.

From expressions (23) and (25) Equation (7) now becomes

$$\sum_{m=1} \sum_{k=2}^{m+1} \sum_{n=0}^{m-k+1} \sum_{p=\min(2,n+1)}^{n+1} kp\ [(p,q) A^{T}_{[p],[q]} \overset{p}{\odot} (V^{1}_{[k-1]} \otimes K^{p-1}_{[n]})] \overset{m}{\odot} x^{[m]}$$

$$+ \sum_{m=1} \sum_{j=0}^{m} \sum_{i=\min(2,m-j+1)}^{m-j+1} [i(i-1,j)\ Q^{1}_{[i-1],[j]} \overset{i-1}{\odot} K^{i-1}_{[m-j]}] \overset{m}{\odot} x^{[m]} = 0, \qquad (26)$$

where $q = m - k + 1 - n$.

Again, for this equality to hold it is a necessary and sufficient condition for the symmetric version of the coefficients of each power of x to be zero. Thus, for $m \geq 1$,

$$\sum_{k=2}^{m+1} \sum_{n=0}^{m-k+1} \sum_{p=\min(2,n+1)}^{n+1} kp[(p,q) A^{T}_{[p],[q]} \overset{p}{\odot} (V^{1}_{[k-1]} \otimes K^{p-1}_{[n]})]$$

$$+ \sum_{j=0}^{m} \sum_{i=\min(2,m-j+1)}^{m-j+1} [i(i-1,j)\ Q^{1}_{[i-1],[j]} \overset{i-1}{\odot} K^{i-1}_{[m-j]}] = 0 \ , \qquad (27)$$

where $q = m - k + 1 - n$.

There are now two sets of equations which must be solved for $V_{[k]}$ and $K^{1}_{[\ell]}$. The solution is recursive in nature and is explained more fully in the next section.

DERIVATION OF CONTROLLER TERMS

In the previous section, two sets of equations were derived containing the controller tensors and the optimal performance tensors. These equations are solved for these tensors in terms of known tensors. The results which are derived in this section are:

1) The solution for $V_{[2]}$ is the solution for the Riccati equation obtained by truncating the system at linear terms and the performance index integrand at quadratic terms. $\overset{1}{K}_1$ is obtained as an affine function of $V_{[2]}$.

2) The equations for $V_{[m]}$ for $m \geq 3$ are first order linear differential equations which depend upon $V_{[2]}, V_{[3]}, \ldots, V_{[m-1]}$ and $\overset{1}{K}_1, \overset{1}{K}_{[2]}, \ldots, \overset{1}{K}_{[m-2]}$. $\overset{1}{K}_{[m-1]}$ is then obtained as an affine function of $V_{[m]}$. $\overset{1}{K}_{[m-1]}$ also depends upon $V_{[2]}, V_{[3]}, \ldots, V_{[m-1]}$ and $\overset{1}{K}_1, \overset{1}{K}_{[2]}, \ldots, \overset{1}{K}_{[m-2]}$.

First, look at (21) for m = 2 and (27) for m = 1:

$$\dot{V}_{[2]} + 2\, V_{[2]} \circledast (A_{1,0} \circledast \overset{1}{K}_1 + A_{0,1}) + Q_{[2],0} \overset{2}{\circledast} \overset{2}{K}_{[2]} + (1,1)\, Q_{1,1} \circledast \overset{1}{K}_1 + Q_{0,[2]} = 0,$$

$$2A^T_{1,0} \circledast V^1_1 + 2\overset{1}{Q}_{1,0} \circledast \overset{1}{K}_1 + \overset{1}{Q}_{0,1} = 0 .$$

Remembering that $A_{1,0} = B$, $A_{0,1} = A$, $Q_{[2],0} = R_{[2]}$, $Q_{0,[2]} = Q$, $\overset{1}{Q}_{1,0} = R$, and $\overset{2}{K}_{[2]} = \overset{1}{K}_1 \otimes \overset{1}{K}_1$, these become

$$\dot{V}_{[2]} + 2V_{[2]} \circledast (B \circledast \overset{1}{K}_1 + A) + R_{[2]} \overset{2}{\circledast} (\overset{1}{K}_1 \otimes \overset{1}{K}_1) + (1,1)Q_{1,1} \circledast \overset{1}{K}_1 + Q = 0 \tag{28}$$

$$2B^T \circledast V^1_1 + 2R \circledast \overset{1}{K}_1 + \overset{1}{Q}_{0,1} = 0 . \tag{29}$$

In order to solve Equation (28) for $V_{[2]}$, Equation (29) should be solved for $\overset{1}{K}_1$ and substituted in Equation (28). This is done most easily by writing Equation (29) in matrix form and obtaining

$$\overset{1}{K}_1 = - R^{-1} (B^T V + 2\overset{1}{Q}_{0,1}) . \tag{30}$$

Substituting into Equation (28),

$$\dot{V} - VBR^{-1} B^T V - VBR^{-1} \overset{1}{Q}_{0,1} + 2VA - \frac{1}{4} \overset{1}{Q}_{0,1}{}^T R^{-1} \overset{1}{Q}_{0,1} + Q = 0 . \tag{31}$$

This is the standard Riccati equation.

The first order differential equation for $\dot{V}_{[m]}$ is obtained from (21), while the equation involving $\overset{1}{K}_{[m-1]}$ is obtained from (27). In order to make (21) clearer, separate the terms involving $V_{[m]}$ and $\overset{1}{K}_{[m-1]}$, for $m > 3$.

$$\dot{V}_{[m]} + mV_{[m]} \odot (A_{1,0} \odot K_1^1 + A_{0,1}) + 2V_{[2]} \odot (A_{1,0} \odot \overset{1}{K}_{[m-1]}) +$$

$$(1,1)Q_{1,1} \odot \overset{1}{K}_{[m-1]} + 2Q_{[2],0} \overset{2}{\odot} (K_1^1 \otimes \overset{1}{K}_{[m-1]}) + F_{[m]} = 0 \,, \qquad (32)$$

where $F_{[m]}$ stands for the remaining terms.

The first idea here is that the symmetric version of this equation is zero. Also, since $V_{[m]}$ is symmetric it is desired to make its factor in this equation symmetric. This can be accomplished by substituting

$$V_{[m]} \overset{m}{\odot} \underbrace{(A_{1,0} \odot K_1^1 + A_{0,1}) \oplus (A_{1,0} \odot K_1^1 + A_{0,1}) \oplus \ldots \oplus (A_{1,0} \odot K_1^1 + A_{0,1})}_{m \text{ times}}$$

$$= V_{[m]} \overset{m}{\odot} (m \otimes (A_{1,0} \odot K_1^1 + A_{0,1}))$$

for $mV_{[m]} \odot (A_{1,0} \odot K_1^1 + A_{0,1})$, where "$m \otimes$" stands for the m-fold direct product. The result of the m-fold contraction will be a symmetric tensor. If $F_{[m]}$ is a symmetric tensor then $\dot{V}_{[m]}$ will be symmetric and thus $V_{[m]}$ will remain symmetric, because the terms involving $\overset{1}{K}_{[m-1]}$ vanish. The second idea here can be seen clearly if the tensors which are contracted with $\overset{1}{K}_{[m-1]}$ in Equation (32) are collected together:

$$(1,1)\ (2V_{[2]} \odot A_{1,0} + Q_{1,1} + 2Q_{[2],0} \odot K_1^1) \odot \overset{1}{K}_{[m-1]} \,.$$

The term in parentheses here is the same as the left side of Equation (29) with the contravariant power lowered and is thus zero. This means that $K^{1}_{[m-1]}$ does not appear in Equation (32) and $V_{[m]}$ therefore does not depend upon $K^{1}_{[m-1]}$.

Now look at Equation (27), for m>2. In this equation, the highest order controller tensor appearing is $K^{1}_{[m]}$. The highest order optimal control tensor appearing is $V_{[m+1]}$. Pulling the terms involving these tensors out of the summations gives

$$(m+1)\ B^T \circledcirc V^{1}_{[m]} + \sum_{k=2}^{m} \sum_{n=0}^{m-k+1} \sum_{p=\min(2,n+1)}^{n+1} kp[(p,q)A^{T}_{[p],[q]} \overset{p}{\circledcirc} (V^{1}_{[k-1]} \otimes K^{p-1}_{[n]})]$$

$$+ 2\ Q^{1}_{1,0} \circledcirc K^{1}_{[m]} + \sum_{i=3}^{m+1} i\ Q^{1}_{[i-1],0} \overset{i-1}{\circledcirc} K^{i-1}_{[m]} +$$

$$+ \sum_{j=1}^{m} \sum_{i=\min(2,m-j+1)}^{m-j+1} i(i-i,j)\ Q^{1}_{[i-1],[j]} \overset{i-1}{\circledcirc} K^{i-1}_{[m-j]} = 0\ ,$$

where $q = m-k+1-n$.

Remembering that $Q^{1}_{1,0}$ is equal to R, this can be rewritten as

$$-2\ R \circledcirc K^{1}_{[m]} = (m+1)\ B^T \circledcirc V^{1}_{[m]} + G^{1}_{[m]},$$

where $G^{1}_{[m]}$ is the symmetric version of the remaining terms. As in the solution for K^{1}_{1}, the fact that R is positive definite is used to obtain

$$K^{1}_{[m]} = -\frac{1}{2} R^{-1} \circledcirc \left[(m+1)B^T \circledcirc V^{1}_{[m]} + G^{1}_{[m]}\right]. \tag{33}$$

Here R^{-1} can be thought of as the tensor such that

$$R^{-1} \circledcirc R = R \circledcirc R^{-1} = (\delta^{i}_{j}),\ \text{where}\ \delta^{i}_{j} = 1\ \text{if}\ i=j,\ \text{and}\ \delta^{i}_{j} = 0\ \text{if}\ i \neq j.$$

To find R^{-1}, write R in matrix form, invert R, then view this resulting matrix as the tensor R^{-1}.

All of the results of this section have now been derived. $V_{[2]}$ was shown to be the solution to a Riccati equation and explicit expressions for $K^{1}_{[\ell]}$ and $V_{[m]}$, for $\ell>1$, $m>3$ have been derived. In the software, the subroutine TNSCLC calculates these tensors. TNSCLC uses all of the other subroutines discussed earlier which perform the tensor functions in Equations 30 through 33. The complicated nature of these equations reflects the complicated nature of TNSCLC. The use of the subroutines, however, lets one follow the calculations with the equations and simplifies the program greatly. The next chapter has examples which were implemented using this software. The subroutine TNSCLC is discussed in more detail in the comments in the software.

CHAPTER V

EXAMPLES

This chapter contains a few example problems. The first problem was studied by Lukes [2] and the results here will be compared to results obtained by Lukes and will thus serve as a verification of the software. The second problem was previously studied by Hill [3]. This problem is looked at closely and the third order feedback terms are shown to have a large effect. The final problem is a variation of the second problem in which an identified model of the system is used instead of the actual system tensors.

EXAMPLE 1

The first example is [2]:

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

$$\dot{x}_2 = x_1^2 - x_2^2 + \frac{u}{1-(x_1-x_2)u} ,$$

$$J = \int_0^\infty [x_1^2 + x_2^2 + \sin^2 u]dt.$$

The goal in this case is to calculate the steady state values of

$$u = \sum_{m=1}^\infty K^1_{[m]} \overset{m}{\odot} x^{[m]}; \; V = \sum_{k=2}^\infty V_{[k]} \overset{k}{\odot} x^{[k]}.$$

In order to see the values of the coefficient tensors more clearly, the system and cost functional are expanded into a power series about the origin, giving

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

$$\dot{x}_2 = u + x_1^2 - x_2^2 + (x_1-x_2)u^2 + (x_1-x_2)^2 u^3 + \dots ,$$

$$J = \int_0^\infty [x_1^2 + x_2^2 + u^2 - \frac{u^4}{3} + \frac{2u^6}{45} - \dots]dt.$$

The expansion for $\dot{x}_2$ is valid for $|(x_1-x_2)u|<1$. The values for A, B, $A_{(p),(q)}$, $Q_{(i),(j)}$, for $p+q>2$, $i+j>2$, can be read immediately from this expansion and are shown in Figure 5.1. Since $u \in R$, several of the tensors used for the calculations will be the same as those displayed. Namely, $A_{[2],0} = A_{(2),0}$, $A_{[3],0} = A_{(3),0}$, $A_{[2],1} = A_{(2),1}$, $Q_{[4],0} = Q_{(4),0}$, and $Q_{[6],0} = Q_{(6),0}$. It is worth noting that the $A = A_{0,1}$ tensor will not be stored as the matrix shown but will be stored, as all of the tensors are, in vector form. A and all of the tensors of Figure 5.1 which change when transformed to $A_{[p],[q]}$ or $Q_{[i],[j]}$ are shown in Figure 5.2.

These tensors are used in the software, and the system is integrated to steady state. The resulting optimal performance function terms and feedback tensor terms are shown in Figure 5.3. Also shown are the terms which were calculated by Lukes. These terms are identical to the terms calculated by the program. This partially verifies the software. As a verification of the third order terms, Lukes' hand calculations were carried out and the results agreed with those calculated by the program.

EXAMPLE 2

This example is a two state, two control example. It has been used by others for identification purposes [7], [8], [9], [10], and by Hill for calculating optimal feedback tensors [3]. Hill calculated $V_{[2]}$, $\overset{1}{K}_{1}$, V_3, and $\overset{1}{K}_{2}$ for this problem. In his calculations he did not use the symmetric tensor algebra or the contraction operator. Also, his software was very problem specific. Here the software was developed taking advantage of the flexibility which the symmetric tensor algebra and the contraction operator allow. The software was written in a very general form, and thus its application to this problem is an easy task.

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad A_{(2),0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$A_{1,1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad A_{0,(2)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}, \quad A_{(3),0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad A_{(2),1} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix},$$

$$A_{1,(2)} = 0, \; A_{0,(3)} = 0, \; A_{(i),(j)} = 0, \; i+j = 4,$$

$$A_{(3),(2)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ -2 \\ 1 \end{bmatrix}, \quad \text{otherwise } A_{(i),(j)} = 0, \; i+j = 5 \text{ or } 6.$$

$$R = Q_{(2),0} = Q_{[2],0} = [1], \quad Q_{1,1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad Q_{0,2} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix},$$

$$Q_{(i),(j)} = 0, \text{ for } i+j = 3, \; Q_{(4),0} = [-\tfrac{1}{3}], \; Q_{(6),0} = [2/45],$$

$$\text{otherwise } Q_{(i),(j)} = 0, \text{ for } i+j = 4,5,6, \text{ or } 7.$$

Figure 5.1 Symmetric System and Cost Functional Tensors for Example 1.

$$A = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad A_{0,[2]} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}, \quad A_{[3],[2]} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}, \quad Q_{0,[2]} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Figure 5.2 Unsymmetrized Tensors for Example 1.

Program: $$V_{(2)} = \begin{bmatrix} 1.732 \\ 2.000 \\ 1.732 \end{bmatrix}, \quad K^{1}_{1} = \begin{bmatrix} -1.000 \\ -1.732 \end{bmatrix}, \quad V_{(3)} = \begin{bmatrix} .6598 \\ 2.000 \\ .9897 \\ -.4762 \end{bmatrix}$$

$$K^{1}_{(2)} = \begin{bmatrix} -1.000 \\ -.9897 \\ .7143 \end{bmatrix}, \quad V_{(4)} = \begin{bmatrix} .8256 \\ 2.667 \\ 2.383 \\ -.5426 \\ -1.879 \end{bmatrix}, \quad K^{1}_{(3)} = \begin{bmatrix} -.178[illegible]-6 \\ -.919[illegible] \\ -6.114 \\ -5.707 \end{bmatrix}.$$

Lukes: $$V_{(2)} = \begin{bmatrix} \sqrt{3} \\ 2 \\ \sqrt{3} \end{bmatrix}, \quad K^{1}_{1} = \begin{bmatrix} -1 \\ -\sqrt{3} \end{bmatrix}, \quad V_{(3)} = \begin{bmatrix} \frac{8}{21}\sqrt{3} \\ 2 \\ \frac{4}{7}\sqrt{3} \\ -\frac{10}{21} \end{bmatrix}, \quad K^{1}_{(2)} = \begin{bmatrix} -1 \\ -\frac{4}{7}\sqrt{3} \\ \frac{5}{7} \end{bmatrix}$$

Figure 5.3

The problem is to minimize

$$(1) \qquad J = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \overset{(2)}{\odot} x^{(2)} + \int_{o}^{5} \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix} \overset{(2)}{\odot} x^{(2)} + \begin{bmatrix} 6 \\ 0 \\ 0 \\ 6 \end{bmatrix} \overset{2}{\odot} u \quad x + \begin{bmatrix} 5 \\ 0 \\ 5 \end{bmatrix} \overset{(2)}{\odot} u^{(2)} \; dt$$

subject to the system equation

$$\dot{x}_1 = u_2 \cosh x_1 x_2 - e^{2u_1} \sinh 2x_1 - 3 \sinh x_2$$

$$(2) \qquad x_2 = e^{u_1 u_2} \sinh x_1 - e^{u_1} u_1 \cosh x_1^2 + \sinh x_2$$

$$x(0) = x_o.$$

For this problem, the integration is over a finite time span whereas for the last problem there was an infinite time horizon. So here the feedback tensors must be saved after each integration step in order to simulate more closely the continuous feedback of time varying tensors contracted with the time varying states. The tensors calculated for the last example were the steady state values.

The tensors for the problem which are of covariant degree three or less are shown in Figure 5.4. The optimal cost tensors $V_{(2)}$, $V_{(3)}$, and $V_{(4)}$, and the optimal feedback tensors K^1_1, $K^1_{(2)}$, and $K^1_{(3)}$, were calculated and their values were saved every .05 second. Tables 5.1 through 5.6 show the values calculated for these tensors. These values are presented in graphical form in Appendix C.

$$A = \begin{bmatrix} -2 & -3 \\ 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$

$$A_{0,[2]} = 0, \; A_{1,[2]} = 0,$$

$$A_{1,1} = [-4,0,0,0,0,0,0,0]$$

$$A_{[2],0}(5) = -1, \; A_{[2],0}(j) = 0, \; 1 \leq j \leq 8, \; j \neq 5,$$

$$A_{0,[3]} = [-4/3,0,0,0,0,0,0,-.5,1/6,0,0,0,0,0,0,1/6]$$

$$A_{[3],0}(9) = -.5, \; A_{[3],0}(j) = 0, \; 1 \leq j \leq 16, \; j \neq 9,$$

$$A_{[2],1} = [-2,0,0,0,0,0,0,0,0,0,.5,0,.5,0,0,0]$$

Figure 5.4 System Tensors for Example 2

These tensors were used to calculate the trajectories of the system for various initial conditions. The program STAB performs an exhaustive search of the phase plane for this problem to determine the "region of usefulness" of the feedback tensors. This determination is done in the same spirit as Lyapunov stability theory [14]. A system is said to be stable if for every $\varepsilon > 0$ there exists a δ such that if $||x(t_o)|| < \delta$ then $||x(t)|| < \varepsilon$ for all $t > t_o$. A system is said to be asymptotically stable if it is stable and there exists a δ_1 such that for any $\mu > 0$ there exists a T such that if $||x(t_o)|| < \delta_1$ then $||x(t)|| < \mu$ for all $t > t_o + T$. Note that both δ and δ_1 may depend on t_o, δ may depend on ε and T may depend on μ, δ_1 and t_o. Also δ need not equal δ_1.

In applying these notions of stability to the finite time case it is desired to incorporate both the elements of asymptotic stability. For the stability part, ε is chosen as the largest value allowed on the digital computer.

TIME	V2S(1)	V2S(2)	V2S(3)
0.000E+00	0.1792E+00	0.3254E+00	0.5178E+00
0.500E-01	0.1797E+00	0.3257E+00	0.5175E+00
0.100E+00	0.1801E+00	0.3262E+00	0.5174E+00
0.150E+00	0.1807E+00	0.3270E+00	0.5174E+00
0.200E+00	0.1813E+00	0.3280E+00	0.5175E+00
0.250E+00	0.1819E+00	0.3293E+00	0.5179E+00
0.300E+00	0.1826E+00	0.3308E+00	0.5184E+00
0.350E+00	0.1832E+00	0.3326E+00	0.5192E+00
0.400E+00	0.1839E+00	0.3345E+00	0.5203E+00
0.450E+00	0.1845E+00	0.3366E+00	0.5216E+00
0.500E+00	0.1851E+00	0.3389E+00	0.5231E+00
0.550E+00	0.1856E+00	0.3411E+00	0.5250E+00
0.600E+00	0.1860E+00	0.3434E+00	0.5270E+00
0.650E+00	0.1863E+00	0.3456E+00	0.5293E+00
0.700E+00	0.1865E+00	0.3476E+00	0.5318E+00
0.750E+00	0.1865E+00	0.3494E+00	0.5343E+00
0.800E+00	0.1865E+00	0.3509E+00	0.5370E+00
0.850E+00	0.1863E+00	0.3520E+00	0.5396E+00
0.900E+00	0.1860E+00	0.3526E+00	0.5422E+00
0.950E+00	0.1856E+00	0.3528E+00	0.5446E+00
0.100E+01	0.1851E+00	0.3524E+00	0.5468E+00
0.105E+01	0.1846E+00	0.3515E+00	0.5487E+00
0.110E+01	0.1840E+00	0.3500E+00	0.5502E+00
0.115E+01	0.1835E+00	0.3480E+00	0.5513E+00
0.120E+01	0.1830E+00	0.3456E+00	0.5519E+00
0.125E+01	0.1827E+00	0.3429E+00	0.5520E+00
0.130E+01	0.1825E+00	0.3399E+00	0.5516E+00
0.135E+01	0.1825E+00	0.3368E+00	0.5507E+00
0.140E+01	0.1829E+00	0.3337E+00	0.5494E+00
0.145E+01	0.1835E+00	0.3310E+00	0.5476E+00
0.150E+01	0.1846E+00	0.3286E+00	0.5456E+00
0.155E+01	0.1860E+00	0.3270E+00	0.5434E+00
0.160E+01	0.1879E+00	0.3262E+00	0.5411E+00
0.165E+01	0.1902E+00	0.3265E+00	0.5390E+00
0.170E+01	0.1929E+00	0.3282E+00	0.5373E+00
0.175E+01	0.1961E+00	0.3312E+00	0.5361E+00
0.180E+01	0.1997E+00	0.3358E+00	0.5357E+00
0.185E+01	0.2036E+00	0.3421E+00	0.5362E+00
0.190E+01	0.2078E+00	0.3501E+00	0.5380E+00
0.195E+01	0.2122E+00	0.3598E+00	0.5411E+00
0.200E+01	0.2166E+00	0.3711E+00	0.5458E+00
0.205E+01	0.2210E+00	0.3838E+00	0.5522E+00
0.210E+01	0.2253E+00	0.3977E+00	0.5604E+00
0.215E+01	0.2293E+00	0.4126E+00	0.5704E+00
0.220E+01	0.2328E+00	0.4280E+00	0.5822E+00
0.225E+01	0.2359E+00	0.4436E+00	0.5957E+00
0.230E+01	0.2383E+00	0.4588E+00	0.6108E+00
0.235E+01	0.2400E+00	0.4732E+00	0.6271E+00
0.240E+01	0.2408E+00	0.4862E+00	0.6446E+00
0.245E+01	0.2408E+00	0.4974E+00	0.6626E+00
0.250E+01	0.2400E+00	0.5061E+00	0.6810E+00

Table 5.1

0.255E+01	0.2383E+00	0.5121E+00	0.6990E+00
0.260E+01	0.2356E+00	0.5147E+00	0.7164E+00
0.265E+01	0.2328E+00	0.5138E+00	0.7325E+00
0.270E+01	0.2292E+00	0.5092E+00	0.7467E+00
0.275E+01	0.2254E+00	0.5007E+00	0.7587E+00
0.280E+01	0.2216E+00	0.4887E+00	0.7679E+00
0.285E+01	0.2181E+00	0.4732E+00	0.7739E+00
0.290E+01	0.2153E+00	0.4550E+00	0.7763E+00
0.295E+01	0.2136E+00	0.4346E+00	0.7755E+00
0.300E+01	0.2132E+00	0.4130E+00	0.7709E+00
0.305E+01	0.2147E+00	0.3913E+00	0.7628E+00
0.310E+01	0.2183E+00	0.3707E+00	0.7517E+00
0.315E+01	0.2244E+00	0.3526E+00	0.7381E+00
0.320E+01	0.2334E+00	0.3386E+00	0.7228E+00
0.325E+01	0.2453E+00	0.3302E+00	0.7068E+00
0.330E+01	0.2604E+00	0.3289E+00	0.6913E+00
0.335E+01	0.2788E+00	0.3361E+00	0.6775E+00
0.340E+01	0.3002E+00	0.3532E+00	0.6670E+00
0.345E+01	0.3247E+00	0.3813E+00	0.6613E+00
0.350E+01	0.3518E+00	0.4212E+00	0.6620E+00
0.355E+01	0.3812E+00	0.4735E+00	0.6707E+00
0.360E+01	0.4124E+00	0.5384E+00	0.6890E+00
0.365E+01	0.4447E+00	0.6155E+00	0.7182E+00
0.370E+01	0.4774E+00	0.7043E+00	0.7595E+00
0.375E+01	0.5097E+00	0.8036E+00	0.8141E+00
0.380E+01	0.5408E+00	0.9118E+00	0.8827E+00
0.385E+01	0.5698E+00	0.1027E+01	0.9657E+00
0.390E+01	0.5959E+00	0.1146E+01	0.1063E+01
0.395E+01	0.6182E+00	0.1267E+01	0.1175E+01
0.400E+01	0.6358E+00	0.1387E+01	0.1300E+01
0.405E+01	0.6481E+00	0.1500E+01	0.1437E+01
0.410E+01	0.6545E+00	0.1605E+01	0.1584E+01
0.415E+01	0.6545E+00	0.1695E+01	0.1740E+01
0.420E+01	0.6479E+00	0.1768E+01	0.1901E+01
0.425E+01	0.6348E+00	0.1819E+01	0.2063E+01
0.430E+01	0.6156E+00	0.1844E+01	0.2223E+01
0.435E+01	0.5909E+00	0.1839E+01	0.2377E+01
0.440E+01	0.5620E+00	0.1801E+01	0.2518E+01
0.445E+01	0.5307E+00	0.1729E+01	0.2643E+01
0.450E+01	0.4992E+00	0.1620E+01	0.2745E+01
0.455E+01	0.4703E+00	0.1478E+01	0.2819E+01
0.460E+01	0.4474E+00	0.1303E+01	0.2861E+01
0.465E+01	0.4346E+00	0.1103E+01	0.2867E+01
0.470E+01	0.4362E+00	0.8849E+00	0.2835E+01
0.475E+01	0.4568E+00	0.6603E+00	0.2764E+01
0.480E+01	0.5011E+00	0.4430E+00	0.2656E+01
0.485E+01	0.5738E+00	0.2494E+00	0 [illegible]16E+01
0.490E+01	0.6790E+00	0.9772E-01	[illegible]353E+01
0.495E+01	0.8201E+00	0.7827E-02	0.2176E+01
0.500E+01	0.1000E+01	0.0000E+00	0.2000E+01

Table 5.1 The Quadratic Terms of the Optimal Cost Functional Indexed by Time for Example 2

TIME	K1(1)	K1(2)	K1(3)	K1(4)
0.000E+00	-0.5675E+00	0.1036E+00	-0.3584E-01	-0.6325E+00
0.500E-01	-0.5674E+00	0.1035E+00	-0.3593E-01	-0.6326E+00
0.100E+00	-0.5674E+00	0.1035E+00	-0.3603E-01	-0.6326E+00
0.150E+00	-0.5673E+00	0.1035E+00	-0.3614E-01	-0.6327E+00
0.200E+00	-0.5672E+00	0.1035E+00	-0.3626E-01	-0.6328E+00
0.250E+00	-0.5671E+00	0.1036E+00	-0.3639E-01	-0.6329E+00
0.300E+00	-0.5669E+00	0.1037E+00	-0.3652E-01	-0.6331E+00
0.350E+00	-0.5667E+00	0.1038E+00	-0.3665E-01	-0.6333E+00
0.400E+00	-0.5665E+00	0.1041E+00	-0.3678E-01	-0.6335E+00
0.450E+00	-0.5663E+00	0.1043E+00	-0.3690E-01	-0.6337E+00
0.500E+00	-0.5661E+00	0.1046E+00	-0.3701E-01	-0.6339E+00
0.550E+00	-0.5659E+00	0.1050E+00	-0.3711E-01	-0.6341E+00
0.600E+00	-0.5657E+00	0.1054E+00	-0.3719E-01	-0.6343E+00
0.650E+00	-0.5654E+00	0.1059E+00	-0.3725E-01	-0.6346E+00
0.700E+00	-0.5652E+00	0.1064E+00	-0.3729E-01	-0.6348E+00
0.750E+00	-0.5651E+00	0.1069E+00	-0.3731E-01	-0.6349E+00
0.800E+00	-0.5649E+00	0.1074E+00	-0.3729E-01	-0.6351E+00
0.850E+00	-0.5648E+00	0.1079E+00	-0.3726E-01	-0.6352E+00
0.900E+00	-0.5647E+00	0.1084E+00	-0.3720E-01	-0.6353E+00
0.950E+00	-0.5647E+00	0.1089E+00	-0.3712E-01	-0.6353E+00
0.100E+01	-0.5648E+00	0.1094E+00	-0.3702E-01	-0.6352E+00
0.105E+01	-0.5649E+00	0.1097E+00	-0.3691E-01	-0.6351E+00
0.110E+01	-0.5650E+00	0.1100E+00	-0.3680E-01	-0.6350E+00
0.115E+01	-0.5652E+00	0.1103E+00	-0.3669E-01	-0.6348E+00
0.120E+01	-0.5654E+00	0.1104E+00	-0.3660E-01	-0.6346E+00
0.125E+01	-0.5657E+00	0.1104E+00	-0.3653E-01	-0.6343E+00
0.130E+01	-0.5660E+00	0.1103E+00	-0.3650E-01	-0.6340E+00
0.135E+01	-0.5663E+00	0.1101E+00	-0.3651E-01	-0.6337E+00
0.140E+01	-0.5666E+00	0.1099E+00	-0.3657E-01	-0.6334E+00
0.145E+01	-0.5669E+00	0.1095E+00	-0.3670E-01	-0.6331E+00
0.150E+01	-0.5671E+00	0.1091E+00	-0.3691E-01	-0.6329E+00
0.155E+01	-0.5673E+00	0.1087E+00	-0.3720E-01	-0.6327E+00
0.160E+01	-0.5674E+00	0.1082E+00	-0.3757E-01	-0.6326E+00
0.165E+01	-0.5673E+00	0.1078E+00	-0.3804E-01	-0.6327E+00
0.170E+01	-0.5672E+00	0.1075E+00	-0.3859E-01	-0.6328E+00
0.175E+01	-0.5669E+00	0.1072E+00	-0.3923E-01	-0.6331E+00
0.180E+01	-0.5664E+00	0.1071E+00	-0.3994E-01	-0.6336E+00
0.185E+01	-0.5658E+00	0.1072E+00	-0.4072E-01	-0.6342E+00
0.190E+01	-0.5650E+00	0.1076E+00	-0.4156E-01	-0.6350E+00
0.195E+01	-0.5640E+00	0.1082E+00	-0.4243E-01	-0.6360E+00
0.200E+01	-0.5629E+00	0.1092E+00	-0.4332E-01	-0.6371E+00
0.205E+01	-0.5616E+00	0.1104E+00	-0.4421E-01	-0.6384E+00
0.210E+01	-0.5602E+00	0.1121E+00	-0.4506E-01	-0.6398E+00
0.215E+01	-0.5587E+00	0.1141E+00	-0.4585E-01	-0.6413E+00
0.220E+01	-0.5572E+00	0.1164E+00	-0.4657E-01	-0.6428E+00
0.225E+01	-0.5556E+00	0.1191E+00	-0.4718E-01	-0.6444E+00
0.230E+01	-0.5541E+00	0.1222E+00	-0.4766E-01	-0.6459E+00
0.235E+01	-0.5527E+00	0.1254E+00	-0.4799E-01	-0.6473E+00
0.240E+01	-0.5514E+00	0.1289E+00	-0.4816E-01	-0.6486E+00
0.245E+01	-0.5503E+00	0.1325E+00	-0.4817E-01	-0.6497E+00
0.250E+01	-0.5494E+00	0.1362E+00	-0.4799E-01	-0.6506E+00

Table 5.2

0.255E+01	-0.5488E+00	0.1[illegible]E+00	-0.4766E-01	-0.6512E+00
0.260E+01	-0.5485E+00	0.1433E+00	-0.4717E-01	-0.6515E+00
0.265E+01	-0.5486E+00	0.1465E+00	-0.4655E-01	-0.6514E+00
0.270E+01	-0.5491E+00	0.1493E+00	-0.4584E-01	-0.6509E+00
0.275E+01	-0.5499E+00	0.1517E+00	-0.4508E-01	-0.6501E+00
0.280E+01	-0.5511E+00	0.1536E+00	-0.4432E-01	-0.6489E+00
0.285E+01	-0.5527E+00	0.1548E+00	-0.4363E-01	-0.6473E+00
0.290E+01	-0.5545E+00	0.1553E+00	-0.4307E-01	-0.6455E+00
0.295E+01	-0.5565E+00	0.1551E+00	-0.4271E-01	-0.6435E+00
0.300E+01	-0.5587E+00	0.1542E+00	-0.4264E-01	-0.6413E+00
0.305E+01	-0.5609E+00	0.1526E+00	-0.4293E-01	-0.6391E+00
0.310E+01	-0.5629E+00	0.1503E+00	-0.4366E-01	-0.6371E+00
0.315E+01	-0.5647E+00	0.1476E+00	-0.4488E-01	-0.6353E+00
0.320E+01	-0.5661E+00	0.1446E+00	-0.4667E-01	-0.6339E+00
0.325E+01	-0.5670E+00	0.1414E+00	-0.4906E-01	-0.6330E+00
0.330E+01	-0.5671E+00	0.1383E+00	-0.5209E-01	-0.6329E+00
0.335E+01	-0.5664E+00	0.1355E+00	-0.5575E-01	-0.6336E+00
0.340E+01	-0.5647E+00	0.1334E+00	-0.6005E-01	-0.6353E+00
0.345E+01	-0.5619E+00	0.1323E+00	-0.6494E-01	-0.6381E+00
0.350E+01	-0.5579E+00	0.1324E+00	-0.7036E-01	-0.6421E+00
0.355E+01	-0.5526E+00	0.1341E+00	-0.7625E-01	-0.6474E+00
0.360E+01	-0.5462E+00	0.1378E+00	-0.8248E-01	-0.6538E+00
0.365E+01	-0.5384E+00	0.1436E+00	-0.8893E-01	-0.6616E+00
0.370E+01	-0.5296E+00	0.1519E+00	-0.9547E-01	-0.6704E+00
0.375E+01	-0.5196E+00	0.1623E+00	-0.1019E+00	-0.6804E+00
0.380E+01	-0.5088E+00	0.1765E+00	-0.1082E+00	-0.6912E+00
0.385E+01	-0.4973E+00	0.1931E+00	-0.1140E+00	-0.7027E+00
0.390E+01	-0.4854E+00	0.2126E+00	-0.1192E+00	-0.7146E+00
0.395E+01	-0.4733E+00	0.2349E+00	-0.1236E+00	-0.7267E+00
0.400E+01	-0.4613E+00	0.2599E+00	-0.1272E+00	-0.7387E+00
0.405E+01	-0.4500E+00	0.2874E+00	-0.1296E+00	-0.7500E+00
0.410E+01	-0.4395E+00	0.3169E+00	-0.1309E+00	-0.7605E+00
0.415E+01	-0.4305E+00	0.3480E+00	-0.1307E+00	-0.7695E+00
0.420E+01	-0.4232E+00	0.3801E+00	-0.1296E+00	-0.7768E+00
0.425E+01	-0.4181E+00	0.4126E+00	-0.1270E+00	-0.7819E+00
0.430E+01	-0.4156E+00	0.4446E+00	-0.1231E+00	-0.7844E+00
0.435E+01	-0.4161E+00	0.4753E+00	-0.1182E+00	-0.7839E+00
0.440E+01	-0.4199E+00	0.5036E+00	-0.1124E+00	-0.7801E+00
0.445E+01	-0.4271E+00	0.5285E+00	-0.1061E+00	-0.7729E+00
0.450E+01	-0.4380E+00	0.5490E+00	-0.9983E-01	-0.7620E+00
0.455E+01	-0.4522E+00	0.5638E+00	-0.9405E-01	-0.7478E+00
0.460E+01	-0.4697E+00	0.5723E+00	-0.8948E-01	-0.7303E+00
0.465E+01	-0.4897E+00	0.5735E+00	-0.8692E-01	-0.7103E+00
0.470E+01	-0.5115E+00	0.5670E+00	-0.8723E-01	-0.6885E+00
0.475E+01	-0.5340E+00	0.5528E+00	-0.9136E-01	-0.6660E+00
0.480E+01	-0.5557E+00	0.5312E+00	-0.1002E+00	-0.6443E+00
0.485E+01	-0.5751E+00	0.5032E+00	-0.1148E+00	-0.6249E+00
0.490E+01	-0.5902E+00	0.4705E+00	-0.1358E+00	-0.6098E+00
0.495E+01	-0.5992E+00	0.4352E+00	-0.1640E+00	-0.6008E+00
0.500E+01	-0.6000E+00	0.4000E+00	-0.2000E+00	-0.6000E+00

Table 5.2 The Linear Feedback Tensors Indexed by Time for Example 2

TIME	V3(1)	V3(2)	V3(3)	V3(4)
0.000E+00	0.5796E-02	-0.4292E+00	-0.5077E+00	-0.6928E+00
0.500E-01	0.5728E-02	-0.4295E+00	-0.5063E+00	-0.6913E+00
0.100E+00	0.5567E-02	-0.4302E+00	-0.5054E+00	-0.6897E+00
0.150E+00	0.5305E-02	-0.4313E+00	-0.5051E+00	-0.6883E+00
0.200E+00	0.4938E-02	-0.4328E+00	-0.5057E+00	-0.6870E+00
0.250E+00	0.4465E-02	-0.4348E+00	-0.5071E+00	-0.6861E+00
0.300E+00	0.3887E-02	-0.4373E+00	-0.5096E+00	-0.6857E+00
0.350E+00	0.3211E-02	-0.4402E+00	-0.5131E+00	-0.6858E+00
0.400E+00	0.2449E-02	-0.4436E+00	-0.5177E+00	-0.6866E+00
0.450E+00	0.1615E-02	-0.4473E+00	-0.5234E+00	-0.6883E+00
0.500E+00	0.7295E-03	-0.4513E+00	-0.5301E+00	-0.6907E+00
0.550E+00	-0.1848E-03	-0.4556E+00	-0.5377E+00	-0.6942E+00
0.600E+00	-0.1101E-02	-0.4600E+00	-0.5461E+00	-0.6986E+00
0.650E+00	-0.1991E-02	-0.4643E+00	-0.5550E+00	-0.7040E+00
0.700E+00	-0.2822E-02	-0.4686E+00	-0.5644E+00	-0.7103E+00
0.750E+00	-0.3564E-02	-0.4727E+00	-0.5738E+00	-0.7174E+00
0.800E+00	-0.4185E-02	-0.4763E+00	-0.5830E+00	-0.7253E+00
0.850E+00	-0.4654E-02	-0.4795E+00	-0.5917E+00	-0.7338E+00
0.900E+00	-0.4945E-02	-0.4821E+00	-0.5996E+00	-0.7427E+00
0.950E+00	-0.5035E-02	-0.4840E+00	-0.6063E+00	-0.7518E+00
0.100E+01	-0.4906E-02	-0.4850E+00	-0.6115E+00	-0.7608E+00
0.105E+01	-0.4550E-02	-0.4852E+00	-0.6150E+00	-0.7694E+00
0.110E+01	-0.3965E-02	-0.4846E+00	-0.6165E+00	-0.7774E+00
0.115E+01	-0.3160E-02	-0.4830E+00	-0.6159E+00	-0.7845E+00
0.120E+01	-0.2155E-02	-0.4807E+00	-0.6131E+00	-0.7904E+00
0.125E+01	-0.9824E-03	-0.4776E+00	-0.6080E+00	-0.7950E+00
0.130E+01	0.3130E-03	-0.4739E+00	-0.6007E+00	-0.7978E+00
0.135E+01	0.1673E-02	-0.4699E+00	-0.5915E+00	-0.7989E+00
0.140E+01	0.3028E-02	-0.4656E+00	-0.5806E+00	-0.7982E+00
0.145E+01	0.4293E-02	-0.4615E+00	-0.5684E+00	-0.7954E+00
0.150E+01	0.5378E-02	-0.4577E+00	-0.5554E+00	-0.7909E+00
0.155E+01	0.6179E-02	-0.4548E+00	-0.5423E+00	-0.7846E+00
0.160E+01	0.6590E-02	-0.4530E+00	-0.5298E+00	-0.7769E+00
0.165E+01	0.6503E-02	-0.4528E+00	-0.5187E+00	-0.7681E+00
0.170E+01	0.5810E-02	-0.4547E+00	-0.5098E+00	-0.7587E+00
0.175E+01	0.4412E-02	-0.4589E+00	-0.5042E+00	-0.7491E+00
0.180E+01	0.2220E-02	-0.4659E+00	-0.5027E+00	-0.7402E+00
0.185E+01	-0.8370E-03	-0.4762E+00	-0.5063E+00	-0.7326E+00
0.190E+01	-0.4806E-02	-0.4899E+00	-0.5158E+00	-0.7270E+00
0.195E+01	-0.9709E-02	-0.5073E+00	-0.5322E+00	-0.7245E+00
0.200E+01	-0.1553E-01	-0.5286E+00	-0.5560E+00	-0.7257E+00
0.205E+01	-0.2223E-01	-0.5538E+00	-0.5879E+00	-0.7317E+00
0.210E+01	-0.2971E-01	-0.5827E+00	-0.6281E+00	-0.7432E+00
0.215E+01	-0.3786E-01	-0.6152E+00	-0.6768E+00	-0.7611E+00
0.220E+01	-0.4649E-01	-0.6508E+00	-0.7338E+00	-0.7861E+00
0.225E+01	-0.5540E-01	-0.6890E+00	-0.7986E+00	-0.8186E+00
0.230E+01	-0.6435E-01	-0.7290E+00	-0.8703E+00	-0.8591E+00
0.235E+01	-0.7305E-01	-0.7699E+00	-0.9479E+00	-0.9078E+00
0.240E+01	-0.8122E-01	-0.8108E+00	-0.1030E+01	-0.9645E+00
0.245E+01	-0.8855E-01	-0.8504E+00	-0.1114E+01	-0.1029E+01
0.250E+01	-0.9471E-01	-0.8875E+00	-0.1198E+01	-0.1101E+01

Table 5.3

0.255E+01	-0.9941E-01	-0.9208E+00	-0.1279E+01	-0.1178E+01
0.260E+01	-0.1024E+00	-0.9490E+00	-0.1355E+01	-0.1260E+01
0.265E+01	-0.1034E+00	-0.9709E+00	-0.1423E+01	-0.1345E+01
0.270E+01	-0.1023E+00	-0.9855E+00	-0.1479E+01	-0.1431E+01
0.275E+01	-0.9898E-01	-0.9917E+00	-0.1520E+01	-0.1516E+01
0.280E+01	-0.9345E-01	-0.9897E+00	-0.1544E+01	-0.1596E+01
0.285E+01	-0.8582E-01	-0.9786E+00	-0.1548E+01	-0.1669E+01
0.290E+01	-0.7628E-01	-0.9592E+00	-0.1531E+01	-0.1733E+01
0.295E+01	-0.6516E-01	-0.9323E+00	-0.1491E+01	-0.1784E+01
0.300E+01	-0.5286E-01	-0.8993E+00	-0.1429E+01	-0.1819E+01
0.305E+01	-0.3988E-01	-0.8622E+00	-0.1346E+01	-0.1838E+01
0.310E+01	-0.2674E-01	-0.8236E+00	-0.1245E+01	-0.1838E+01
0.315E+01	-0.1402E-01	-0.7862E+00	-0.1130E+01	-0.1819E+01
0.320E+01	-0.2249E-02	-0.7531E+00	-0.1006E+01	-0.1781E+01
0.325E+01	0.8079E-02	-0.7275E+00	-0.8801E+00	-0.1725E+01
0.330E+01	0.1658E-01	-0.7122E+00	-0.7589E+00	-0.1655E+01
0.335E+01	0.2300E-01	-0.7097E+00	-0.6502E+00	-0.1572E+01
0.340E+01	0.2723E-01	-0.7218E+00	-0.5617E+00	-0.1482E+01
0.345E+01	0.2937E-01	-0.7493E+00	-0.5004E+00	-0.1390E+01
0.350E+01	0.2965E-01	-0.7922E+00	-0.4725E+00	-0.1302E+01
0.355E+01	0.2849E-01	-0.8493E+00	-0.4825E+00	-0.1223E+01
0.360E+01	0.2643E-01	-0.9182E+00	-0.5329E+00	-0.1160E+01
0.365E+01	0.2411E-01	-0.9954E+00	-0.6243E+00	-0.1117E+01
0.370E+01	0.2221E-01	-0.1077E+01	-0.7543E+00	-0.1100E+01
0.375E+01	0.2142E-01	-0.1157E+01	-0.9182E+00	-0.1111E+01
0.380E+01	0.2234E-01	-0.1230E+01	-0.1109E+01	-0.1154E+01
0.385E+01	0.2549E-01	-0.1292E+01	-0.1316E+01	-0.1229E+01
0.390E+01	0.3120E-01	-0.1335E+01	-0.1528E+01	-0.1335E+01
0.395E+01	0.3959E-01	-0.1355E+01	-0.1732E+01	-0.1470E+01
0.400E+01	0.5056E-01	-0.1349E+01	-0.1913E+01	-0.1628E+01
0.405E+01	0.6372E-01	-0.1314E+01	-0.2055E+01	-0.1802E+01
0.410E+01	0.7843E-01	-0.1249E+01	-0.2143E+01	-0.1985E+01
0.415E+01	0.9382E-01	-0.1155E+01	-0.2165E+01	-0.2165E+01
0.420E+01	0.1089E+00	-0.1038E+01	-0.2109E+01	-0.2330E+01
0.425E+01	0.1224E+00	-0.9033E+00	-0.1968E+01	-0.2468E+01
0.430E+01	0.1336E+00	-0.7611E+00	-0.1741E+01	-0.2564E+01
0.435E+01	0.1415E+00	-0.6232E+00	-0.1434E+01	-0.2607E+01
0.440E+01	0.1462E+00	-0.5025E+00	-0.1061E+01	-0.2586E+01
0.445E+01	0.1481E+00	-0.4120E+00	-0.6435E+00	-0.2494E+01
0.450E+01	0.1487E+00	-0.3621E+00	-0.2130E+00	-0.2329E+01
0.455E+01	0.1503E+00	-0.3585E+00	0.1943E+00	-0.2096E+01
0.460E+01	0.1560E+00	-0.3995E+00	0.5396E+00	-0.1808E+01
0.465E+01	0.1684E+00	-0.4742E+00	0.7881E+00	-0.1483E+01
0.470E+01	0.1890E+00	-0.5616E+00	0.9149E+00	-0.1146E+01
0.475E+01	0.2166E+00	-0.6329E+00	0.9117E+00	-0.8227E+00
0.480E+01	0.2451E+00	-0.6559E+00	0.7915E+00	-0.5385E+00
0.485E+01	0.2622E+00	-0.6031E+00	0.5884E+00	-0.3115E+00
0.490E+01	0.2475E+00	-0.4416E+00	0.3531E+00	-0.1500E+00
0.495E+01	0.1725E+00	-0.2444E+00	0.1418E+00	-0.5075E-01
0.500E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00

Table 5.3 The Third Order Tensors of the Optimal Cost Functional Indexed by Time for Example 2

TIME	K2(1)	K2(2)	K2(3)
	K2(4)	K2(5)	K2(6)
----	-----	-----	-----
0.000E+00	0.6353E-01	-0.8218E-01	-0.1864E+00
	-0.1739E-02	0.8584E-01	0.5077E-01
0.500E-01	0.6381E-01	-0.8171E-01	-0.1860E+00
	-0.1718E-02	0.8590E-01	0.5063E-01
0.100E+00	0.6408E-01	-0.8126E-01	-0.1855E+00
	-0.1670E-02	0.8604E-01	0.5054E-01
0.150E+00	0.6433E-01	-0.8086E-01	-0.1851E+00
	-0.1592E-02	0.8626E-01	0.5051E-01
0.200E+00	0.6454E-01	-0.8055E-01	-0.1847E+00
	-0.1481E-02	0.8656E-01	0.5057E-01
0.250E+00	0.6471E-01	-0.8035E-01	-0.1844E+00
	-0.1339E-02	0.8696E-01	0.5071E-01
0.300E+00	0.6482E-01	-0.8028E-01	-0.1842E+00
	-0.1166E-02	0.8746E-01	0.5096E-01
0.350E+00	0.6487E-01	-0.8038E-01	-0.1842E+00
	-0.9633E-03	0.8804E-01	0.5131E-01
0.400E+00	0.6484E-01	-0.8067E-01	-0.1843E+00
	-0.7346E-03	0.8871E-01	0.5177E-01
0.450E+00	0.6474E-01	-0.8115E-01	-0.1847E+00
	-0.4845E-03	0.8946E-01	0.5234E-01
0.500E+00	0.6455E-01	-0.8184E-01	-0.1853E+00
	-0.2189E-03	0.9026E-01	0.5301E-01
0.550E+00	0.6427E-01	-0.8275E-01	-0.1862E+00
	0.5544E-04	0.9111E-01	0.5377E-01
0.600E+00	0.6392E-01	-0.8386E-01	-0.1874E+00
	0.3304E-03	0.9199E-01	0.5461E-01
0.650E+00	0.6350E-01	-0.8518E-01	-0.1888E+00
	0.5973E-03	0.9287E-01	0.5550E-01
0.700E+00	0.6301E-01	-0.8667E-01	-0.1905E+00
	0.8467E-03	0.9373E-01	0.5644E-01
0.750E+00	0.6247E-01	-0.8831E-01	-0.1924E+00
	0.1069E-02	0.9453E-01	0.5738E-01
0.800E+00	0.6190E-01	-0.9006E-01	-0.1945E+00
	0.1255E-02	0.9527E-01	0.5830E-01
0.850E+00	0.6132E-01	-0.9187E-01	-0.1969E+00
	0.1396E-02	0.9590E-01	0.5917E-01
0.900E+00	0.6076E-01	-0.9370E-01	-0.1993E+00
	0.1483E-02	0.9642E-01	0.5996E-01
0.950E+00	0.6023E-01	-0.9548E-01	-0.2018E+00
	0.1510E-02	0.9679E-01	0.6063E-01
0.100E+01	0.5978E-01	-0.9716E-01	-0.2043E+00
	0.1472E-02	0.9700E-01	0.6115E-01
0.105E+01	0.5942E-01	-0.9867E-01	-0.2067E+00
	0.1365E-02	0.9705E-01	0.6150E-01
0.110E+01	0.5920E-01	-0.9995E-01	-0.2090E+00
	0.1190E-02	0.9691E-01	0.6165E-01
0.115E+01	0.5913E-01	-0.1009E+00	-0.2110E+00
	0.9480E-03	0.9661E-01	0.6159E-01
0.120E+01	0.5925E-01	-0.1016E+00	-0.2128E+00
	0.6465E-03	0.9614E-01	0.6131E-01

Table 5.4

0.125E+01	0.5957E-01	-0.1018E+00	-0.2141E+00
	0.2947E-03	0.9552E-01	0.6080E-01
0.130E+01	0.6012E-01	-0.1016E+00	-0.2150E+00
	-0.9[illegible]E-04	0.9479E-01	0.6007E-01
0.135E+01	0.6[illegible]E-01	-0.1009E+00	-0.2154E+00
	-0.5020E-03	0.9397E-01	0.5915E-01
0.140E+01	0.6191E-01	-0.9980E-01	-0.2153E+00
	-0.9083E-03	0.9312E-01	0.5806E-01
0.145E+01	0.6315E-01	-0.9822E-01	-0.2146E+00
	-0.1288E-02	0.9229E-01	0.5684E-01
0.150E+01	0.6459E-01	-0.9622E-01	-0.2135E+00
	-0.1613E-02	0.9155E-01	0.5554E-01
0.155E+01	0.6621E-01	-0.9386E-01	-0.2118E+00
	-0.1854E-02	0.9096E-01	0.5423E-01
0.160E+01	0.6797E-01	-0.9121E-01	-0.2097E+00
	-0.1977E-02	0.9061E-01	0.5298E-01
0.165E+01	0.6981E-01	-0.8840E-01	-0.2072E+00
	-0.1951E-02	0.9057E-01	0.5187E-01
0.170E+01	0.7166E-01	-0.8555E-01	-0.2045E+00
	-0.1743E-02	0.9093E-01	0.5098E-01
0.175E+01	0.7346E-01	-0.8281E-01	-0.2018E+00
	-0.1324E-02	0.9178E-01	0.5042E-01
0.180E+01	0.7513E-01	-0.8038E-01	-0.1991E+00
	-0.6659E-03	0.9319E-01	0.5027E-01
0.185E+01	0.7657E-01	-0.7842E-01	-0.1968E+00
	0.2511E-03	0.9523E-01	0.5063E-01
0.190E+01	0.7769E-01	-0.7717E-01	-0.1950E+00
	0.1442E-02	0.9798E-01	0.5158E-01
0.195E+01	0.7842E-01	-0.7681E-01	-0.1939E+00
	0.2913E-02	0.1015E+00	0.5322E-01
0.200E+01	0.7866E-01	-0.7757E-01	-0.1939E+00
	0.4660E-02	0.1057E+00	0.5560E-01
0.205E+01	0.7834E-01	-0.7964E-01	-0.1951E+00
	0.6669E-02	0.1108E+00	0.5879E-01
0.210E+01	0.7740E-01	-0.8320E-01	-0.1978E+00
	0.8914E-02	0.1165E+00	0.6281E-01
0.215E+01	0.7580E-01	-0.8840E-01	-0.2023E+00
	0.1136E-01	0.1230E+00	0.6768E-01
0.220E+01	0.7350E-01	-0.9535E-01	-0.2087E+00
	0.1395E-01	0.1302E+00	0.7338E-01
0.225E+01	0.7053E-01	-0.1041E+00	-0.2172E+00
	0.1662E-01	0.1378E+00	0.7986E-01
0.230E+01	0.6690E-01	-0.1147E+00	-0.2279E+00
	0.1930E-01	0.1458E+00	0.8703E-01
0.235E+01	0.6268E-01	-0.1271E+00	-0.2409E+00
	0.2192E-01	0.1540E+00	0.9479E-01
0.240E+01	0.5796E-01	-0.1411E+00	-0.2561E+00
	0.2437E-01	0.1622E+00	0.1030E+00
0.245E+01	0.5289E-01	-0.1565E+00	-0.2736E+00
	0.2656E-01	0.1701E+00	0.1114E+00
0.250E+01	0.4762E-01	-0.1730E+00	-0.2931E+00
	0.2841E-01	0.1775E+00	0.1198E+00
0.255E+01	0.4235E-01	-0.1902E+00	-0.3143E+00
	0.2982E-01	0.1842E+00	0.1279E+00

Table 5.4

0.260E+01	0.3731E-01	-0.2077E+00	-0.3370E+00
	0.3072E-01	0.1898E+00	0.1355E+00
0.265E+01	0.3273E-01	-0.2248E+00	-0.3607E+00
	0.3102E-01	0.1942E+00	0.1423E+00
0.270E+01	0.2890E-01	-0.2409E+00	-0.3848E+00
	0.3069E-01	0.1971E+00	0.1479E+00
0.275E+01	0.2605E-01	-0.2555E+00	-0.4087E+00
	0.2969E-01	0.1984E+00	0.1520E+00
0.280E+01	0.2446E-01	-0.2676E+00	-0.4317E+00
	0.2804E-01	0.1979E+00	0.1544E+00
0.285E+01	0.2434E-01	-0.2768E+00	-0.4529E+00
	0.2574E-01	0.1957E+00	0.1548E+00
0.290E+01	0.2589E-01	-0.2822E+00	-0.4716E+00
	0.2288E-01	0.1918E+00	0.1531E+00
0.295E+01	0.2926E-01	-0.2835E+00	-0.4870E+00
	0.1955E-01	0.1865E+00	0.1491E+00
0.300E+01	0.3450E-01	-0.2801E+00	-0.4983E+00
	0.1586E-01	0.1799E+00	0.1429E+00
0.305E+01	0.4162E-01	-0.2719E+00	-0.5049E+00
	0.1196E-01	0.1724E+00	0.1346E+00
0.310E+01	0.5054E-01	-0.2588E+00	-0.5062E+00
	0.8022E-02	0.1647E+00	0.1245E+00
0.315E+01	0.6109E-01	-0.2413E+00	-0.5021E+00
	0.4205E-02	0.1572E+00	0.1130E+00
0.320E+01	0.7303E-01	-0.2197E+00	-0.4925E+00
	0.6748E-03	0.1506E+00	0.1006E+00
0.325E+01	0.8605E-01	-0.1949E+00	-0.4776E+00
	-0.2424E-02	0.1455E+00	0.8801E-01
0.330E+01	0.9982E-01	-0.1680E+00	-0.4581E+00
	-0.4974E-02	0.1424E+00	0.7589E-01
0.335E+01	0.1140E+00	-0.1400E+00	-0.4349E+00
	-0.6899E-02	0.1419E+00	0.6502E-01
0.340E+01	0.1281E+00	-0.1123E+00	-0.4091E+00
	-0.8170E-02	0.1444E+00	0.5617E-01
0.345E+01	0.1420E+00	-0.8611E-01	-0.3821E+00
	-0.8811E-02	0.1499E+00	0.5004E-01
0.350E+01	0.1552E+00	-0.6258E-01	-0.3556E+00
	-0.8896E-02	0.1584E+00	0.4725E-01
0.355E+01	0.1677E+00	-0.4265E-01	-0.3310E+00
	-0.8548E-02	0.1699E+00	0.4825E-01
0.360E+01	0.1793E+00	-0.2692E-01	-0.3099E+00
	-0.7930E-02	0.1836E+00	0.5329E-01
0.365E+01	0.1899E+00	-0.1564E-01	-0.2938E+00
	-0.7233E-02	0.1991E+00	0.6243E-01
0.370E+01	0.1996E+00	-0.8627E-02	-0.2837E+00
	-0.6664E-02	0.2153E+00	0.7543E-01
0.375E+01	0.2086E+00	-0.5251E-02	-0.2803E+00
	-0.6425E-02	0.2314E+00	0.9182E-01
0.380E+01	0.2168E+00	-0.4465E-02	-0.2839E+00
	-0.6702E-02	0.2461E+00	0.1109E+00
0.385E+01	0.2246E+00	-0.4858E-02	-0.2942E+00
	-0.7647E-02	0.2583E+00	0.1316E+00
0.390E+01	0.2320E+00	-0.4743E-02	-0.3102E+00
	-0.9359E-02	0.2670E+00	0.1528E+00

Table 5.4

0.395E+01	0.2390E+00	-0.2282E-02	-0.3306E+00
	-0.1188E-01	0.2711E+00	0.1732E+00
0.400E+01	0.2458E+00	0.4359E-02	-0.3532E+00
	-0.1517E-01	0.2698E+00	0.1913E+00
0.405E+01	0.2521E+00	0.1683E-01	-0.3755E+00
	-0.1912E-01	0.2628E+00	0.2055E+00
0.410E+01	0.2577E+00	0.3637E-01	-0.3947E+00
	-0.2353E-01	0.2497E+00	0.2143E+00
0.415E+01	0.2621E+00	0.6363E-01	-0.4073E+00
	-0.2815E-01	0.2311E+00	0.2165E+00
0.420E+01	0.2649E+00	0.9837E-01	-0.4100E+00
	-0.3266E-01	0.2076E+00	0.2109E+00
0.425E+01	0.2654E+00	0.1393E+00	-0.3998E+00
	-0.3673E-01	0.1807E+00	0.1968E+00
0.430E+01	0.2631E+00	0.1838E+00	-0.3738E+00
	-0.4007E-01	0.1522E+00	0.1741E+00
0.435E+01	0.2573E+00	0.2281E+00	-0.3302E+00
	-0.4246E-01	0.1246E+00	0.1434E+00
0.440E+01	0.2481E+00	0.2669E+00	-0.2684E+00
	-0.4385E-01	0.1005E+00	0.1061E+00
0.445E+01	0.2357E+00	0.2940E+00	-0.1894E+00
	-0.4442E-01	0.8239E-01	0.6435E-01
0.450E+01	0.2212E+00	0.3026E+00	-0.9599E-01
	-0.4460E-01	0.7241E-01	0.2130E-01
0.455E+01	0.2067E+00	0.2865E+00	0.6892E-02
	-0.4510E-01	0.7170E-01	-0.1943E-01
0.460E+01	0.1956E+00	0.2408E+00	0.1125E+00
	-0.4680E-01	0.7991E-01	-0.5396E-01
0.465E+01	0.1922E+00	0.1636E+00	0.2127E+00
	-0.5051E-01	0.9484E-01	-0.7881E-01
0.470E+01	0.2022E+00	0.5726E-01	0.2992E+00
	-0.5670E-01	0.1123E+00	-0.9149E-01
0.475E+01	0.2316E+00	-0.7083E-01	0.3643E+00
	-0.6497E-01	0.1266E+00	-0.9117E-01
0.480E+01	0.2861E+00	-0.2078E+00	0.4028E+00
	-0.7353E-01	0.1312E+00	-0.7915E-01
0.485E+01	0.3701E+00	-0.3363E+00	0.4131E+00
	-0.7865E-01	0.1206E+00	-0.5884E-01
0.490E+01	0.4855E+00	-0.4365E+00	0.3978E+00
	-0.7426E-01	0.9233E-01	-0.3531E-01
0.495E+01	0.6307E+00	-0.4894E+00	0.3635E+00
	-0.5175E-01	0.4889E-01	-0.1418E-01
0.500E+01	0.8000E+00	0.0000E+00	0.0000E+00
	0.0000E+00	0.0000E+00	0.0000E+00

Table 5.4 The Quadratic Feedback Tensors Indexed by Time for Example 2

TIME	V4(1)	V4(2)	V4(3)	V4(4)	V4(5)
0.000E+00	-0.8297E-02	0.2551E+00	0.9511E+00	0.1005E+01	0.1051E+01
0.500E-01	-0.8716E-02	0.2562E+00	0.9534E+00	0.1009E+01	0.1053E+01
0.100E+00	-0.9288E-02	0.2570E+00	0.9555E+00	0.1013E+01	0.1055E+01
0.150E+00	-0.1002E-01	0.2574E+00	0.9573E+00	0.1016E+01	0.1058E+01
0.200E+00	-0.1090E-01	0.2574E+00	0.9586E+00	0.1019E+01	0.1060E+01
0.250E+00	-0.1193E-01	0.2569E+00	0.9593E+00	0.1022E+01	0.1063E+01
0.300E+00	-0.1310E-01	0.2558E+00	0.9593E+00	0.1024E+01	0.1066E+01
0.350E+00	-0.1438E-01	0.2541E+00	0.9586E+00	0.1025E+01	0.1068E+01
0.400E+00	-0.1575E-01	0.2518E+00	0.9568E+00	0.1024E+01	0.1070E+01
0.450E+00	-0.1718E-01	0.2489E+00	0.9541E+00	0.1023E+01	0.1072E+01
0.500E+00	-0.1863E-01	0.2453E+00	0.9503E+00	0.1020E+01	0.1074E+01
0.550E+00	-0.2005E-01	0.2412E+00	0.9454E+00	0.1015E+01	0.1074E+01
0.600E+00	-0.2142E-01	0.2366E+00	0.9394E+00	0.1009E+01	0.1074E+01
0.650E+00	-0.2267E-01	0.2315E+00	0.9323E+00	0.1001E+01	0.1073E+01
0.700E+00	-0.2377E-01	0.2262E+00	0.9243E+00	0.9920E+00	0.1071E+01
0.750E+00	-0.2465E-01	0.2208E+00	0.9156E+00	0.9810E+00	0.1068E+01
0.800E+00	-0.2529E-01	0.2154E+00	0.9062E+00	0.9687E+00	0.1063E+01
0.850E+00	-0.2564E-01	0.2102E+00	0.8965E+00	0.9552E+00	0.1058E+01
0.900E+00	-0.2566E-01	0.2055E+00	0.8868E+00	0.9409E+00	0.1051E+01
0.950E+00	-0.2534E-01	0.2016E+00	0.8774E+00	0.9263E+00	0.1044E+01
0.100E+01	-0.2466E-01	0.1986E+00	0.8687E+00	0.9118E+00	0.1035E+01
0.105E+01	-0.2363E-01	0.1967E+00	0.8612E+00	0.8980E+00	0.1026E+01
0.110E+01	-0.2227E-01	0.1962E+00	0.8552E+00	0.8855E+00	0.1016E+01
0.115E+01	-0.2063E-01	0.1973E+00	0.8512E+00	0.8750E+00	0.1006E+01
0.120E+01	-0.1875E-01	0.2001E+00	0.8496E+00	0.8670E+00	0.9965E+00
0.125E+01	-0.1673E-01	0.2046E+00	0.8507E+00	0.8624E+00	0.9875E+00
0.130E+01	-0.1466E-01	0.2109E+00	0.8547E+00	0.8616E+00	0.9794E+00
0.135E+01	-0.1266E-01	0.2188E+00	0.8619E+00	0.8652E+00	0.9729E+00
0.140E+01	-0.1087E-01	0.2282E+00	0.8721E+00	0.8735E+00	0.9683E+00
0.145E+01	-0.9408E-02	0.2388E+00	0.8853E+00	0.8867E+00	0.9661E+00
0.150E+01	-0.8423E-02	0.2503E+00	0.9010E+00	0.9048E+00	0.9666E+00
0.155E+01	-0.8045E-02	0.2622E+00	0.9189E+00	0.9274E+00	0.9703E+00
0.160E+01	-0.8387E-02	0.2741E+00	0.9382E+00	0.9542E+00	0.9771E+00
0.165E+01	-0.9540E-02	0.2856E+00	0.9584E+00	0.9842E+00	0.9874E+00
0.170E+01	-0.1156E-01	0.2962E+00	0.9785E+00	0.1017E+01	0.1001E+01
0.175E+01	-0.1448E-01	0.3054E+00	0.9979E+00	0.1050E+01	0.1017E+01
0.180E+01	-0.1826E-01	0.3131E+00	0.1016E+01	0.1083E+01	0.1037E+01
0.185E+01	-0.2283E-01	0.3189E+00	0.1032E+01	0.1116E+01	0.1058E+01
0.190E+01	-0.2809E-01	0.3229E+00	0.1046E+01	0.1145E+01	0.1081E+01
0.195E+01	-0.3387E-01	0.3250E+00	0.1058E+01	0.1171E+01	0.1106E+01
0.200E+01	-0.3996E-01	0.3256E+00	0.1069E+01	0.1194E+01	0.1130E+01
0.205E+01	-0.4614E-01	0.3248E+00	0.1078E+01	0.1212E+01	0.1155E+01
0.210E+01	-0.5216E-01	0.3233E+00	0.1087E+01	0.1226E+01	0.1179E+01
0.215E+01	-0.5773E-01	0.3214E+00	0.1097E+01	0.1237E+01	0.1202E+01
0.220E+01	-0.6260E-01	0.3199E+00	0.1110E+01	0.1247E+01	0.1225E+01
0.225E+01	-0.6650E-01	0.3192E+00	0.1126E+01	0.1256E+01	0.1247E+01
0.230E+01	-0.6918E-01	0.3200E+00	0.1149E+01	0.1269E+01	0.1269E+01
0.235E+01	-0.7041E-01	0.3227E+00	0.1177E+01	0.1285E+01	0.1292E+01
0.240E+01	-0.7002E-01	0.3280E+00	0.1214E+01	0.1310E+01	0.1318E+01
0.245E+01	-0.6786E-01	0.3361E+00	0.1259E+01	0.1344E+01	0.1347E+01
0.250E+01	-0.6386E-01	0.3475E+00	0.1313E+01	0.1390E+01	0.1382E+01

Table 5.5

0.255E+01	-0.5800E-01	0.3623E+00	0.1376E+01	0.1449E+01	0.1423E+01
0.260E+01	-0.5035E-01	0.3805E+00	0.1446E+01	0.1523E+01	0.1474E+01
0.265E+01	-0.4103E-01	0.4022E+00	0.1523E+01	0.1611E+01	0.1534E+01
0.270E+01	-0.3030E-01	0.4272E+00	0.1605E+01	0.1712E+01	0.1606E+01
0.275E+01	-0.1847E-01	0.4551E+00	0.1690E+01	0.1825E+01	0.1688E+01
0.280E+01	-0.5955E-02	0.4853E+00	0.1776E+01	0.1946E+01	0.1782E+01
0.285E+01	0.6752E-02	0.5173E+00	0.1860E+01	0.2070E+01	0.1886E+01
0.290E+01	0.1911E-01	0.5501E+00	0.1939E+01	0.2193E+01	0.1999E+01
0.295E+01	0.3053E-01	0.5829E+00	0.2011E+01	0.2308E+01	0.2116E+01
0.300E+01	0.4045E-01	0.6147E+00	0.2074E+01	0.2412E+01	0.2236E+01
0.305E+01	0.4836E-01	0.6446E+00	0.2127E+01	0.2497E+01	0.2353E+01
0.310E+01	0.5380E-01	0.6719E+00	0.2169E+01	0.2560E+01	0.2464E+01
0.315E+01	0.5648E-01	0.6961E+00	0.2201E+01	0.2598E+01	0.2563E+01
0.320E+01	0.5621E-01	0.7168E+00	0.2224E+01	0.2612E+01	0.2648E+01
0.325E+01	0.5298E-01	0.7343E+00	0.2243E+01	0.2601E+01	0.2715E+01
0.330E+01	0.4690E-01	0.7487E+00	0.2260E+01	0.2572E+01	0.2762E+01
0.335E+01	0.3824E-01	0.7605E+00	0.2281E+01	0.2531E+01	0.2791E+01
0.340E+01	0.2732E-01	0.7700E+00	0.2309E+01	0.2488E+01	0.2802E+01
0.345E+01	0.1452E-01	0.7773E+00	0.2349E+01	0.2451E+01	0.2799E+01
0.350E+01	0.2349E-03	0.7820E+00	0.2403E+01	0.2431E+01	0.2789E+01
0.355E+01	-0.1518E-01	0.7832E+00	0.2471E+01	0.2437E+01	0.2776E+01
0.360E+01	-0.3140E-01	0.7794E+00	0.2550E+01	0.2474E+01	0.2769E+01
0.365E+01	-0.4813E-01	0.7685E+00	0.2634E+01	0.2546E+01	0.2774E+01
0.370E+01	-0.6511E-01	0.7481E+00	0.2715E+01	0.2649E+01	0.2797E+01
0.375E+01	-0.8209E-01	0.7155E+00	0.2782E+01	0.2778E+01	0.2843E+01
0.380E+01	-0.9876E-01	0.6686E+00	0.2821E+01	0.2919E+01	0.2914E+01
0.385E+01	-0.1148E+00	0.6056E+00	0.2820E+01	0.3054E+01	0.3010E+01
0.390E+01	-0.1296E+00	0.5263E+00	0.2766E+01	0.3163E+01	0.3127E+01
0.395E+01	-0.1427E+00	0.4317E+00	0.2647E+01	0.3220E+01	0.3258E+01
0.400E+01	-0.1534E+00	0.3250E+00	0.2459E+01	0.3199E+01	0.3392E+01
0.405E+01	-0.1610E+00	0.2114E+00	0.2202E+01	0.3075E+01	0.3516E+01
0.410E+01	-0.1648E+00	0.9773E-01	0.1883E+01	0.2829E+01	0.3610E+01
0.415E+01	-0.1645E+00	-0.7737E-02	0.1521E+01	0.2450E+01	0.3656E+01
0.420E+01	-0.1599E+00	-0.9663E-01	0.1142E+01	0.1940E+01	0.3632E+01
0.425E+01	-0.1512E+00	-0.1619E+00	0.7812E+00	0.1319E+01	0.3521E+01
0.430E+01	-0.1392E+00	-0.1994E+00	0.4744E+00	0.6276E+00	0.3310E+01
0.435E+01	-0.1250E+00	-0.2096E+00	0.2553E+00	-0.7521E-01	0.2995E+01
0.440E+01	-0.1101E+00	-0.1979E+00	0.1451E+00	-0.7160E+00	0.2586E+01
0.445E+01	-0.9572E-01	-0.1739E+00	0.1445E+00	-0.1218E+01	0.2106E+01
0.450E+01	-0.8288E-01	-0.1486E+00	0.2284E+00	-0.1517E+01	0.1593E+01
0.455E+01	-0.7216E-01	-0.1296E+00	0.3473E+00	-0.1579E+01	0.1093E+01
0.460E+01	-0.6404E-01	-0.1168E+00	0.4384E+00	-0.1414E+01	0.6540E+00
0.465E+01	-0.5971E-01	-0.9937E-01	0.4453E+00	-0.1081E+01	0.3136E+00
0.470E+01	-0.6179E-01	-0.5994E-01	0.3432E+00	-0.6780E+00	0.8989E-01
0.475E+01	-0.7431E-01	0.1498E-01	0.1553E+00	-0.3115E+00	-0.2321E-01
0.480E+01	-0.1000E+00	0.1188E+00	-0.4819E-01	-0.6209E-01	-0.5342E-01
0.485E+01	-0.1333E+00	0.2116E+00	-0.1789E+00	0.4461E-01	-0.3925E-01
0.490E+01	-0.1476E+00	0.2243E+00	-0.1784E+00	0.4411E-01	-0.1554E-01
0.495E+01	-0.7746E-01	0.9342E-01	-0.6108E-01	0.8386E-02	-0.1852E-02
0.500E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00

Table 5.5 The Fourth Order Tensors of the Optimal Cost Functional Indexed by Time for Example 2

ORIGINAL PAGE IS
OF POOR QUALITY

TIME	K3(1)	K3(2)	K3(3)	K3(4)
	K3(5)	K3(6)	K3(7)	K3(8)
0.000E+00	0.2083E-01	-0.2942E-01	0.3455E+00	0.3405E+00
	0.2178E-01	-0.2114E-01	-0.2009E+00	-0.1005E+00
0.500E-01	0.2074E-01	-0.2950E-01	0.3468E+00	0.3415E+00
	0.2197E-01	-0.2149E-01	-0.2014E+00	-0.1009E+00
0.100E+00	0.2053E-01	-0.2984E-01	0.3479E+00	0.3426E+00
	0.2222E-01	-0.2175E-01	-0.2018E+00	-0.1013E+00
0.150E+00	0.2020E-01	-0.3048E-01	0.3486E+00	0.3438E+00
	0.2256E-01	-0.2190E-01	-0.2022E+00	-0.1016E+00
0.200E+00	0.1975E-01	-0.3143E-01	0.3490E+00	0.3450E+00
	0.2297E-01	-0.2190E-01	-0.2024E+00	-0.1019E+00
0.250E+00	0.1917E-01	-0.3271E-01	0.3488E+00	0.3461E+00
	0.2345E-01	-0.2173E-01	-0.2026E+00	-0.1022E+00
0.300E+00	0.1846E-01	-0.3434E-01	0.3481E+00	0.3471E+00
	0.2399E-01	-0.2138E-01	-0.2026E+00	-0.1024E+00
0.350E+00	0.1764E-01	-0.3630E-01	0.3469E+00	0.3480E+00
	0.2460E-01	-0.2083E-01	-0.2025E+00	-0.1025E+00
0.400E+00	0.1671E-01	-0.3860E-01	0.3449E+00	0.3486E+00
	0.2525E-01	-0.2007E-01	-0.2022E+00	-0.1024E+00
0.450E+00	0.1569E-01	-0.4120E-01	0.3424E+00	0.3490E+00
	0.2594E-01	-0.1909E-01	-0.2017E+00	-0.1023E+00
0.500E+00	0.1461E-01	-0.4406E-01	0.3392E+00	0.3490E+00
	0.2663E-01	-0.1791E-01	-0.2010E+00	-0.1020E+00
0.550E+00	0.1349E-01	-0.4714E-01	0.3353E+00	0.3486E+00
	0.2733E-01	-0.1652E-01	-0.2001E+00	-0.1015E+00
0.600E+00	0.1237E-01	-0.5038E-01	0.3309E+00	0.3477E+00
	0.2799E-01	-0.1496E-01	-0.1990E+00	-0.1009E+00
0.650E+00	0.1127E-01	-0.5370E-01	0.3260E+00	0.3463E+00
	0.2861E-01	-0.1325E-01	-0.1977E+00	-0.1001E+00
0.700E+00	0.1023E-01	-0.5702E-01	0.3208E+00	0.3443E+00
	0.2915E-01	-0.1144E-01	-0.1962E+00	-0.9920E-01
0.750E+00	0.9299E-02	-0.6025E-01	0.3152E+00	0.3418E+00
	0.2960E-01	-0.9573E-02	-0.1945E+00	-0.9810E-01
0.800E+00	0.8509E-02	-0.6330E-01	0.3095E+00	0.3387E+00
	0.2994E-01	-0.7706E-02	-0.1928E+00	-0.9687E-01
0.850E+00	0.7900E-02	-0.6606E-01	0.3039E+00	0.3350E+00
	0.3013E-01	-0.5907E-02	-0.1910E+00	-0.9552E-01
0.900E+00	0.7508E-02	-0.6845E-01	0.2986E+00	0.3309E+00
	0.3018E-01	-0.4248E-02	-0.1891E+00	-0.9409E-01
0.950E+00	0.7365E-02	-0.7036E-01	0.2938E+00	0.3263E+00
	0.3006E-01	-0.2807E-02	-0.1873E+00	-0.9263E-01
0.100E+01	0.7497E-02	-0.7170E-01	0.2896E+00	0.3214E+00
	0.2977E-01	-0.1660E-02	-0.1857E+00	-0.9118E-01
0.105E+01	0.7921E-02	-0.7239E-01	0.2863E+00	0.3163E+00
	0.2931E-01	-0.8849E-03	-0.1843E+00	-0.8980E-01
0.110E+01	0.8642E-02	-0.7238E-01	0.2842E+00	0.3111E+00
	0.2868E-01	-0.5513E-03	-0.1831E+00	-0.8855E-01
0.115E+01	0.9655E-02	-0.7162E-01	0.2834E+00	0.3061E+00
	0.2792E-01	-0.7190E-03	-0.1824E+00	-0.8750E-01
0.120E+01	0.1094E-01	-0.7010E-01	0.2841E+00	0.3013E+00
	0.2704E-01	-0.1433E-02	-0.1821E+00	-0.8670E-01

Table 5.6

ORIGINAL PAGE IS
OF POOR QUALITY

0.125E+01	0.1245E-01	-0.4784E-01	0.2864E+00	0.2971E+00
	0.2609E-01	-0.2718E-02	-0.1823E+00	-0.8624E-01
0.130E+01	0.1415E-01	-0.6491E-01	0.2905E+00	0.2935E+00
	0.2510E-01	-0.4576E-02	-0.1831E+00	-0.8616E-01
0.135E+01	0.1594E-01	-0.6141E-01	0.2963E+00	0.2909E+00
	0.2414E-01	-0.6980E-02	-0.1845E+00	-0.8652E-01
0.140E+01	0.1775E-01	-0.5750E-01	0.3037E+00	0.2894E+00
	0.2326E-01	-0.9878E-02	-0.1865E+00	-0.8735E-01
0.145E+01	0.1947E-01	-0.5338E-01	0.3126E+00	0.2891E+00
	0.2253E-01	-0.1319E-01	-0.1890E+00	-0.8867E-01
0.150E+01	0.2098E-01	-0.4929E-01	0.3228E+00	0.2902E+00
	0.2201E-01	-0.1679E-01	-0.1921E+00	-0.9048E-01
0.155E+01	0.2216E-01	-0.4553E-01	0.3340E+00	0.2928E+00
	0.2177E-01	-0.2057E-01	-0.1956E+00	-0.9274E-01
0.160E+01	0.2287E-01	-0.4241E-01	0.3457E+00	0.2969E+00
	0.2186E-01	-0.2437E-01	-0.1994E+00	-0.9542E-01
0.165E+01	0.2300E-01	-0.4027E-01	0.3574E+00	0.3025E+00
	0.2234E-01	-0.2804E-01	-0.2033E+00	-0.9842E-01
0.170E+01	0.2242E-01	-0.3944E-01	0.3685E+00	0.3093E+00
	0.2324E-01	-0.3143E-01	-0.2073E+00	-0.1017E+00
0.175E+01	0.2105E-01	-0.4023E-01	0.3785E+00	0.3174E+00
	0.2457E-01	-0.3440E-01	-0.2111E+00	-0.1050E+00
0.180E+01	0.1881E-01	-0.4291E-01	0.3867E+00	0.3263E+00
	0.2632E-01	-0.3684E-[illegible]	-0.2147E+00	-0.1083E+00
0.185E+01	0.1566E-01	-0.4770E-[illegible]	0.3926E+00	0.3358E+00
	0.2849E-01	-0.3867E-01	-0.2179E+00	-0.1116E+00
0.190E+01	0.1161E-01	-0.5469E-01	0.3956E+00	0.3456E+00
	0.3102E-01	-0.3984E-01	-0.2208E+00	-0.1145E+00
0.195E+01	0.6682E-02	-0.6393E-01	0.3953E+00	0.3552E+00
	0.3384E-01	-0.4036E-01	-0.2234E+00	-0.1171E+00
0.200E+01	0.9611E-03	-0.7531E-01	0.3915E+00	0.3642E+00
	0.3687E-01	-0.4027E-01	-0.2256E+00	-0.1194E+00
0.205E+01	-0.5433E-02	-0.8862E-01	0.3840E+00	0.3724E+00
	0.4001E-01	-0.3966E-01	-0.2277E+00	-0.1212E+00
0.210E+01	-0.1234E-01	-0.1035E+00	0.3730E+00	0.3794E+00
	0.4314E-01	-0.3865E-01	-0.2299E+00	-0.1226E+00
0.215E+01	-0.1956E-01	-0.1196E+00	0.3588E+00	0.3849E+00
	0.4615E-01	-0.3740E-01	-0.2324E+00	-0.1237E+00
0.220E+01	-0.2685E-01	-0.1362E+00	0.3421E+00	0.3887E+00
	0.4889E-01	-0.3606E-01	-0.2355E+00	-0.1247E+00
0.225E+01	-0.3396E-01	-0.1528E+00	0.3235E+00	0.3910E+00
	0.5125E-01	-0.3484E-01	-0.2395E+00	-0.1256E+00
0.230E+01	-0.4061E-01	-0.1685E+00	0.3041E+00	0.3916E+00
	0.5309E-01	-0.3390E-01	-0.2446E+00	-0.1269E+00
0.235E+01	-0.4651E-01	-0.1828E+00	0.2852E+00	0.3911E+00
	0.5432E-01	-0.3343E-01	-0.2512E+00	-0.1235E+00
0.240E+01	-0.5134E-01	-0.1946E+00	0.2682E+00	0.3896E+00
	0.5482E-01	-0.3359E-01	-0.2594E+00	-0.1310E+00
0.245E+01	-0.5483E-01	-0.2033E+00	0.2544E+00	0.3879E+00
	0.5451E-01	-0.3451E-01	-0.2694E+00	-0.1344E+00
0.250E+01	-0.5670E-01	-0.2081E+00	0.2455E+00	0.3867E+00
	0.5335E-01	-0.3632E-01	-0.2812E+00	-0.1390E+00
0.255E+01	-0.5671E-01	-0.2084E+00	0.2430E+00	0.3867E+00
	0.5130E-01	-0.3911E-01	-0.2947E+00	-0.1449E+00

Table 5.6

0.260E+01	-0.5469E-01	-0.2036E+00	0.2482E+00	0.3890E+00
	0.4837E-01	-0.4294E-01	-0.3097E+00	-0.1523E+00
0.265E+01	-0.5053E-01	-0.1933E+00	0.2625E+00	0.3944E+00
	0.4460E-01	-0.4782E-01	-0.3261E+00	-0.1611E+00
0.270E+01	-0.4419E-01	-0.1774E+00	0.2868E+00	0.4040E+00
	0.4008E-01	-0.5375E-01	-0.3433E+00	-0.1712E+00
0.275E+01	-0.3575E-01	-0.1560E+00	0.3216E+00	0.4185E+00
	0.3492E-01	-0.6067E-01	-0.3611E+00	-0.1825E+00
0.280E+01	-0.2541E-01	-0.1296E+00	0.3670E+00	0.4387E+00
	0.2931E-01	-0.6845E-01	-0.3788E+00	-0.1946E+00
0.285E+01	-0.1348E-01	-0.9916E-01	0.4223E+00	0.4649E+00
	0.2345E-01	-0.7696E-01	-0.3959E+00	-0.2070E+00
0.290E+01	-0.3753E-03	-0.6578E-01	0.4864E+00	0.4971E+00
	0.1759E-01	-0.8599E-01	-0.4119E+00	-0.2193E+00
0.295E+01	0.1337E-01	-0.3109E-01	0.5572E+00	0.5351E+00
	0.1198E-01	-0.9530E-01	-0.4262E+00	-0.2308E+00
0.300E+01	0.2717E-01	0.3044E-02	0.6322E+00	0.5779E+00
	0.6893E-02	-0.1046E+00	-0.4386E+00	-0.2412E+00
0.305E+01	0.4041E-01	0.3459E-01	0.7082E+00	0.6242E+00
	0.2602E-02	-0.1138E+00	-0.4486E+00	-0.2497E+00
0.310E+01	0.5250E-01	0.6150E-01	0.7818E+00	0.6725E+00
	-0.6549E-03	-0.1225E+00	-0.4564E+00	-0.2560E+00
0.315E+01	0.6295E-01	0.8191E-01	0.8494E+00	0.7207E+00
	-0.2677E-02	-0.1307E+00	-0.4619E+00	-0.2598E+00
0.320E+01	0.7138E-01	0.9436E-01	0.9077E+00	0.7668E+00
	-0.3313E-02	-0.1381E+00	-0.4657E+00	-0.2612E+00
0.325E+01	0.7758E-01	0.9794E-01	0.9540E+00	0.8087E+00
	-0.2469E-02	-0.1448E+00	-0.4685E+00	-0.2601E+00
0.330E+01	0.8154E-01	0.9242E-01	0.9867E+00	0.8449E+00
	-0.1114E-03	-0.1508E+00	-0.4711E+00	-0.2572E+00
0.335E+01	0.8344E-01	0.7832E-01	0.1005E+01	0.8743E+00
	0.3739E-02	-0.1560E+00	-0.4745E+00	-0.2531E+00
0.340E+01	0.8363E-01	0.5686E-01	0.1009E+01	0.8964E+00
	0.9016E-02	-0.1604E+00	-0.4797E+00	-0.2488E+00
0.345E+01	0.8259E-01	0.2981E-01	0.1002E+01	0.9117E+00
	0.1561E-01	-0.1639E+00	-0.4874E+00	-0.2451E+00
0.350E+01	0.8086E-01	-0.6740E-03	0.9838E+00	0.9214E+00
	0.2341E-01	-0.1663E+00	-0.4982E+00	-0.2431E+00
0.355E+01	0.7902E-01	-0.3233E-01	0.9587E+00	0.9269E+00
	0.3224E-01	-0.1672E+00	-0.5122E+00	-0.2437E+00
0.360E+01	0.7760E-01	-0.6300E-01	0.9292E+00	0.9303E+00
	0.4196E-01	-0.1660E+00	-0.5290E+00	-0.2474E+00
0.365E+01	0.7706E-01	-0.9082E-01	0.8973E+00	0.9335E+00
	0.5239E-01	-0.1621E+00	-0.5475E+00	-0.2546E+00
0.370E+01	0.7773E-01	-0.1144E+00	0.8645E+00	0.9377E+00
	0.6334E-01	-0.1547E+00	-0.5661E+00	-0.2649E+00
0.375E+01	0.7985E-01	-0.1327E+00	0.8309E+00	0.9438E+00
	0.7459E-01	-0.1431E+00	-0.5829E+00	-0.2778E+00
0.380E+01	0.8354E-01	-0.1452E+00	0.7959E+00	0.9513E+00
	0.8590E-01	-0.1268E+00	-0.5954E+00	-0.2919E+00
0.385E+01	0.8882E-01	-0.1515E+00	0.7579E+00	0.9586E+00
	0.9697E-01	-0.1055E+00	-0.6014E+00	-0.3054E+00
0.390E+01	0.9561E-01	-0.1515E+00	0.7148E+00	0.9628E+00
	0.1075E+00	-0.7906E-01	-0.5984E+00	-0.3163E+00

Table 5.6

C-2

0.395E+01	0.1038E+00	-0.1448E+00	0.6649E+00	0.9601E+00
	0.1171E+00	-0.4810E-01	-0.5847E+00	-0.3220E+00
0.400E+01	0.1131E+00	-0.1308E+00	0.6074E+00	0.9458E+00
	0.1253E+00	-0.1363E-01	-0.5594E+00	-0.3199E+00
0.405E+01	0.1232E+00	-0.1087E+00	0.5433E+00	0.9153E+00
	0.1319E+00	0.2277E-01	-0.5229E+00	-0.3075E+00
0.410E+01	0.1337E+00	-0.7760E-01	0.4758E+00	0.8644E+00
	0.1365E+00	0.5911E-01	-0.4770E+00	-0.2829E+00
0.415E+01	0.1441E+00	-0.3683E-01	0.4111E+00	0.7902E+00
	0.1388E+00	0.9311E-01	-0.4253E+00	-0.2450E+00
0.420E+01	0.1536E+00	0.1354E-01	0.3582E+00	0.6922E+00
	0.1388E+00	0.1226E+00	-0.3730E+00	-0.1940E+00
0.425E+01	0.1613E+00	0.7189E-01	0.3279E+00	0.5730E+00
	0.1365E+00	0.1460E+00	-0.3265E+00	-0.1319E+00
0.430E+01	0.1664E+00	0.1343E+00	0.3320E+00	0.4393E+00
	0.1323E+00	0.1626E+00	-0.2926E+00	-0.6276E-01
0.435E+01	0.1682E+00	0.1938E+00	0.3798E+00	0.3016E+00
	0.1265E+00	0.1732E+00	-0.2770E+00	0.7521E-02
0.440E+01	0.1663E+00	0.2406E+00	0.4755E+00	0.1741E+00
	0.1197E+00	0.1801E+00	-0.2827E+00	0.7160E-01
0.445E+01	0.1613E+00	0.2628E+00	0.6139E+00	0.7275E-01
	0.1121E+00	0.1866E+00	-0.3082E+00	0.1218E+00
0.450E+01	0.1549E+00	0.2492E+00	0.7775E+00	0.1272E-01
	0.1041E+00	0.1960E+00	-0.3470E+00	0.1517E+00
0.455E+01	0.1501E+00	0.1932E+00	0.9358E+00	0.4590E-02
	0.9568E-01	0.2106E+00	-0.3874E+00	0.1579E+00
0.460E+01	0.1507E+00	0.9818E-01	0.1049E+01	0.5059E-01
	0.8683E-01	0.2292E+00	-0.4151E+00	0.1414E+00
0.465E+01	0.1599E+00	-0.1759E-01	0.1074E+01	0.1419E+00
	0.7789E-01	0.2474E+00	-0.4179E+00	0.1081E+00
0.470E+01	0.1780E+00	-0.1193E+00	0.9800E+00	0.2588E+00
	0.6998E-01	0.2578E+00	-0.3901E+00	0.6780E-01
0.475E+01	0.1987E+00	-0.1599E+00	0.7612E+00	0.3739E+00
	0.6498E-01	0.2542E+00	-0.3366E+00	0.3115E-01
0.480E+01	0.2064E+00	-0.9132E-01	0.4454E+00	0.4597E+00
	0.6462E-01	0.2360E+00	-0.2725E+00	0.6209E-02
0.485E+01	0.1748E+00	0.1164E+00	0.9405E-01	0.4972E+00
	0.6765E-01	0.2134E+00	-0.2175E+00	-0.4461E-02
0.490E+01	0.6929E-01	0.4566E+00	-0.2124E+00	0.4820E+00
	0.6482E-01	0.2058E+00	-0.1857E+00	-0.4411E-02
0.495E+01	-0.1460E+00	0.8744E+00	-0.4006E+00	0.4260E+00
	0.3145E-01	0.2324E+00	-0.1772E+00	-0.8386E-03
0.500E+01	-0.4800E+00	0.1256E+01	-0.4800E-01	0.9600E-01
	0.0000E+00	0.2400E+00	-0.1600E+00	0.0000E+00

Table 5.6 The Optimal Third Order Feedback Tensors Indexed by Time for Example 2

This was chosen by taking into account the nature of the problem at hand, especially the finite escape time for $||x|| > 10$. So the norm of the state must remain in the region with no finite escape time. (Finite escape time implies that the norm of the state becomes infinite almost instaneously.) For the asymptotic part of the stability, it was desired that the norm of the state be approaching zero. Since it is a finite time problem, T can not be chosen arbitrarily large and μ can not be chosen arbitrarily small. It is logical to choose T as large as possible (5 sec); μ is chosen equal to the interval between tested points in the region, 0.1 (the smallest change in the norm that mattered). So a point is in the region of usefulness if the norm of the state remains bounded and the final norm is less than 0.1.

There were three regions of usefulness calculated. The first used only linear feedback tensors, the second used linear and quadratic tensors, and the third used K^1_1, $K^1_{(2)}$, and $K^1_{(3)}$. The three regions are shown in Figures 5.5, 5.6 and 5.7, respectively. The regions get larger as new terms are added and the feedback more closely approximates the optimal feedback. The plots bear out the fact that the second order terms help drive the states to the origin faster than the linear terms alone and that the third order terms help even more. Some plots are included in Appendix B.

When nonlinear feedback is used in practice, these regions of usefulness could be very meaningful. In general these regions show the deviation from an equilibrium point from which the system can recover using nonlinear feedback. Obviously, the larger these regions are, the larger the initial conditions which can be tolerated are. So the demonstration that the regions grow as more feedback terms are used is an important result for the application of nonlinear feedback control.

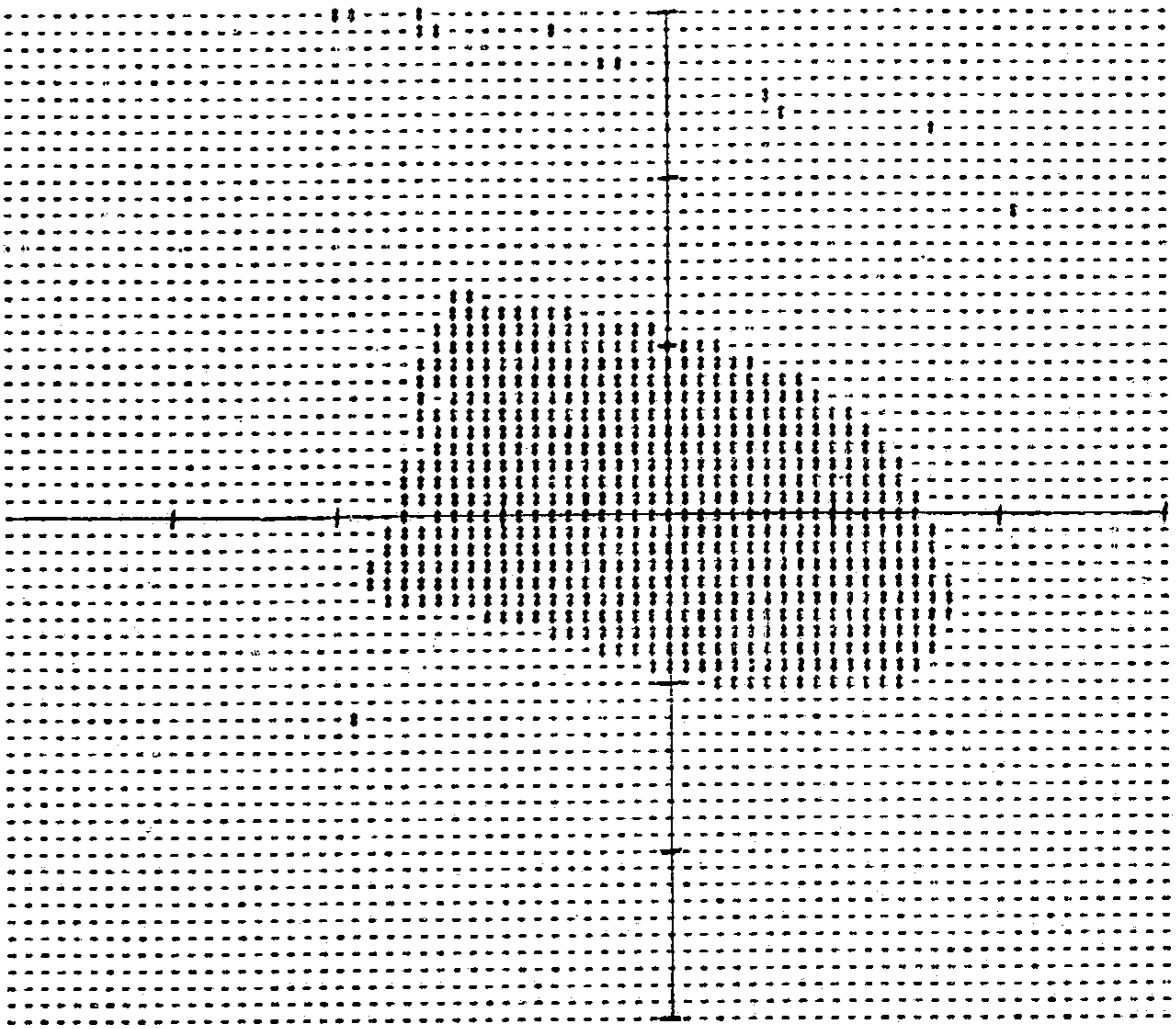


Figure 5.5 The Linear Feedback Region of Usefulness for Example 2. This Region was calculated by the program STAB. Every initial condition in the region $-4 < x(1) < 3$ and $-3 < x(2) < 3$ at 0.1 intervals was input to the subroutine XCALC and integrated for 5 seconds. The asterisks represent the acceptable initial conditions. The has marks are at 1.0 spacings.

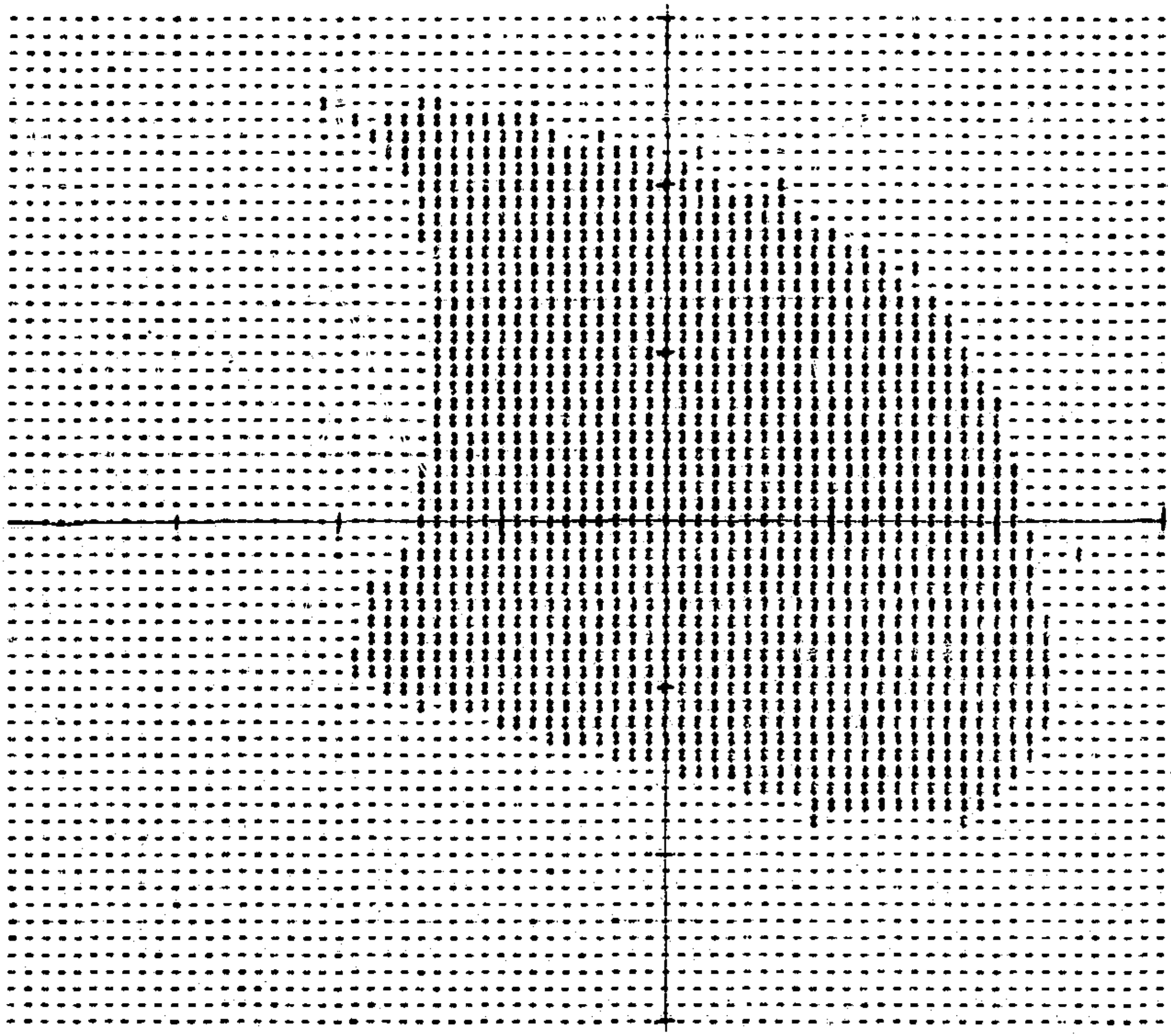


Figure 5.6 The Quadratic Region of Usefulness for Example 2. The dimensions are the same as those in Figure 5.5.

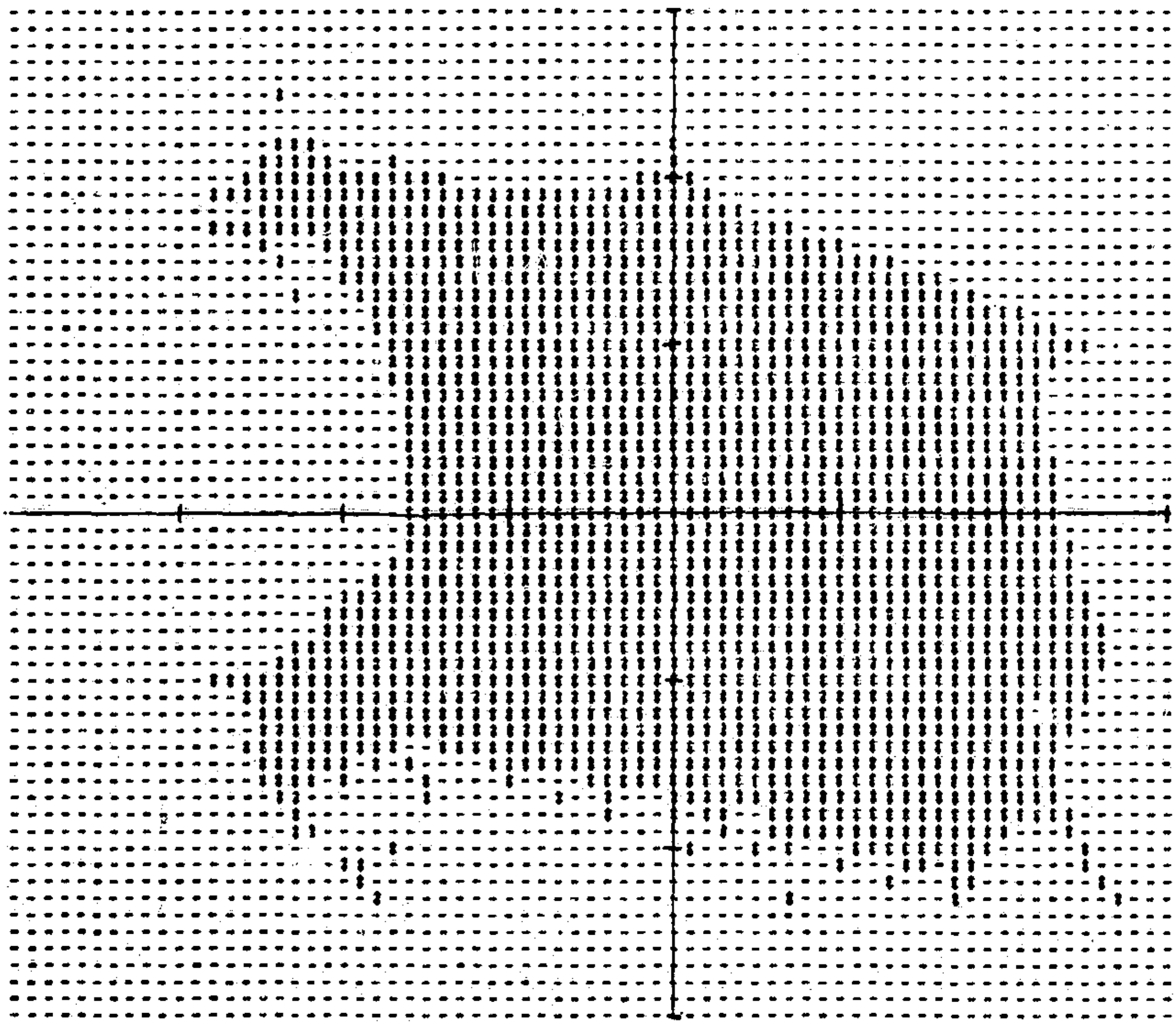


Figure 5.7 The Third Order Region of Usefulness for Example 2. The dimensions are the same as those in Figure 5.5.

EXAMPLE 3

The final example is motivated by the potential applications of nonlinear feedback. When nonlinear feedback is used in practice, the designer will have access to only a model of the actual system. The actual system will not be known explicitly. Thus the system tensors used to calculate the feedback tensors will be only approximations of the actual system tensors obtained by some identification scheme. This example looks at how the feedback tensors calculated from this sort of model perform in comparison to the tensors calculated assuming the actual system is known explicitly, in one case.

The system chosen is the one in the last example. The model used was obtained by Yurkovich [8] and Bugajski [9] using an identification technique devised by Yurkovich. The model was obtained in symmetric tensor form and is shown in Figure 5.8. The same cost functional as for the last example, Equation (1), is used so that the results may be compared directly. Again the tensors $V_{(2)}$, $V_{(3)}$, $V_{(4)}$, K^1_1, $K^1_{(2)}$, and $K^1_{(3)}$ were calculated over the 5 seconds time span and were saved every .05 seconds. Their values are shown in Tables 5.7 through 5.12, respectively. These values are in fact close to the values calculated on the actual system.

The program STAB was run for these new values of the feedback tensors. The possible application was the motivation for the manner in which this program was executed also. In practice it is anticipated that the designer will first obtain a model of the system which approximates system behavior in the region of interest. Next, the feedback tensors for this model will be calculated to minimize some cost functional. Then these feedback tensors will be used on the actual system to calculate the input to that system. The important point here is that the feedback will not be applied to the model of the system but to the actual system itself. Models of systems are developed for analysis purposes while the goal of calculating the tensors is for controlling real systems. So here STAB uses the feedback tensors calculated using the model in Figure 5.8 to control the actual system in Equation (2).

The linear, quadratic, and third order regions of usefulness are shown in Figures 5.9, 5.10, and 5.11, respectively. These regions are very similar to those in Figures 5.5, 5.6, and 5.7. For this example, there is no significant difference between using an identified model of the system and the actual system.

$$A = \begin{bmatrix} -1.9977 & -3.0008 \\ .99967 & 1.0002 \end{bmatrix} \quad B = \begin{bmatrix} .0019891 & .99779 \\ -1.0001 & -6.2231E-7 \end{bmatrix}$$

$$A_{0,(2)} = \begin{bmatrix} -.19024 \\ -.012365 \\ -.0034771 \\ .036033 \\ .0022184 \\ 1.5017E-4 \end{bmatrix} \quad A_{1,1} = \begin{bmatrix} -4.0692 \\ .027909 \\ .026129 \\ -.0078805 \\ .021083 \\ .0026825 \\ .002591 \\ .0035398 \end{bmatrix} \quad A_{(2),0} = \begin{bmatrix} -.0039675 \\ .0071601 \\ -.012386 \\ -.99787 \\ .0010537 \\ 7.4681E-4 \end{bmatrix}$$

$$A_{0,(3)} = \begin{bmatrix} .70548 \\ 1.6627 \\ 3.3163 \\ .78087 \\ -.2223 \\ -.6684 \\ -1.2624 \\ -.3395 \end{bmatrix} \quad A_{(3),0} = \begin{bmatrix} -.18738 \\ -.032069 \\ -.0251 \\ -.019747 \\ -.51749 \\ -.016328 \\ -.019048 \\ -.0011802 \end{bmatrix}$$

$$A_{1,(2)} = \begin{bmatrix} .62148 \\ -.25223 \\ .44724 \\ -.61366 \\ -.76455 \\ -.67544 \\ -.28909 \\ -.05473 \\ -.048646 \\ -.025264 \\ -.058202 \\ -.0056896 \end{bmatrix} \quad A_{(2),1} = \begin{bmatrix} -3.5488 \\ .16518 \\ -.10235 \\ .036896 \\ .41411 \\ .075615 \\ -.17155 \\ -.055923 \\ .96002 \\ -.035393 \\ -.044252 \\ -.013022 \end{bmatrix}$$

Figure 5.8 Identified Tensors for the System in Equation (2).

TIME	V2S(1)	V2S(2)	V2S(3)
0.000E+00	0.1795E+00	0.3258E+00	0.5183E+00
0.500E-01	0.1799E+00	0.3261E+00	0.5181E+00
0.100E+00	0.1804E+00	0.3266E+00	0.5180E+00
0.150E+00	0.1810E+00	0.3274E+00	0.5180E+00
0.200E+00	0.1816E+00	0.3284E+00	0.5181E+00
0.250E+00	0.1822E+00	0.3297E+00	0.5184E+00
0.300E+00	0.1829E+00	0.3313E+00	0.5190E+00
0.350E+00	0.1835E+00	0.3330E+00	0.5198E+00
0.400E+00	0.1842E+00	0.3350E+00	0.5208E+00
0.450E+00	0.1848E+00	0.3371E+00	0.5222E+00
0.500E+00	0.1853E+00	0.3393E+00	0.5237E+00
0.550E+00	0.1858E+00	0.3416E+00	0.5256E+00
0.600E+00	0.1863E+00	0.3439E+00	0.5277E+00
0.650E+00	0.1866E+00	0.3461E+00	0.5300E+00
0.700E+00	0.1868E+00	0.3481E+00	0.5324E+00
0.750E+00	0.1868E+00	0.3499E+00	0.5350E+00
0.800E+00	0.1868E+00	0.3514E+00	0.5377E+00
0.850E+00	0.1866E+00	0.3525E+00	0.5403E+00
0.900E+00	0.1863E+00	0.3532E+00	0.5429E+00
0.950E+00	0.1859E+00	0.3533E+00	0.5453E+00
0.100E+01	0.1854E+00	0.3529E+00	0.5475E+00
0.105E+01	0.1848E+00	0.3520E+00	0.5494E+00
0.110E+01	0.1843E+00	0.3505E+00	0.5509E+00
0.115E+01	0.1837E+00	0.3485E+00	0.5520E+00
0.120E+01	0.1833E+00	0.3461E+00	0.5526E+00
0.125E+01	0.1829E+00	0.3433E+00	0.5528E+00
0.130E+01	0.1827E+00	0.3403E+00	0.5523E+00
0.135E+01	0.1828E+00	0.3372E+00	0.5514E+00
0.140E+01	0.1831E+00	0.3342E+00	0.5501E+00
0.145E+01	0.1838E+00	0.3314E+00	0.5483E+00
0.150E+01	0.1848E+00	0.3290E+00	0.5463E+00
0.155E+01	0.1863E+00	0.3274E+00	0.5440E+00
0.160E+01	0.1882E+00	0.3266E+00	0.5418E+00
0.165E+01	0.1905E+00	0.3269E+00	0.5397E+00
0.170E+01	0.1933E+00	0.3286E+00	0.5379E+00
0.175E+01	0.1965E+00	0.3316E+00	0.5367E+00
0.180E+01	0.2001E+00	0.3363E+00	0.5363E+00
0.185E+01	0.2040E+00	0.3426E+00	0.5369E+00
0.190E+01	0.2082E+00	0.3506E+00	0.5386E+00
0.195E+01	0.2126E+00	0.3604E+00	0.5418E+00
0.200E+01	0.2170E+00	0.3717E+00	0.5465E+00
0.205E+01	0.2215E+00	0.3845E+00	0.5529E+00
0.210E+01	0.2257E+00	0.3984E+00	0.5612E+00
0.215E+01	0.2297E+00	0.4134E+00	0.5712E+00
0.220E+01	0.2333E+00	0.4288E+00	0.5830E+00
0.225E+01	0.2364E+00	0.4444E+00	0.5966E+00
0.230E+01	0.2388E+00	0.4597E+00	0.6117E+00
0.235E+01	0.2404E+00	0.4741E+00	0.6281E+00
0.240E+01	0.2413E+00	0.4872E+00	0.6456E+00
0.245E+01	0.2413E+00	0.4984E+00	0.6637E+00
0.250E+01	0.2404E+00	0.5071E+00	0.6821E+00
0.255E+01	0.2387E+00	0.5130E+00	0.7002E+00

Table 5.7

0.260E+01	0.2363E+00	0.5157E+00	0.7176E+00
0.265E+01	0.2332E+00	0.5147E+00	0.7337E+00
0.270E+01	0.2296E+00	0.5101E+00	0.7480E+00
0.275E+01	0.2258E+00	0.5016E+00	0.7600E+00
0.280E+01	0.2220E+00	0.4895E+00	0.7692E+00
0.285E+01	0.2185E+00	0.4740E+00	0.7752E+00
0.290E+01	0.2157E+00	0.4557E+00	0.7778E+00
0.295E+01	0.2139E+00	0.4352E+00	0.7767E+00
0.300E+01	0.2135E+00	0.4135E+00	0.7721E+00
0.305E+01	0.2150E+00	0.3917E+00	0.7640E+00
0.310E+01	0.2186E+00	0.3711E+00	0.7528E+00
0.315E+01	0.2248E+00	0.3530E+00	0.7392E+00
0.320E+01	0.2337E+00	0.3390E+00	0.7239E+00
0.325E+01	0.2457E+00	0.3305E+00	0.7078E+00
0.330E+01	0.2609E+00	0.3292E+00	0.6922E+00
0.335E+01	0.2793E+00	0.3364E+00	0.6784E+00
0.340E+01	0.3008E+00	0.3536E+00	0.6679E+00
0.345E+01	0.3253E+00	0.3818E+00	0.6622E+00
0.350E+01	0.3525E+00	0.4218E+00	0.6629E+00
0.355E+01	0.3819E+00	0.4742E+00	0.6716E+00
0.360E+01	0.4131E+00	0.5392E+00	0.6898E+00
0.365E+01	0.4454E+00	0.6164E+00	0.7190E+00
0.370E+01	0.4782E+00	0.7053E+00	0.7605E+00
0.375E+01	0.5105E+00	0.8048E+00	0.8151E+00
0.380E+01	0.5417E+00	0.9131E+00	0.8838E+00
0.385E+01	0.5707E+00	0.1028E+01	0.9668E+00
0.390E+01	0.5968E+00	0.1148E+01	0.1064E+01
0.395E+01	0.6190E+00	0.1269E+01	0.1176E+01
0.400E+01	0.6366E+00	0.1388E+01	0.1301E+01
0.405E+01	0.6489E+00	0.1502E+01	0.1438E+01
0.410E+01	0.6552E+00	0.1606E+01	0.1586E+01
0.415E+01	0.6552E+00	0.1697E+01	0.1742E+01
0.420E+01	0.6486E+00	0.1770E+01	0.1902E+01
0.425E+01	0.6354E+00	0.1821E+01	0.2065E+01
0.430E+01	0.6160E+00	0.1845E+01	0.2225E+01
0.435E+01	0.5913E+00	0.1840E+01	0.2378E+01
0.440E+01	0.5624E+00	0.1802E+01	0.2520E+01
0.445E+01	0.5310E+00	0.1729E+01	0.2644E+01
0.450E+01	0.4994E+00	0.1621E+01	0.2746E+01
0.455E+01	0.4705E+00	0.1478E+01	0.2820E+01
0.460E+01	0.4477E+00	0.1303E+01	0.2862E+01
0.465E+01	0.4349E+00	0.1103E+01	0.2868E+01
0.470E+01	0.4364E+00	0.8849E+00	0.2835E+01
0.475E+01	0.4570E+00	0.6603E+00	0.2764E+01
0.480E+01	0.5014E+00	0.4430E+00	0.2656E+01
0.485E+01	0.5741E+00	0.2494E+00	0.2516E+01
0.490E+01	0.6792E+00	0.9776E-01	0.2353E+01
0.495E+01	0.8202E+00	0.7870E-02	0.2176E+01
0.500E+01	0.1000E+01	0.0000E+00	0.2000E+01

Table 5.7 Quadratic Terms of the Optimal Cost Functional Indexed by Time for Example 3

TIME	V3(1)	V3(2)	V3(3)	V3(4)
0.000E+00	0.3501E-02	-0.1363E+00	-0.1606E+00	-0.6564E+00
0.500E-01	0.3429E-02	-0.1364E+00	-0.1601E+00	-0.6549E+00
0.100E+00	0.3267E-02	-0.1366E+00	-0.1598E+00	-0.6534E+00
0.150E+00	0.3009E-02	-0.1370E+00	-0.1598E+00	-0.6520E+00
0.200E+00	0.2649E-02	-0.1375E+00	-0.1599E+00	-0.6508E+00
0.250E+00	0.2187E-02	-0.1381E+00	-0.1604E+00	-0.6499E+00
0.300E+00	0.1625E-02	-0.1389E+00	-0.1612E+00	-0.6495E+00
0.350E+00	0.9697E-03	-0.1398E+00	-0.1623E+00	-0.6496E+00
0.400E+00	0.2324E-03	-0.1409E+00	-0.1637E+00	-0.6504E+00
0.450E+00	-0.5721E-03	-0.1421E+00	-0.1655E+00	-0.6519E+00
0.500E+00	-0.1425E-02	-0.1434E+00	-0.1677E+00	-0.6542E+00
0.550E+00	-0.2304E-02	-0.1447E+00	-0.1701E+00	-0.6575E+00
0.600E+00	-0.3183E-02	-0.1461E+00	-0.1727E+00	-0.6617E+00
0.650E+00	-0.4034E-02	-0.1475E+00	-0.1756E+00	-0.6668E+00
0.700E+00	-0.4827E-02	-0.1489E+00	-0.1786E+00	-0.6728E+00
0.750E+00	-0.5533E-02	-0.1502E+00	-0.1816E+00	-0.6796E+00
0.800E+00	-0.6120E-02	-0.1514E+00	-0.1845E+00	-0.6872E+00
0.850E+00	-0.6561E-02	-0.1524E+00	-0.1873E+00	-0.6952E+00
0.900E+00	-0.6831E-02	-0.1532E+00	-0.1898E+00	-0.7037E+00
0.950E+00	-0.6907E-02	-0.1538E+00	-0.1919E+00	-0.7123E+00
0.100E+01	-0.6774E-02	-0.1541E+00	-0.1936E+00	-0.7209E+00
0.105E+01	-0.6423E-02	-0.1542E+00	-0.1947E+00	-0.7291E+00
0.110E+01	-0.5855E-02	-0.1540E+00	-0.1952E+00	-0.7368E+00
0.115E+01	-0.5078E-02	-0.1535E+00	-0.1950E+00	-0.7436E+00
0.120E+01	-0.4112E-02	-0.1527E+00	-0.1941E+00	-0.7492E+00
0.125E+01	-0.2988E-02	-0.1517E+00	-0.1925E+00	-0.7535E+00
0.130E+01	-0.1750E-02	-0.1505E+00	-0.1902E+00	-0.7563E+00
0.135E+01	-0.4536E-03	-0.1492E+00	-0.1872E+00	-0.7573E+00
0.140E+01	0.8341E-03	-0.1479E+00	-0.1838E+00	-0.7566E+00
0.145E+01	0.2034E-02	-0.1465E+00	-0.1799E+00	-0.7540E+00
0.150E+01	0.3058E-02	-0.1453E+00	-0.1758E+00	-0.7497E+00
0.155E+01	0.3810E-02	-0.1444E+00	-0.1716E+00	-0.7437E+00
0.160E+01	0.4188E-02	-0.1438E+00	-0.1676E+00	-0.7364E+00
0.165E+01	0.4090E-02	-0.1437E+00	-0.1640E+00	-0.7279E+00
0.170E+01	0.3413E-02	-0.1443E+00	-0.1612E+00	-0.7189E+00
0.175E+01	0.2065E-02	-0.1456E+00	-0.1594E+00	-0.7098E+00
0.180E+01	-0.3781E-04	-0.1478E+00	-0.1588E+00	-0.7012E+00
0.185E+01	-0.2962E-02	-0.1510E+00	-0.1599E+00	-0.6939E+00
0.190E+01	-0.6753E-02	-0.1553E+00	-0.1629E+00	-0.6885E+00
0.195E+01	-0.1143E-01	-0.1608E+00	-0.1680E+00	-0.6859E+00
0.200E+01	-0.1698E-01	-0.1675E+00	-0.1754E+00	-0.6869E+00
0.205E+01	-0.2335E-01	-0.1754E+00	-0.1854E+00	-0.6924E+00
0.210E+01	-0.3047E-01	-0.1846E+00	-0.1980E+00	-0.7031E+00
0.215E+01	-0.3821E-01	-0.1948E+00	-0.2132E+00	-0.7198E+00
0.220E+01	-0.4641E-01	-0.2060E+00	-0.2311E+00	-0.7431E+00
0.225E+01	-0.5487E-01	-0.2181E+00	-0.2514E+00	-0.7736E+00
0.230E+01	-0.6336E-01	-0.2307E+00	-0.2740E+00	-0.8116E+00
0.235E+01	-0.7162E-01	-0.2436E+00	-0.2984E+00	-0.8573E+00
0.240E+01	-0.7937E-01	-0.2565E+00	-0.3241E+00	-0.9107E+00
0.245E+01	-0.8632E-01	-0.2690E+00	-0.3505E+00	-0.9713E+00
0.250E+01	-0.9216E-01	-0.2807E+00	-0.3770E+00	-0.1039E+01
0.255E+01	-0.9662E-01	-0.2913E+00	-0.4027E+00	-0.1112E+01

Table 5.8

0.260E+01	-0.9943E-01	-0.3002E+00	-0.4267E+00	-0.1189E+01
0.265E+01	-0.1004E+00	-0.3072E+00	-0.4481E+00	-0.1270E+01
0.270E+01	-0.9934E-01	-0.3119E+00	-0.4659E+00	-0.1351E+01
0.275E+01	-0.9619E-01	-0.3139E+00	-0.4790E+00	-0.1430E+01
0.280E+01	-0.9094E-01	-0.3133E+00	-0.4868E+00	-0.1506E+01
0.285E+01	-0.8370E-01	-0.3098E+00	-0.4883E+00	-0.1576E+01
0.290E+01	-0.7466E-01	-0.3037E+00	-0.4830E+00	-0.1638E+01
0.295E+01	-0.6412E-01	-0.2952E+00	-0.4708E+00	-0.1685E+01
0.300E+01	-0.5250E-01	-0.2847E+00	-0.4515E+00	-0.1719E+01
0.305E+01	-0.4024E-01	-0.2729E+00	-0.4256E+00	-0.1737E+01
0.310E+01	-0.2790E-01	-0.2606E+00	-0.3939E+00	-0.1738E+01
0.315E+01	-0.1602E-01	-0.2486E+00	-0.3577E+00	-0.1720E+01
0.320E+01	-0.5133E-02	-0.2379E+00	-0.3187E+00	-0.1685E+01
0.325E+01	0.4264E-02	-0.2297E+00	-0.2788E+00	-0.1633E+01
0.330E+01	0.1178E-01	-0.2247E+00	-0.2404E+00	-0.1566E+01
0.335E+01	0.1715E-01	-0.2238E+00	-0.2058E+00	-0.1488E+01
0.340E+01	0.2026E-01	-0.2276E+00	-0.1776E+00	-0.1403E+01
0.345E+01	0.2118E-01	-0.2366E+00	-0.1579E+00	-0.1316E+01
0.350E+01	0.2014E-01	-0.2506E+00	-0.1489E+00	-0.1232E+01
0.355E+01	0.1756E-01	-0.2694E+00	-0.1520E+00	-0.1157E+01
0.360E+01	0.1397E-01	-0.2922E+00	-0.1682E+00	-0.1097E+01
0.365E+01	0.1003E-01	-0.3179E+00	-0.1976E+00	-0.1057E+01
0.370E+01	0.6463E-02	-0.3453E+00	-0.2397E+00	-0.1040E+01
0.375E+01	0.3987E-02	-0.3725E+00	-0.2929E+00	-0.1053E+01
0.380E+01	0.3271E-02	-0.3979E+00	-0.3551E+00	-0.1095E+01
0.385E+01	0.4883E-02	-0.4194E+00	-0.4231E+00	-0.1169E+01
0.390E+01	0.9231E-02	-0.4353E+00	-0.4931E+00	-0.1273E+01
0.395E+01	0.1651E-01	-0.4439E+00	-0.5609E+00	-0.1406E+01
0.400E+01	0.2668E-01	-0.4437E+00	-0.6214E+00	-0.1561E+01
0.405E+01	0.3940E-01	-0.4339E+00	-0.6699E+00	-0.1734E+01
0.410E+01	0.5408E-01	-0.4141E+00	-0.7012E+00	-0.1916E+01
0.415E+01	0.6985E-01	-0.3847E+00	-0.7108E+00	-0.2096E+01
0.420E+01	0.8568E-01	-0.3470E+00	-0.6950E+00	-0.2262E+01
0.425E+01	0.1004E+00	-0.3031E+00	-0.6514E+00	-0.2402E+01
0.430E+01	0.1129E+00	-0.2562E+00	-0.5794E+00	-0.2502E+01
0.435E+01	0.1225E+00	-0.2101E+00	-0.4805E+00	-0.2550E+01
0.440E+01	0.1287E+00	-0.1691E+00	-0.3592E+00	-0.2535E+01
0.445E+01	0.1320E+00	-0.1376E+00	-0.2227E+00	-0.2451E+01
0.450E+01	0.1337E+00	-0.1191E+00	-0.8092E-01	-0.2294E+01
0.455E+01	0.1359E+00	-0.1159E+00	0.5402E-01	-0.2070E+01
0.460E+01	0.1414E+00	-0.1276E+00	0.1694E+00	-0.1789E+01
0.465E+01	0.1531E+00	-0.1508E+00	0.2534E+00	-0.1471E+01
0.470E+01	0.1724E+00	-0.1789E+00	0.2977E+00	-0.1140E+01
0.475E+01	0.1985E+00	-0.2025E+00	0.2991E+00	-0.8204E+00
0.480E+01	0.2258E+00	-0.2111E+00	0.2615E+00	-0.5385E+00
0.485E+01	0.2427E+00	-0.1955E+00	0.1957E+00	-0.3123E+00
0.490E+01	0.2302E+00	-0.1509E+00	0.1183E+00	-0.1507E+00
0.495E+01	0.1610E+00	-0.8079E-01	0.4785E-01	-0.5104E-01
0.500E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00

Table 5.8 Third Order Terms of the Optimal Cost Functional Indexed by Time for Example 3

TIME	V4(1)	V4(2)	V4(3)	V4(4)	V4(5)
0.000E+00	0.5847E-01	0.3938E+00	0.1174E+01	0.1187E+01	0.1216E+01
0.500E-01	0.5845E-01	0.3958E+00	0.1177E+01	0.1191E+01	0.1218E+01
0.100E+00	0.5833E-01	0.3977E+00	0.1181E+01	0.1195E+01	0.1220E+01
0.150E+00	0.5808E-01	0.3995E+00	0.1185E+01	0.1200E+01	0.1222E+01
0.200E+00	0.5770E-01	0.4010E+00	0.1188E+01	0.1204E+01	[illegible]
0.250E+00	0.5719E-01	0.4021E+00	0.1192E+01	0.1209E+01	0.1228E+01
0.300E+00	0.5654E-01	0.4028E+00	0.1195E+01	0.1214E+01	0.1231E+01
0.350E+00	0.5576E-01	0.4030E+00	0.1198E+01	0.1218E+01	0.1234E+01
0.400E+00	0.5486E-01	0.4027E+00	0.1200E+01	0.1222E+01	0.1238E+01
0.450E+00	0.5387E-01	0.4017E+00	0.1201E+01	0.1225E+01	0.1241E+01
0.500E+00	0.5280E-01	0.4000E+00	0.1201E+01	0.1227E+01	0.1245E+01
0.550E+00	0.5167E-01	0.3976E+00	0.1199E+01	0.1227E+01	0.1248E+01
0.600E+00	0.5053E-01	0.3946E+00	0.1197E+01	0.1226E+01	0.1250E+01
0.650E+00	0.4941E-01	0.3909E+00	0.1193E+01	0.1223E+01	0.1252E+01
0.700E+00	0.4835E-01	0.3867E+00	0.1188E+01	0.1219E+01	0.1253E+01
0.750E+00	0.4739E-01	0.3821E+00	0.1182E+01	0.1213E+01	0.1254E+01
0.800E+00	0.4659E-01	0.3771E+00	0.1175E+01	0.1205E+01	0.1253E+01
0.850E+00	0.4598E-01	0.3720E+00	0.1167E+01	0.1195E+01	0.1251E+01
0.900E+00	0.4560E-01	0.3669E+00	0.1158E+01	0.1184E+01	0.1249E+01
0.950E+00	0.4550E-01	0.3621E+00	0.1149E+01	0.1172E+01	0.1245E+01
0.100E+01	0.4571E-01	0.3578E+00	0.1140E+01	0.1159E+01	0.1240E+01
0.105E+01	0.4626E-01	0.3543E+00	0.1131E+01	0.1145E+01	0.1234E+01
0.110E+01	0.4714E-01	0.3519E+00	0.1123E+01	0.1132E+01	0.1227E+01
0.115E+01	0.4837E-01	0.3507E+00	0.1116E+01	0.1120E+01	0.1219E+01
0.120E+01	0.4992E-01	0.3511E+00	0.1111E+01	0.1109E+01	0.1211E+01
0.125E+01	0.5176E-01	0.3533E+00	0.1108E+01	0.1100E+01	0.1203E+01
0.130E+01	0.5383E-01	0.3573E+00	0.1108E+01	0.1094E+01	0.1196E+01
0.135E+01	0.5606E-01	0.3634E+00	0.1111E+01	0.1092E+01	0.1189E+01
0.140E+01	0.5837E-01	0.3714E+00	0.1118E+01	0.1093E+01	0.1183E+01
0.145E+01	0.6064E-01	0.3814E+00	0.1128E+01	0.1100E+01	0.1179E+01
0.150E+01	0.6276E-01	0.3931E+00	0.1142E+01	0.1111E+01	0.1177E+01
0.155E+01	0.6462E-01	0.4064E+00	0.1160E+01	0.1128E+01	0.1177E+01
0.160E+01	0.6610E-01	0.4208E+00	0.1181E+01	0.1149E+01	0.1180E+01
0.165E+01	0.6709E-01	0.4361E+00	0.1205E+01	0.1176E+01	0.1187E+01
0.170E+01	0.6750E-01	0.4517E+00	0.1231E+01	0.1208E+01	0.1197E+01
0.175E+01	0.6726E-01	0.4671E+00	0.1259E+01	0.1243E+01	0.1210E+01
0.180E+01	0.6634E-01	0.4821E+00	0.1289E+01	0.1282E+01	0.1227E+01
0.185E+01	0.6473E-01	0.4961E+00	0.1318E+01	0.1322E+01	0.1247E+01
0.190E+01	0.6249E-01	0.5089E+00	0.1348E+01	0.1364E+01	0.1271E+01
0.195E+01	0.5968E-01	0.5202E+00	0.1376E+01	0.1405E+01	0.1297E+01
0.200E+01	0.5642E-01	0.5300E+00	0.1404E+01	0.1445E+01	0.1325E+01
0.205E+01	0.5286E-01	0.5383E+00	0.1430E+01	0.1484E+01	0.1355E+01
0.210E+01	0.4917E-01	0.5454E+00	0.1456E+01	0.1519E+01	0.1386E+01
0.215E+01	0.4555E-01	0.5513E+00	0.1482E+01	0.1553E+01	0.1418E+01
0.220E+01	0.4221E-01	0.5566E+00	0.1508E+01	0.1584E+01	0.1451E+01
0.225E+01	0.3934E-01	0.5616E+00	0.1536E+01	0.1614E+01	0.1485E+01
0.230E+01	0.3714E-01	0.5666E+00	0.1566E+01	0.1645E+01	0.1519E+01
0.235E+01	0.3580E-01	0.5722E+00	0.1601E+01	0.1677E+01	0.1555E+01
0.240E+01	0.3547E-01	0.5785E+00	0.1640E+01	0.1713E+01	0.1593E+01
0.245E+01	0.3629E-01	0.5859E+00	0.1685E+01	0.1755E+01	0.1634E+01
0.250E+01	0.3836E-01	0.5945E+00	0.1735E+01	0.1804E+01	0.1680E+01

Table 5.9

0.255E+01	0.4174E-01	0.6043E+00	0.1791E+01	0.1863E+01	0.1731E+01
0.260E+01	0.4645E-01	0.6155E+00	0.1852E+01	0.1933E+01	0.1789E+01
0.265E+01	0.5248E-01	0.6279E+00	0.1917E+01	0.2012E+01	0.1855E+01
0.270E+01	0.5975E-01	0.6414E+00	0.1983E+01	0.2101E+01	0.1930E+01
0.275E+01	0.6816E-01	0.6559E+00	0.2050E+01	0.2197E+01	0.2014E+01
0.280E+01	0.7753E-01	0.6714E+00	0.2114E+01	0.2296E+01	0.2107E+01
0.285E+01	0.8767E-01	0.6878E+00	0.2173E+01	0.2396E+01	0.2207E+01
0.290E+01	0.9832E-01	0.7053E+00	0.2225E+01	0.2491E+01	0.2313E+01
0.295E+01	0.1092E+00	0.7240E+00	0.2269E+01	0.2575E+01	0.2421E+01
0.300E+01	0.1200E+00	0.7443E+00	0.2305E+01	0.2643E+01	0.2529E+01
0.305E+01	0.1305E+00	0.7666E+00	0.2332E+01	0.2692E+01	0.2632E+01
0.310E+01	0.1403E+00	0.7915E+00	0.2354E+01	0.2719E+01	0.2725E+01
0.315E+01	0.1491E+00	0.8196E+00	0.2374E+01	0.2723E+01	0.2805E+01
0.320E+01	0.1569E+00	0.8513E+00	0.2395E+01	0.2706E+01	0.2869E+01
0.325E+01	0.1633E+00	0.8870E+00	0.2424E+01	0.2674E+01	0.2914E+01
0.330E+01	0.1683E+00	0.9268E+00	0.2465E+01	0.2635E+01	0.2941E+01
0.335E+01	0.1717E+00	0.9702E+00	0.2524E+01	0.2596E+01	0.2951E+01
0.340E+01	0.1735E+00	0.1016E+01	0.2605E+01	0.2571E+01	0.2947E+01
0.345E+01	0.1738E+00	0.1063E+01	0.2708E+01	0.2569E+01	0.2935E+01
0.350E+01	0.1723E+00	0.1109E+01	0.2834E+01	0.2599E+01	0.2921E+01
0.355E+01	0.1691E+00	0.1150E+01	0.2976E+01	0.2669E+01	0.2913E+01
0.360E+01	0.1643E+00	0.1183E+01	0.3129E+01	0.2781E+01	0.2916E+01
0.365E+01	0.1576E+00	0.1204E+01	0.3279E+01	0.2932E+01	0.2939E+01
0.370E+01	0.1492E+00	0.1210E+01	0.3415E+01	0.3114E+01	0.2985E+01
0.375E+01	0.1391E+00	0.1197E+01	0.3521E+01	0.3314E+01	0.3058E+01
0.380E+01	0.1274E+00	0.1162E+01	0.3580E+01	0.3512E+01	0.3157E+01
0.385E+01	0.1142E+00	0.1104E+01	0.3578E+01	0.3685E+01	0.3279E+01
0.390E+01	0.9989E-01	0.1023E+01	0.3501E+01	0.3806E+01	0.3417E+01
0.395E+01	0.8486E-01	0.9194E+00	0.3341E+01	0.3849E+01	0.3560E+01
0.400E+01	0.6964E-01	0.7971E+00	0.3094E+01	0.3785E+01	0.3696E+01
0.405E+01	0.5486E-01	0.6612E+00	0.2763E+01	0.3592E+01	0.3806E+01
0.410E+01	0.4117E-01	0.5187E+00	0.2361E+01	0.3254E+01	0.3873E+01
0.415E+01	0.2920E-01	0.3774E+00	0.1908E+01	0.2764E+01	0.3875E+01
0.420E+01	0.1944E-01	0.2456E+00	0.1435E+01	0.2132E+01	0.3792E+01
0.425E+01	0.1220E-01	0.1302E+00	0.9758E+00	0.1387E+01	0.3609E+01
0.430E+01	0.7572E-02	0.3582E-01	0.5687E+00	0.5749E+00	0.3317E+01
0.435E+01	0.5518E-02	-0.3645E-01	0.2461E+00	-0.2381E+00	0.2915E+01
0.440E+01	0.5997E-02	-0.8919E-01	0.2826E-01	-0.9743E+00	0.2419E+01
0.445E+01	0.9145E-02	-0.1274E+00	-0.8474E-01	-0.1554E+01	0.1856E+01
0.450E+01	0.1536E-01	-0.1556E+00	-0.1166E+00	-0.1914E+01	0.1270E+01
0.455E+01	0.2511E-01	-0.1734E+00	-0.1121E+00	-0.2020E+01	0.7115E+00
0.460E+01	0.3823E-01	-0.1716E+00	-0.1249E+00	-0.1885E+01	0.2306E+00
0.465E+01	0.5271E-01	-0.1319E+00	-0.1974E+00	-0.1568E+01	-0.1322E+00
0.470E+01	0.6347E-01	-0.3232E-01	-0.3378E+00	-0.1165E+01	-0.3568E+00
0.475E+01	0.6188E-01	0.1385E+00	-0.5062E+00	-0.7760E+00	-0.4481E+00
0.480E+01	0.3809E-01	0.3618E+00	-0.6224E+00	-0.4752E+00	-0.4330E+00
0.485E+01	-0.1243E-01	0.5719E+00	-0.6031E+00	-0.2828E+00	-0.3498E+00
0.490E+01	-0.7666E-01	0.6566E+00	-0.4193E+00	-0.1665E+00	-0.2351E+00
0.495E+01	-0.1068E+00	0.4891E+00	-0.1508E+00	-0.7624E-01	-0.1138E+00
0.500E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00

Table 5.9 Fourth Order Terms of the Optimal Cost Functional Indexed by Time for Example 3

TIME	K1(1)	K1(2)	K1(3)	K1(4)
0.000E+00	-0.5675E+00	0.1036E+00	-0.3582E-01	-0.6325E+00
0.500E-01	-0.5675E+00	0.1036E+00	-0.3590E-01	-0.6325E+00
0.100E+00	-0.5674E+00	0.1035E+00	-0.3600E-01	-0.6326E+00
0.150E+00	-0.5673E+00	0.1035E+00	-0.3611E-01	-0.6327E+00
0.200E+00	-0.5672E+00	0.1036E+00	-0.3623E-01	-0.6328E+00
0.250E+00	-0.5671E+00	0.1036E+00	-0.3636E-01	-0.6329E+00
0.300E+00	-0.5669E+00	0.1037E+00	-0.3649E-01	-0.6331E+00
0.350E+00	-0.5668E+00	0.1039E+00	-0.3662E-01	-0.6332E+00
0.400E+00	-0.5666E+00	0.1041E+00	-0.3675E-01	-0.6334E+00
0.450E+00	-0.5664E+00	0.1044E+00	-0.3687E-01	-0.6336E+00
0.500E+00	-0.5661E+00	0.1047E+00	-0.3699E-01	-0.6339E+00
0.550E+00	-0.5659E+00	0.1051E+00	-0.3709E-01	-0.6341E+00
0.600E+00	-0.5657E+00	0.1055E+00	-0.3717E-01	-0.6343E+00
0.650E+00	-0.5655E+00	0.1059E+00	-0.3723E-01	-0.6345E+00
0.700E+00	-0.5653E+00	0.1064E+00	-0.3727E-01	-0.6347E+00
0.750E+00	-0.5651E+00	0.1069E+00	-0.3728E-01	-0.6349E+00
0.800E+00	-0.5649E+00	0.1075E+00	-0.3727E-01	-0.6351E+00
0.850E+00	-0.5648E+00	0.1080E+00	-0.3723E-01	-0.6352E+00
0.900E+00	-0.5648E+00	0.1085E+00	-0.3717E-01	-0.6352E+00
0.950E+00	-0.5647E+00	0.1090E+00	-0.3709E-01	-0.6353E+00
0.100E+01	-0.5648E+00	0.1094E+00	-0.3699E-01	-0.6352E+00
0.105E+01	-0.5649E+00	0.1098E+00	-0.3689E-01	-0.6351E+00
0.110E+01	-0.5650E+00	0.1101E+00	-0.3677E-01	-0.6350E+00
0.115E+01	-0.5652E+00	0.1103E+00	-0.3667E-01	-0.6348E+00
0.120E+01	-0.5655E+00	0.1105E+00	-0.3657E-01	-0.6345E+00
0.125E+01	-0.5657E+00	0.1105E+00	-0.3650E-01	-0.6343E+00
0.130E+01	-0.5660E+00	0.1104E+00	-0.3647E-01	-0.6340E+00
0.135E+01	-0.5663E+00	0.1102E+00	-0.3648E-01	-0.6336E+00
0.140E+01	-0.5667E+00	0.1100E+00	-0.3655E-01	-0.6333E+00
0.145E+01	-0.5669E+00	0.1096E+00	-0.3668E-01	-0.6331E+00
0.150E+01	-0.5672E+00	0.1092E+00	-0.3688E-01	-0.6328E+00
0.155E+01	-0.5673E+00	0.1088E+00	-0.3717E-01	-0.6327E+00
0.160E+01	-0.5674E+00	0.1083E+00	-0.3755E-01	-0.6326E+00
0.165E+01	-0.5674E+00	0.1079E+00	-0.3801E-01	-0.6326E+00
0.170E+01	-0.5672E+00	0.1075E+00	-0.3857E-01	-0.6328E+00
0.175E+01	-0.5669E+00	0.1073E+00	-0.3921E-01	-0.6331E+00
0.180E+01	-0.5664E+00	0.1072E+00	-0.3992E-01	-0.6336E+00
0.185E+01	-0.5658E+00	0.1073E+00	-0.4071E-01	-0.6342E+00
0.190E+01	-0.5650E+00	0.1077E+00	-0.4155E-01	-0.6350E+00
0.195E+01	-0.5640E+00	0.1083E+00	-0.4242E-01	-0.6360E+00
0.200E+01	-0.5629E+00	0.1092E+00	-0.4331E-01	-0.6371E+00
0.205E+01	-0.5616E+00	0.1105E+00	-0.4419E-01	-0.6384E+00
0.210E+01	-0.5602E+00	0.1122E+00	-0.4505E-01	-0.6398E+00
0.215E+01	-0.5588E+00	0.1142E+00	-0.4584E-01	-0.6412E+00
0.220E+01	-0.5572E+00	0.1165E+00	-0.4656E-01	-0.6428E+00
0.225E+01	-0.5556E+00	0.1192E+00	-0.4717E-01	-0.6443E+00
0.230E+01	-0.5541E+00	0.1223E+00	-0.4765E-01	-0.6459E+00
0.235E+01	-0.5527E+00	0.1255E+00	-0.4798E-01	-0.6473E+00
0.240E+01	-0.5514E+00	0.1290E+00	-0.4815E-01	-0.6486E+00
0.245E+01	-0.5503E+00	0.1327E+00	-0.4815E-01	-0.6497E+00
0.250E+01	-0.5494E+00	0.1363E+00	-0.4798E-01	-0.6506E+00
0.255E+01	-0.5488E+00	0.1400E+00	-0.4764E-01	-0.6512E+00

Table 5.10

0.260E+01	-0.5485E+00	0.1434E+00	-0.4715E-01	-0.6515E+00
0.265E+01	-0.5486E+00	0.1467E+00	-0.4653E-01	-0.6514E+00
0.270E+01	-0.5491E+00	0.1495E+00	-0.4582E-01	-0.6509E+00
0.275E+01	-0.5499E+00	0.1519E+00	-0.4505E-01	-0.6500E+00
0.280E+01	-0.5511E+00	0.1538E+00	-0.4429E-01	-0.6488E+00
0.285E+01	-0.5527E+00	0.1550E+00	-0.4360E-01	-0.6473E+00
0.290E+01	-0.5545E+00	0.1555E+00	-0.4304E-01	-0.6455E+00
0.295E+01	-0.5566E+00	0.1553E+00	-0.4269E-01	-0.6434E+00
0.300E+01	-0.5587E+00	0.1543E+00	-0.4262E-01	-0.6413E+00
0.305E+01	-0.5609E+00	0.1527E+00	-0.4291E-01	-0.6391E+00
0.310E+01	-0.5630E+00	0.1505E+00	-0.4363E-01	-0.6370E+00
0.315E+01	-0.5648E+00	0.1478E+00	-0.4486E-01	-0.6352E+00
0.320E+01	-0.5662E+00	0.1447E+00	-0.4665E-01	-0.6338E+00
0.325E+01	-0.5670E+00	0.1415E+00	-0.4904E-01	-0.6330E+00
0.330E+01	-0.5672E+00	0.1384E+00	-0.5206E-01	-0.6328E+00
0.335E+01	-0.5665E+00	0.1356E+00	-0.5573E-01	-0.6336E+00
0.340E+01	-0.5648E+00	0.1335E+00	-0.6002E-01	-0.6353E+00
0.345E+01	-0.5619E+00	0.1324E+00	-0.6491E-01	-0.6381E+00
0.350E+01	-0.5580E+00	0.1325E+00	-0.7034E-01	-0.6421E+00
0.355E+01	-0.5527E+00	0.1342E+00	-0.7621E-01	-0.6473E+00
0.360E+01	-0.5462E+00	0.1379E+00	-0.8244E-01	-0.6538E+00
0.365E+01	-0.5385E+00	0.1437E+00	-0.8889E-01	-0.6615E+00
0.370E+01	-0.5296E+00	0.1520E+00	-0.9542E-01	-0.6704E+00
0.375E+01	-0.5197E+00	0.1629E+00	-0.1019E+00	-0.6803E+00
0.380E+01	-0.5089E+00	0.1766E+00	-0.1081E+00	-0.6911E+00
0.385E+01	-0.4974E+00	0.1932E+00	-0.1139E+00	-0.7026E+00
0.390E+01	-0.4854E+00	0.2127E+00	-0.1191E+00	-0.7145E+00
0.395E+01	-0.4733E+00	0.2350E+00	-0.1235E+00	-0.7266E+00
0.400E+01	-0.4614E+00	0.2600E+00	-0.1270E+00	-0.7385E+00
0.405E+01	-0.4500E+00	0.2874E+00	-0.1295E+00	-0.7499E+00
0.410E+01	-0.4396E+00	0.3169E+00	-0.1308E+00	-0.7603E+00
0.415E+01	-0.4305E+00	0.3480E+00	-0.1307E+00	-0.7693E+00
0.420E+01	-0.4232E+00	0.3801E+00	-0.1294E+00	-0.7766E+00
0.425E+01	-0.4182E+00	0.4126E+00	-0.1268E+00	-0.7817E+00
0.430E+01	-0.4157E+00	0.4446E+00	-0.1229E+00	-0.7841E+00
0.435E+01	-0.4162E+00	0.4753E+00	-0.1180E+00	-0.7836E+00
0.440E+01	-0.4200E+00	0.5036E+00	-0.1122E+00	-0.7798E+00
0.445E+01	-0.4273E+00	0.5285E+00	-0.1060E+00	-0.7725E+00
0.450E+01	-0.4381E+00	0.5489E+00	-0.9967E-01	-0.7617E+00
0.455E+01	-0.4524E+00	0.5638E+00	-0.9390E-01	-0.7475E+00
0.460E+01	-0.4698E+00	0.5722E+00	-0.8934E-01	-0.7300E+00
0.465E+01	-0.4899E+00	0.5734E+00	-0.8678E-01	-0.7101E+00
0.470E+01	-0.5117E+00	0.5669E+00	-0.8709E-01	-0.6883E+00
0.475E+01	-0.5341E+00	0.5527E+00	-0.9121E-01	-0.6659E+00
0.480E+01	-0.5559E+00	0.5312E+00	-0.1001E+00	-0.6442E+00
0.485E+01	-0.5753E+00	0.5033E+00	-0.1146E+00	-0.6249E+00
0.490E+01	-0.5905E+00	0.4705E+00	-0.1355E+00	-0.6098E+00
0.495E+01	-0.5995E+00	0.4352E+00	-0.1637E+00	-0.6008E+00
0.500E+01	-0.6004E+00	0.4000E+00	-0.1996E+00	-0.6000E+00

Table 5.10 Optimal Linear Feedback Tensors for Example 3 Indexed by Time

TIME	K2(1)	K2(2)	K2(3)
	K2(4)	K2(5)	K2(6)
----	-----	-----	-----
0.000E+00	0.6744E-01	-0.7752E-01	-0.1763E+00
	-0.1935E-02	0.8026E-01	0.4751E-01
0.500E-01	0.6773E-01	-0.7705E-01	-0.1759E+00
	-0.1916E-02	0.8032E-01	0.4738E-01
0.100E+00	0.6801E-01	-0.7660E-01	-0.1755E+00
	-0.1870E-02	0.8044E-01	0.4729E-01
0.150E+00	0.6827E-01	-0.7620E-01	-0.1751E+00
	-0.1795E-02	0.8065E-01	0.4726E-01
0.200E+00	0.6850E-01	-0.7587E-01	-0.1747E+00
	-0.1691E-02	0.8094E-01	0.4731E-01
0.250E+00	0.6868E-01	-0.7564E-01	-0.1744E+00
	-0.1556E-02	0.8131E-01	0.4745E-01
0.300E+00	0.6881E-01	-0.7554E-01	-0.1742E+00
	-0.1391E-02	0.8178E-01	0.4768E-01
0.350E+00	0.6888E-01	-0.7560E-01	-0.1742E+00
	-0.1198E-02	0.8233E-01	0.4801E-01
0.400E+00	0.6887E-01	-0.7582E-01	-0.1744E+00
	-0.9811E-03	0.8296E-01	0.4844E-01
0.450E+00	0.6879E-01	-0.7624E-01	-0.1747E+00
	-0.7437E-03	0.8367E-01	0.4898E-01
0.500E+00	0.6862E-01	-0.7686E-01	-0.1753E+00
	-0.4916E-03	0.8443E-01	0.4961E-01
0.550E+00	0.6837E-01	-0.7768E-01	-0.1761E+00
	-0.2314E-03	0.8524E-01	0.5033E-01
0.600E+00	0.6803E-01	-0.7871E-01	-0.1772E+00
	0.2925E-04	0.8607E-01	0.5113E-01
0.650E+00	0.6763E-01	-0.7994E-01	-0.1785E+00
	0.2821E-03	0.8690E-01	0.5198E-01
0.700E+00	0.6715E-01	-0.8134E-01	-0.1801E+00
	0.5182E-03	0.8772E-01	0.5286E-01
0.750E+00	0.6663E-01	-0.8289E-01	-0.1820E+00
	0.7286E-03	0.8848E-01	0.5376E-01
0.800E+00	0.6607E-01	-0.8455E-01	-0.1840E+00
	0.9044E-03	0.8918E-01	0.5463E-01
0.850E+00	0.6550E-01	-0.8629E-01	-0.1862E+00
	0.1037E-02	0.8978E-01	0.5546E-01
0.900E+00	0.6494E-01	-0.8806E-01	-0.1885E+00
	0.1119E-02	0.9027E-01	0.5621E-01
0.950E+00	0.6442E-01	-0.8979E-01	-0.1909E+00
	0.1144E-02	0.9062E-01	0.5684E-01
0.100E+01	0.6396E-01	-0.9144E-01	-0.1933E+00
	0.1106E-02	0.9082E-01	0.5734E-01
0.105E+01	0.6360E-01	-0.9293E-01	-0.1956E+00
	0.1004E-02	0.9086E-01	0.5767E-01
0.110E+01	0.6337E-01	-0.9421E-01	-0.1977E+00
	0.8370E-03	0.9073E-01	0.5782E-01
0.115E+01	0.6329E-01	-0.9522E-01	-0.1997E+00
	0.6074E-03	0.9044E-01	0.5777E-01
0.120E+01	0.6340E-01	-0.9591E-01	-0.2013E+00
	0.3210E-03	0.9000E-01	0.5750E-01
0.125E+01	0.6371E-01	-0.9621E-01	-0.2026E+00

Table 5.11

	-0.1304E-04	0.8941E-01	0.5702E-01
0.130E+01	0.6425E-01	-0.9610E-01	-0.2034E+00
	-0.3821E-03	0.8872E-01	0.5633E-01
0.135E+01	0.6502E-01	-0.9555E-01	-0.2038E+00
	-0.7697E-03	0.8794E-01	0.5546E-01
0.140E+01	0.6603E-01	-0.9455E-01	-0.2037E+00
	-0.1156E-02	0.8713E-01	0.5442E-01
0.145E+01	0.6727E-01	-0.9310E-01	-0.2031E+00
	-0.1518E-02	0.8634E-01	0.5326E-01
0.150E+01	0.6873E-01	-0.9123E-01	-0.2020E+00
	-0.1829E-02	0.8563E-01	0.5203E-01
0.155E+01	0.7037E-01	-0.8899E-01	-0.2004E+00
	-0.2060E-02	0.8507E-01	0.5079E-01
0.160E+01	0.7215E-01	-0.8646E-01	-0.1984E+00
	-0.2182E-02	0.8472E-01	0.4959E-01
0.165E+01	0.7403E-01	-0.8373E-01	-0.1960E+00
	-0.2164E-02	0.8467E-01	0.4853E-01
0.170E+01	0.7595E-01	-0.8094E-01	-0.1935E+00
	-0.1975E-02	0.8499E-01	0.4768E-01
0.175E+01	0.7782E-01	-0.7821E-01	-0.1908E+00
	-0.1588E-02	0.8577E-01	0.4713E-01
0.180E+01	0.7958E-01	-0.7573E-01	-0.1883E+00
	-0.9765E-03	0.8707E-01	0.4697E-01
0.185E+01	0.8113E-01	-0.7367E-01	-0.1861E+00
	-0.1215E-03	0.8897E-01	0.4728E-01
0.190E+01	0.8238E-01	-0.7222E-01	-0.1843E+00
	0.9912E-03	0.9152E-01	0.4816E-01
0.195E+01	0.8326E-01	-0.7160E-01	-0.1833E+00
	0.2367E-02	0.9476E-01	0.4967E-01
0.200E+01	0.8366E-01	-0.7200E-01	-0.1832E+00
	0.4004E-02	0.9873E-01	0.5188E-01
0.205E+01	0.8352E-01	-0.7361E-01	-0.1843E+00
	0.5888E-02	0.1034E+00	0.5485E-01
0.210E+01	0.8278E-01	-0.7662E-01	-0.1868E+00
	0.7995E-02	0.1088E+00	0.5861E-01
0.215E+01	0.8139E-01	-0.8118E-01	-0.1909E+00
	0.1029E-01	0.1149E+00	0.6316E-01
0.220E+01	0.7931E-01	-0.8740E-01	-0.1969E+00
	0.1272E-01	0.1216E+00	0.6848E-01
0.225E+01	0.7655E-01	-0.9535E-01	-0.2048E+00
	0.1524E-01	0.1287E+00	0.7455E-01
0.230E+01	0.7314E-01	-0.1051E+00	-0.2148E+00
	0.1776E-01	0.1362E+00	0.8127E-01
0.235E+01	0.6914E-01	-0.1165E+00	-0.2269E+00
	0.2022E-01	0.1439E+00	0.8854E-01
0.240E+01	0.6463E-01	-0.1295E+00	-0.2411E+00
	0.2254E-01	0.1516E+00	0.9621E-01
0.245E+01	0.5973E-01	-0.1440E+00	-0.2574E+00
	0.2461E-01	0.1590E+00	0.1041E+00
0.250E+01	0.5462E-01	-0.1595E+00	-0.2757E+00
	0.2636E-01	0.1660E+00	0.1120E+00
0.255E+01	0.4948E-01	-0.1759E+00	-0.2956E+00
	0.2771E-01	0.1723E+00	0.1197E+00
0.260E+01	0.4453E-01	-0.1926E+00	-0.3169E+00
	0.2856E-01	0.1777E+00	0.1269E+00

Table 5.11

0.265E+01	0.4001E-01	-0.2091E+00	-0.3391E+00
	0.2886E-01	0.1819E+00	0.1333E+00
0.270E+01	0.3618E-01	-0.2248E+00	-0.3618E+00
	0.2856E-01	0.1847E+00	0.1386E+00
0.275E+01	0.3331E-01	-0.2391E+00	-0.3842E+00
	0.2764E-01	0.1859E+00	0.1425E+00
0.280E+01	0.3165E-01	-0.2512E+00	-0.4059E+00
	0.2609E-01	0.1856E+00	0.1448E+00
0.285E+01	0.3145E-01	-0.2606E+00	-0.4259E+00
	0.2395E-01	0.1836E+00	0.1453E+00
0.290E+01	0.3288E-01	-0.2666E+00	-0.4436E+00
	0.2126E-01	0.1799E+00	0.1437E+00
0.295E+01	0.3612E-01	-0.2686E+00	-0.4583E+00
	0.1812E-01	0.1749E+00	0.1401E+00
0.300E+01	0.4125E-01	-0.2662E+00	-0.4691E+00
	0.1464E-01	0.1687E+00	0.1343E+00
0.305E+01	0.4827E-01	-0.2591E+00	-0.4755E+00
	0.1097E-01	0.1617E+00	0.1266E+00
0.310E+01	0.5712E-01	-0.2474E+00	-0.4770E+00
	0.7266E-02	0.1543E+00	0.1172E+00
0.315E+01	0.6765E-01	-0.2313E+00	-0.4734E+00
	0.3683E-02	0.1472E+00	0.1064E+00
0.320E+01	0.7964E-01	-0.2111E+00	-0.4645E+00
	0.3851E-03	0.1409E+00	0.9471E-01
0.325E+01	0.9278E-01	-0.1877E+00	-0.4507E+00
	-0.2484E-02	0.1359E+00	0.8279E-01
0.330E+01	0.1067E+00	-0.1619E+00	-0.4325E+00
	-0.4808E-02	0.1329E+00	0.7129E-01
0.335E+01	0.1211E+00	-0.1349E+00	-0.4107E+00
	-0.6508E-02	0.1323E+00	0.6094E-01
0.340E+01	0.1356E+00	-0.1079E+00	-0.3864E+00
	-0.7551E-02	0.1346E+00	0.5247E-01
0.345E+01	0.1499E+00	-0.8204E-01	-0.3609E+00
	-0.7956E-02	0.1398E+00	0.4657E-01
0.350E+01	0.1635E+00	-0.5849E-01	-0.3358E+00
	-0.7793E-02	0.1480E+00	0.4383E-01
0.355E+01	0.1764E+00	-0.3819E-01	-0.3125E+00
	-0.7183E-02	0.1590E+00	0.4471E-01
0.360E+01	0.1883E+00	-0.2178E-01	-0.2925E+00
	-0.6288E-02	0.1724E+00	0.4947E-01
0.365E+01	0.1993E+00	-0.9561E-02	-0.2773E+00
	-0.5301E-02	0.1875E+00	0.5817E-01
0.370E+01	0.2092E+00	-0.1427E-02	-0.2678E+00
	-0.4431E-02	0.2035E+00	0.7063E-01
0.375E+01	0.2182E+00	0.3152E-02	-0.2647E+00
	-0.3891E-02	0.2195E+00	0.8642E-01
0.380E+01	0.2264E+00	0.5111E-02	-0.2683E+00
	-0.3878E-02	0.2342E+00	0.1049E+00
0.385E+01	0.2340E+00	0.5752E-02	-0.2784E+00
	-0.4555E-02	0.2466E+00	0.1250E+00
0.390E+01	0.2411E+00	0.6650E-02	-0.2942E+00
	-0.6038E-02	0.2556E+00	0.1457E+00
0.395E+01	0.2477E+00	0.9545E-02	-0.3142E+00
	-0.8379E-02	0.2602E+00	0.1658E+00
0.400E+01	0.2539E+00	0.1619E-01	-0.3364E+00

Table 5.11

	-0.1156E-01	0.2596E+00	0.1836E+00
0.405E+01	0.2595E+00	0.2818E-01	-0.3585E+00
	-0.1547E-01	0.2532E+00	0.1978E+00
0.410E+01	0.2643E+00	0.4673E-01	-0.3776E+00
	-0.1993E-01	0.2409E+00	0.2069E+00
0.415E+01	0.2681E+00	0.7249E-01	-0.3904E+00
	-0.2468E-01	0.2229E+00	0.2095E+00
0.420E+01	0.2701E+00	0.1053E+00	-0.3937E+00
	-0.2940E-01	0.2000E+00	0.2045E+00
0.425E+01	0.2700E+00	0.1438E+00	-0.3844E+00
	-0.3375E-01	0.1735E+00	0.1912E+00
0.430E+01	0.2672E+00	0.1858E+00	-0.3598E+00
	-0.3740E-01	0.1453E+00	0.1694E+00
0.435E+01	0.2611E+00	0.2273E+00	-0.3180E+00
	-0.4011E-01	0.1177E+00	0.1397E+00
0.440E+01	0.2518E+00	0.2634E+00	-0.2583E+00
	-0.4179E-01	0.9334E-01	0.1033E+00
0.445E+01	0.2395E+00	0.2879E+00	-0.1816E+00
	-0.4258E-01	0.7475E-01	0.6249E-01
0.450E+01	0.2253E+00	0.2944E+00	-0.9062E-01
	-0.4287E-01	0.6414E-01	0.2017E-01
0.455E+01	0.2113E+00	0.2767E+00	0.9827E-02
	-0.4334E-01	0.6273E-01	-0.2004E-01
0.460E+01	0.2008E+00	0.2300E+00	0.1132E+00
	-0.4483E-01	0.7032E-01	-0.5431E-01
0.465E+01	0.1982E+00	0.1523E+00	0.2116E+00
	-0.4819E-01	0.8487E-01	-0.7916E-01
0.470E+01	0.2089E+00	0.4601E-01	0.2968E+00
	-0.5394E-01	0.1024E+00	-0.9205E-01
0.475E+01	0.2390E+00	-0.8173E-01	0.3611E+00
	-0.6178E-01	0.1172E+00	-0.9210E-01
0.480E+01	0.2941E+00	-0.2183E+00	0.3995E+00
	-0.7013E-01	0.1230E+00	-0.8046E-01
0.485E+01	0.3788E+00	-0.3466E+00	0.4102E+00
	-0.7555E-01	0.1141E+00	-0.6043E-01
0.490E+01	0.4951E+00	-0.4470E+00	0.3954E+00
	-0.7236E-01	0.8779E-01	-0.3698E-01
0.495E+01	0.6417E+00	-0.5004E+00	0.3617E+00
	-0.5245E-01	0.4594E-01	-0.1568E-01
0.500E+01	0.8138E+00	-0.1401E-01	-0.1073E-02
	-0.5226E-02	0.5397E-03	-0.1416E-02

Table 5.11 Optimal Quadratic Feedback Tensors for Example 3

TIME	K3(1)	K3(2)	K3(3)	K3(4)
	K3(5)	K3(6)	K3(7)	K3(8)
----	-----	-----	-----	-----
0.000E+00	-0.4478E-01	-0.1316E-01	0.4101E+00	0.4023E+00
	0.1678E-01	0.7880E-02	-0.1709E+00	-0.9253E-01
0.500E-01	-0.4498E-01	-0.1301E-01	0.4115E+00	0.4031E+00
	0.1686E-01	0.7394E-02	-0.1715E+00	-0.9290E-01
0.100E+00	-0.4529E-01	-0.1307E-01	0.4128E+00	0.4040E+00
	0.1700E-01	0.6971E-02	-0.1721E+00	-0.9329E-01
0.150E+00	-0.4572E-01	-0.1337E-01	0.4139E+00	0.4051E+00
	0.1721E-01	0.6638E-02	-0.1726E+00	-0.9370E-01
0.200E+00	-0.4626E-01	-0.1394E-01	0.4148E+00	0.4062E+00
	0.1750E-01	0.6424E-02	-0.1731E+00	-0.9410E-01
0.250E+00	-0.4693E-01	-0.1480E-01	0.4155E+00	0.4074E+00
	0.1786E-01	0.6357E-02	-0.1736E+00	-0.9448E-01
0.300E+00	-0.4771E-01	-0.1598E-01	0.4158E+00	0.4086E+00
	0.1829E-01	0.6461E-02	-0.1739E+00	-0.9482E-01
0.350E+00	-0.4859E-01	-0.1747E-01	0.4157E+00	0.4098E+00
	0.1878E-01	0.6757E-02	-0.1741E+00	-0.9511E-01
0.400E+00	-0.4956E-01	-0.1927E-01	0.4151E+00	0.4109E+00
	0.1933E-01	0.7263E-02	-0.1741E+00	-0.9532E-01
0.450E+00	-0.5060E-01	-0.2137E-01	0.4141E+00	0.4118E+00
	0.1993E-01	0.7988E-02	-0.1739E+00	-0.9543E-01
0.500E+00	-0.5168E-01	-0.2374E-01	0.4126E+00	0.4126E+00
	0.2055E-01	0.8935E-02	-0.1736E+00	-0.9542E-01
0.550E+00	-0.5277E-01	-0.2634E-01	0.4105E+00	0.4131E+00
	0.2120E-01	0.1010E-01	-0.1730E+00	-0.9528E-01
0.600E+00	-0.5385E-01	-0.2912E-01	0.4079E+00	0.4133E+00
	0.2183E-01	0.1147E-01	-0.1722E+00	-0.9499E-01
0.650E+00	-0.5488E-01	-0.3202E-01	0.4049E+00	0.4131E+00
	0.2244E-01	0.1301E-01	-0.1712E+00	-0.9454E-01
0.700E+00	-0.5582E-01	-0.3496E-01	0.4014E+00	0.4125E+00
	0.2301E-01	0.1470E-01	-0.1700E+00	-0.9393E-01
0.750E+00	-0.5665E-01	-0.3786E-01	0.3976E+00	0.4114E+00
	0.2350E-01	0.1650E-01	-0.1686E+00	-0.9315E-01
0.800E+00	-0.5732E-01	-0.4064E-01	0.3935E+00	0.4098E+00
	0.2390E-01	0.1834E-01	-0.1670E+00	-0.9223E-01
0.850E+00	-0.5780E-01	-0.4319E-01	0.3893E+00	0.4078E+00
	0.2418E-01	0.2018E-01	-0.1653E+00	-0.9117E-01
0.900E+00	-0.5807E-01	-0.4543E-01	0.3852E+00	0.4053E+00
	0.2433E-01	0.2193E-01	-0.1636E+00	-0.9000E-01
0.950E+00	-0.5811E-01	-0.4726E-01	0.3812E+00	0.4023E+00
	0.2433E-01	0.2353E-01	-0.1619E+00	-0.8875E-01
0.100E+01	-0.5789E-01	-0.4860E-01	0.3775E+00	0.3989E+00
	0.2416E-01	0.2491E-01	-0.1602E+00	-0.8747E-01
0.105E+01	-0.5742E-01	-0.4936E-01	0.3744E+00	0.3952E+00
	0.2383E-01	0.2597E-01	-0.1587E+00	-0.8620E-01
0.110E+01	-0.5669E-01	-0.4948E-01	0.3720E+00	0.3914E+00
	0.2332E-01	0.2665E-01	-0.1575E+00	-0.8500E-01
0.115E+01	-0.5573E-01	-0.4892E-01	0.3705E+00	0.3874E+00
	0.2265E-01	0.2687E-01	-0.1565E+00	-0.8391E-01
0.120E+01	-0.5456E-01	-0.4764E-01	0.3700E+00	0.3835E+00
	0.2183E-01	0.2658E-01	-0.1559E+00	-0.8301E-01

Table 5.12

0.125E+01	-0.5322E-01	-0.4365E-01	0.3708E+00	0.3799E+00
	0.2090E-01	0.2572E-01	-0.1559E+00	-0.8236E-01
0.130E+01	-0.5178E-01	-0.4298E-01	0.3729E+00	0.3766E+00
	0.1968E-01	0.2427E-01	-0.1563E+00	-0.8201E-01
0.135E+01	-0.5031E-01	-0.3971E-01	0.3764E+00	0.3739E+00
	0.1881E-01	0.2222E-01	-0.1573E+00	-0.8202E-01
0.140E+01	-0.4889E-01	-0.3593E-01	0.3813E+00	0.3720E+00
	0.1776E-01	0.1961E-01	-0.1590E+00	-0.8244E-01
0.145E+01	-0.4764E-01	-0.3182E-01	0.3877E+00	0.3710E+00
	0.1678E-01	0.1647E-01	-0.1613E+00	-0.8330E-01
0.150E+01	-0.4666E-01	-0.2755E-01	0.3954E+00	0.3711E+00
	0.1592E-01	0.1289E-01	-0.1642E+00	-0.8463E-01
0.155E+01	-0.4607E-01	-0.2337E-01	0.4042E+00	0.3724E+00
	0.1526E-01	0.8988E-02	-0.1677E+00	-0.8643E-01
0.160E+01	-0.4598E-01	-0.1954E-01	0.4140E+00	0.3749E+00
	0.1487E-01	0.4888E-02	-0.1716E+00	-0.8869E-01
0.165E+01	-0.4652E-01	-0.1637E-01	0.4244E+00	0.3788E+00
	0.1478E-01	0.7462E-03	-0.1759E+00	-0.9136E-01
0.170E+01	-0.4779E-01	-0.1418E-01	0.4351E+00	0.3840E+00
	0.1506E-01	-0.3273E-02	-0.1805E+00	-0.9439E-01
0.175E+01	-0.4987E-01	-0.1327E-01	0.4457E+00	0.3906E+00
	0.1574E-01	-0.7007E-02	-0.1852E+00	-0.9770E-01
0.180E+01	-0.5283E-01	-0.1395E-01	0.4556E+00	0.3983E+00
	0.1682E-01	-0.1030E-01	-0.1898E+00	-0.1012E+00
0.185E+01	-0.5668E-01	-0.1647E-01	0.4644E+00	0.4069E+00
	0.1833E-01	-0.1303E-01	-0.1942E+00	-0.1047E+00
0.190E+01	-0.6143E-01	-0.2103E-01	0.4716E+00	0.4164E+00
	0.2022E-01	-0.1508E-01	-0.1983E+00	-0.1082E+00
0.195E+01	-0.6702E-01	-0.2773E-01	0.4767E+00	0.4263E+00
	0.2246E-01	-0.1641E-01	-0.2021E+00	-0.1116E+00
0.200E+01	-0.7336E-01	-0.3660E-01	0.4793E+00	0.4365E+00
	0.2497E-01	-0.1701E-01	-0.2054E+00	-0.1147E+00
0.205E+01	-0.8031E-01	-0.4753E-01	0.4795E+00	0.4465E+00
	0.2769E-01	-0.1690E-01	-0.2084E+00	-0.1174E+00
0.210E+01	-0.8769E-01	-0.6028E-01	0.4768E+00	0.4561E+00
	0.3051E-01	-0.1618E-01	-0.2111E+00	-0.1199E+00
0.215E+01	-0.9528E-01	-0.7449E-01	0.4714E+00	0.4650E+00
	0.3332E-01	-0.1498E-01	-0.2138E+00	-0.1219E+00
0.220E+01	-0.1028E+00	-0.8968E-01	0.4636E+00	0.4729E+00
	0.3599E-01	-0.1346E-01	-0.2166E+00	-0.1238E+00
0.225E+01	-0.1101E+00	-0.1052E+00	0.4537E+00	0.4798E+00
	0.3841E-01	-0.1180E-01	-0.2198E+00	-0.1255E+00
0.230E+01	-0.1167E+00	-0.1204E+00	0.4427E+00	0.4855E+00
	0.4047E-01	-0.1020E-01	-0.2238E+00	-0.1272E+00
0.235E+01	-0.1224E+00	-0.1345E+00	0.4313E+00	0.4903E+00
	0.4203E-01	-0.8855E-02	-0.2289E+00	-0.1292E+00
0.240E+01	-0.1270E+00	-0.1466E+00	0.4208E+00	0.4943E+00
	0.4301E-01	-0.7941E-02	-0.2353E+00	-0.1316E+00
0.245E+01	-0.1301E+00	-0.1560E+00	0.4123E+00	0.4980E+00
	0.4331E-01	-0.7612E-02	-0.2432E+00	-0.1348E+00
0.250E+01	-0.1314E+00	-0.1617E+00	0.4073E+00	0.5019E+00
	0.4287E-01	-0.7987E-02	-0.2528E+00	-0.1389E+00
0.255E+01	-0.1310E+00	-0.1632E+00	0.4071E+00	0.5068E+00
	0.4164E-01	-0.9150E-02	-0.2640E+00	-0.1442E+00

Table 5.12

0.260E+01	-0.1288E+00	-0.1599E+00	0.4129E+00	0.5135E+00
	0.3961E-01	-0.1115E-01	-0.2769E+00	-0.1508E+00
0.265E+01	-0.1239E+00	-0.1514E+00	0.4261E+00	0.5227E+00
	0.3677E-01	-0.1399E-01	-0.2911E+00	-0.1587E+00
0.270E+01	-0.1174E+00	-0.1374E+00	0.4472E+00	0.5353E+00
	0.3316E-01	-0.1766E-01	-0.3062E+00	-0.1678E+00
0.275E+01	-0.1090E+00	-0.1183E+00	0.4769E+00	0.5520E+00
	0.2887E-01	-0.2212E-01	-0.3218E+00	-0.1779E+00
0.280E+01	-0.9907E-01	-0.9439E-01	0.5150E+00	0.5735E+00
	0.2398E-01	-0.2731E-01	-0.3373E+00	-0.1888E+00
0.285E+01	-0.8787E-01	-0.6650E-01	0.5610E+00	0.5999E+00
	0.1864E-01	-0.3317E-01	-0.3521E+00	-0.1999E+00
0.290E+01	-0.7591E-01	-0.3571E-01	0.6136E+00	0.6311E+00
	0.1301E-01	-0.3966E-01	-0.3658E+00	-0.2108E+00
0.295E+01	-0.6375E-01	-0.3430E-02	0.6711E+00	0.6667E+00
	0.7294E-02	-0.4672E-01	-0.3778E+00	-0.2208E+00
0.300E+01	-0.5197E-01	0.2876E-01	0.7312E+00	0.7058E+00
	0.1703E-02	-0.5431E-01	-0.3879E+00	-0.2294E+00
0.305E+01	-0.4122E-01	0.5914E-01	0.7913E+00	0.7469E+00
	-0.3533E-02	-0.6242E-01	-0.3960E+00	-0.2360E+00
0.310E+01	-0.3209E-01	0.8605E-01	0.8486E+00	0.7886E+00
	-0.8177E-02	-0.7102E-01	-0.4023E+00	-0.2404E+00
0.315E+01	-0.2510E-01	0.1080E+00	0.9006E+00	0.8288E+00
	-0.1200E-01	-0.8007E-01	-0.4073E+00	-0.2424E+00
0.320E+01	-0.2067E-01	0.1238E+00	0.9451E+00	0.8659E+00
	-0.1478E-01	-0.8949E-01	-0.4116E+00	-0.2420E+00
0.325E+01	-0.1904E-01	0.1326E+00	0.9807E+00	0.8983E+00
	-0.1634E-01	-0.9913E-01	-0.4161E+00	-0.2396E+00
0.330E+01	-0.2029E-01	0.1343E+00	0.1007E+01	0.9246E+00
	-0.1649E-01	-0.1087E+00	-0.4217E+00	-0.2359E+00
0.335E+01	-0.2430E-01	0.1292E+00	0.1023E+01	0.9449E+00
	-0.1510E-01	-0.1179E+00	-0.4292E+00	-0.2317E+00
0.340E+01	-0.3076E-01	0.1179E+00	0.1032E+01	0.9590E+00
	-0.1209E-01	-0.1262E+00	-0.4392E+00	-0.2279E+00
0.345E+01	-0.3922E-01	0.1016E+00	0.1035E+01	0.9681E+00
	-0.7388E-02	-0.1327E+00	-0.4520E+00	-0.2256E+00
0.350E+01	-0.4910E-01	0.8160E-01	0.1034E+01	0.9737E+00
	-0.9923E-03	-0.1368E+00	-0.4672E+00	-0.2256E+00
0.355E+01	-0.5976E-01	0.5918E-01	0.1032E+01	0.9778E+00
	0.7067E-02	-0.1373E+00	-0.4840E+00	-0.2284E+00
0.360E+01	-0.7049E-01	0.3562E-01	0.1030E+01	0.9826E+00
	0.1671E-01	-0.1333E+00	-0.5009E+00	-0.2344E+00
0.365E+01	-0.8064E-01	0.1194E-01	0.1028E+01	0.9898E+00
	0.2780E-01	-0.1237E+00	-0.5160E+00	-0.2432E+00
0.370E+01	-0.8955E-01	-0.1102E-01	0.1028E+01	0.1000E+01
	0.4015E-01	-0.1077E+00	-0.5268E+00	-0.2542E+00
0.375E+01	-0.9666E-01	-0.3261E-01	0.1026E+01	0.1014E+01
	0.5351E-01	-0.8458E-01	-0.5307E+00	-0.2659E+00
0.380E+01	-0.1015E+00	-0.5225E-01	0.1021E+01	0.1030E+01
	0.6759E-01	-0.5400E-01	-0.5251E+00	-0.2767E+00
0.385E+01	-0.1036E+00	-0.6931E-01	0.1010E+01	0.1045E+01
	0.8201E-01	-0.1607E-01	-0.5078E+00	-0.2844E+00
0.390E+01	-0.1029E+00	-0.8291E-01	0.9889E+00	0.1055E+01
	0.9638E-01	0.2864E-01	-0.4769E+00	-0.2865E+00

Table 5.12

ORIGINAL PAGE IS
OF POOR QUALITY

0.395E+01	-0.9918E-01	-0.9189E-01	0.9555E+00	0.1056E+01
	0.1102E+00	0.7899E-01	-0.4315E+00	-0.2805E+00
0.400E+01	-0.9254E-01	-0.9472E-01	0.9087E+00	0.1041E+01
	0.1231E+00	0.1333E+00	-0.3720E+00	-0.2640E+00
0.405E+01	-0.8328E-01	-0.8964E-01	0.8476E+00	0.1005E+01
	0.1345E+00	0.1893E+00	-0.3003E+00	-0.2351E+00
0.410E+01	-0.7194E-01	-0.7489E-01	0.7822E+00	0.9439E+00
	0.1439E+00	0.2445E+00	-0.2196E+00	-0.1925E+00
0.415E+01	-0.5931E-01	-0.4919E-01	0.7133E+00	0.8539E+00
	0.1511E+00	0.2961E+00	-0.1350E+00	-0.1361E+00
0.420E+01	-0.4645E-01	-0.1233E-01	0.6528E+00	0.7349E+00
	0.1558E+00	0.3416E+00	-0.5331E-01	-0.6711E-01
0.425E+01	-0.3460E-01	0.3403E-01	0.6126E+00	0.5904E+00
	0.1577E+00	0.3791E+00	0.1815E-01	0.1118E-01
0.430E+01	-0.2511E-01	0.8565E-01	0.6046E+00	0.4280E+00
	0.1571E+00	0.4077E+00	0.7190E-01	0.9374E-01
0.435E+01	-0.1912E-01	0.1352E+00	0.6385E+00	0.2599E+00
	0.1539E+00	0.4277E+00	0.1019E+00	0.1737E+00
0.440E+01	-0.1726E-01	0.1725E+00	0.7178E+00	0.1018E+00
	0.1483E+00	0.4408E+00	0.1053E+00	0.2432E+00
0.445E+01	-0.1924E-01	0.1857E+00	0.8364E+00	-0.2848E-01
	0.1404E+00	0.4497E+00	0.8366E-01	0.2941E+00
0.450E+01	-0.2350E-01	0.1642E+00	0.9758E+00	-0.1139E+00
	0.1302E+00	0.4566E+00	0.4394E-01	0.3202E+00
0.455E+01	-0.2713E-01	0.1032E+00	0.1105E+01	-0.1421E+00
	0.1178E+00	0.4623E+00	-0.2266E-02	0.3183E+00
0.460E+01	-0.2653E-01	0.8851E-02	0.1182E+01	-0.1093E+00
	0.1034E+00	0.4644E+00	-0.4133E-01	0.2901E+00
0.465E+01	-0.1298E-01	-0.9695E-01	0.1167E+01	-0.2325E-01
	0.8825E-01	0.4576E+00	-0.6189E-01	0.2421E+00
0.470E+01	-0.5313E-02	-0.1754E+00	0.1030E+01	0.9733E-01
	0.7505E-01	0.4349E+00	-0.5966E-01	0.1844E+00
0.475E+01	0.6521E-02	-0.1759E+00	0.7696E+00	0.2257E+00
	0.6818E-01	0.3925E+00	-0.4042E-01	0.1279E+00
0.480E+01	-0.1852E-02	-0.4922E-01	0.4183E+00	0.3345E+00
	0.7302E-01	0.3352E+00	-0.1860E-01	0.8112E-01
0.485E+01	-0.6067E-01	0.2322E+00	0.4221E-01	0.4039E+00
	0.9335E-01	0.2830E+00	-0.1002E-01	0.4707E-01
0.490E+01	-0.2097E+00	0.6534E+00	-0.2761E+00	0.4279E+00
	0.1266E+00	0.2704E+00	-0.2016E-01	0.2387E-01
0.495E+01	-0.4910E+00	0.1144E+01	-0.4666E+00	0.4151E+00
	0.1567E+00	0.3381E+00	-0.3296E-01	0.7890E-02
0.500E+01	-0.9371E+00	0.1583E+01	-0.1212E+00	0.1270E+00
	0.1463E+00	0.5110E+00	-0.7517E-02	-0.4125E-03

Table 5.12 Optimal Third Order Feedback Tensors for Example 3

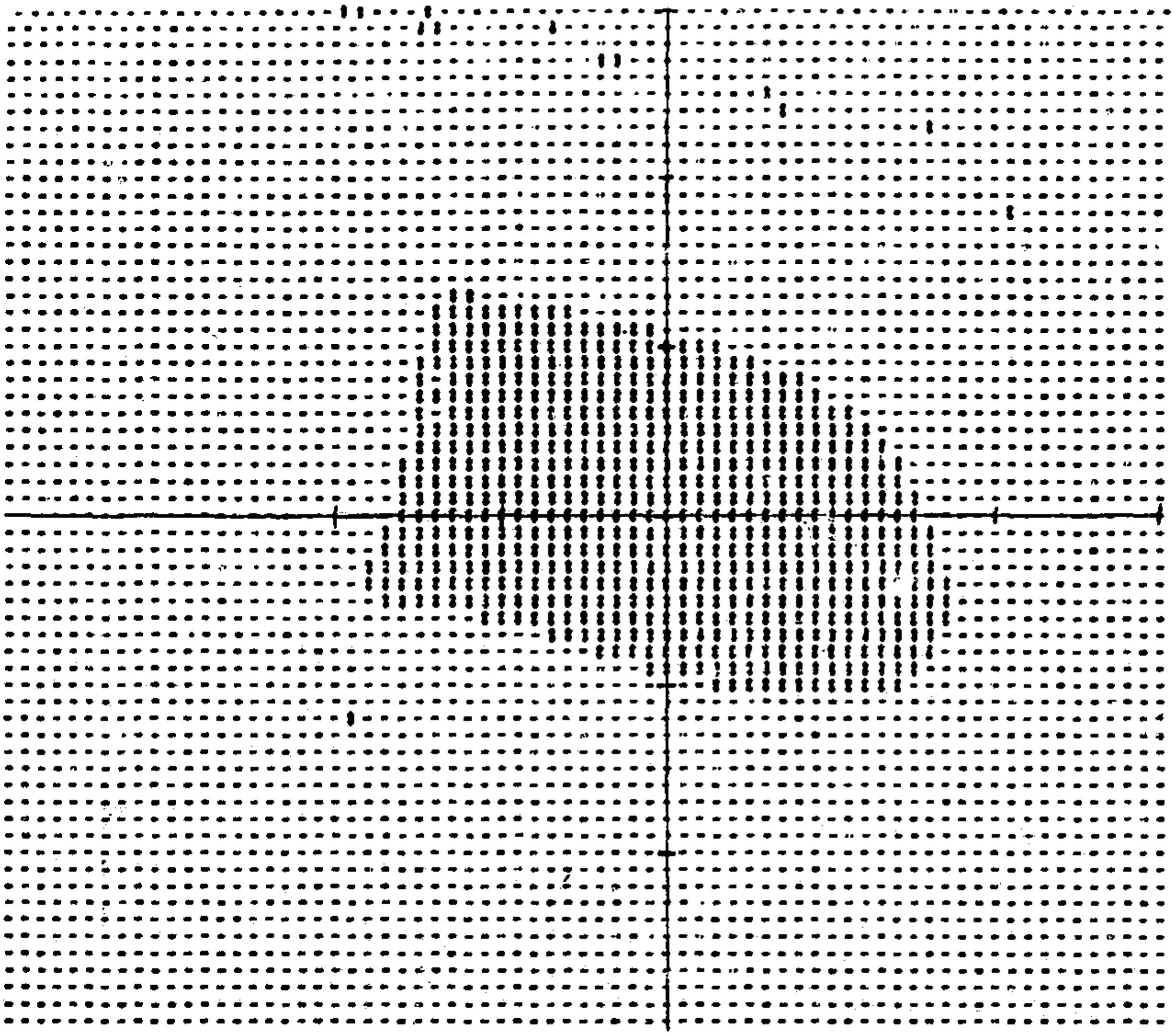


Figure 5.9 The Linear Feedback Region of Usefulness for Example 3. The hash marks show divisions of 1.0 units. There is 0.1 spacing between points. The asteriks represent the acceptable initial conditions.

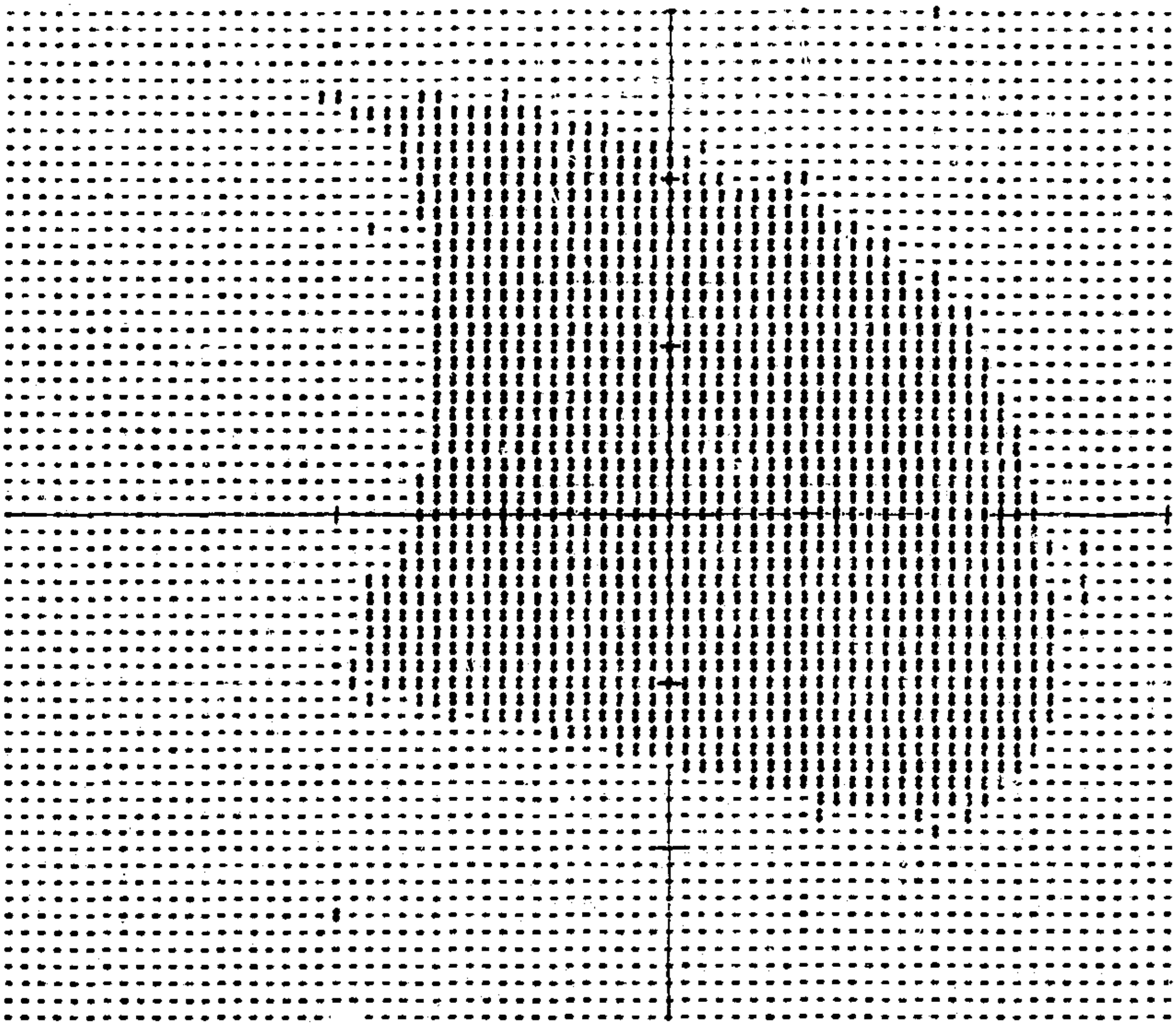


Figure 5.10 The Quadratic Region of Usefulness for Example 3.
The dimensions are the same as those in Figure 5.9.

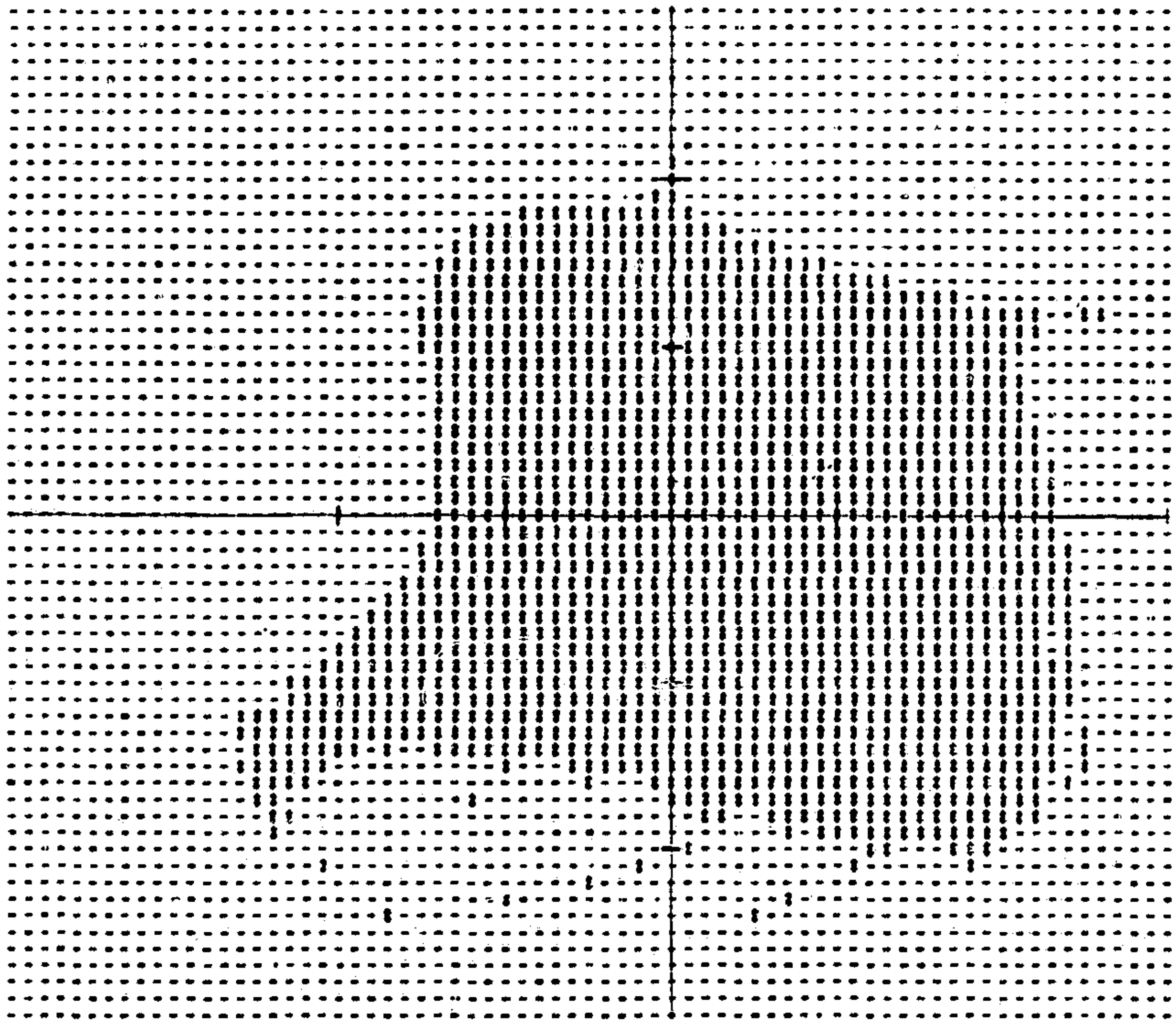


Figure 5.11 The Third Order Region of Usefulness for Example 3. The dimensions are the same as those in Figure 5.9.

CHAPTER VI

CONCLUSIONS

This thesis explains research on nonlinear optimal control performed over the last two years. The main accomplishment of this research is a software package which finds the solution to the nonlinear optimal control problem when the plant cost functional admits a polynomial expansion.

There are several concepts which must be understood before the methods the software uses can be fully appreciated. First, an understanding of tensor algebra is necessary. The second and third chapters of this thesis discuss some topics in tensor algebra. Some of the tensor functions which are developed are very problem specific. The permutation of covariant powers, the raising and lowering of covariant and contravariant powers, and the transposition operation all aid greatly in the manipulation of tensors in complicated expressions. Such functions are generally not found in the literature in the form in which they appear here. Greub [11,12] talks about permutation operators on tensor spaces $T_p(V)$, but not on spaces $T_{p,q}(U,V)$. Similarly, Greub refers to dual tensors in $T_p(V)$ for tensors in $T^p(V)$, but not duals of tensors in $T^{p,q}(U,V)$. This notion of dual tensors is generalized here to encompass the raising and lowering of powers. Also, transposition is just the simultaneous lowering of one contravariant power and raising of one covariant power.

Other tensor functions used here are more standard in the literature. The tensor product, tensor contraction, and tensor symmetrization are all discussed by Greub. Two more basic tensor functions which are necessary to define tensor algebra as an algebra are scalar multiplication of a tensor and tensor addition. There are subroutines to accomplish these functions, but they are so basic they were not discussed in Chapter III. The tensor product is also a basic function, but its action is more complicated than scalar multiplication and tensor addition. Indeed, with

$$T(V) = \sum_{i=0,j=0}^{\infty} T_j^i(V), \text{ where the sum is direct and } T_0^0(V) = R,$$

T(V) is a graded algebra with the tensor product used to combine elements of T(V) to get other elements of T(V), and with both scalar multiplication and the tensor product distributing over addition.

Tensor contraction was generalized here from the description in Greub. Greub describes a contraction operator which acts on only one contravariant and one covariant power at a time. The contraction here operates on an arbitrary number of powers at a time from arbitrary spaces.

The symmetrization here was developed similar to that in Greub. The main difference here is in the ability to symmetrize an arbitrary number of covariant or contravariant powers and leave some powers unsymmetrized. The software was developed with this generality.

The tensor functions just described all followed Buric [1] in both notation and the functions with one exception. The contravariant symmetrization is defined in a different manner than the covariant symmetrization in this thesis. This is done to ensure that the contractions performed in the symmetric space yield the same results as in the normal tensor space.

The next concept which must be understood is nonlinear optimal control theory. This was discussed briefly at the beginning of Chapter IV. Classic nonlinear control theory derives conditions which the optimal cost functional, V(x,t), and the optimal feedback control, u*(x,t), must solve. These conditions are that the Hamilton-Jacobi-Bellman (HJB) equation and the partial derivative of the HJB equation with respect to the control $u(t)$ must vanish. When the system

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

and the cost functional

$$J = M(x(t_1)) + \int_{t_o}^{t_1} L(x,u,t)dt$$

admit a polynomial expansion, then tensor algebra is a natural setting in which to imbed the problem. Furthermore, if the linear truncation of f is stabilizable, M and L begin with quadratic powers of x and u, $f(0,0,t) = 0$, and the set of admissible controllers is

$$\Omega = \{u(x,t) | u(x,t) = \sum_{m=1} K^{1}_{[m]}(t) \overset{m}{\odot} x^{[m]}\}$$

then Buric showed that the above conditions were sufficient as well and that

$$V(x,t) = \sum_{k=2} V_{[k]} \overset{k}{\circ} x^{[k]}.$$

The $V_{[k]}$ and $K^{1}_{[m]}$ solve the explicit equations derived in Chapter IV.

After these concepts are assimilated, the software can be dissected in a straightforward manner. The appendix describes the software and shows how the different subroutines interrelate. It should be noted that TLIB, the tensor subroutine library contains all of the function subroutines. These subroutines are used primarily by TNSCLC and XCALC. TNSCLC calculates the feedback tensors given the system tensors and the parameter file PARAM. XCALC uses these feedback tensors to calculate the state trajectories, given the system and PARAM. PARAM is the set of parameters, generated by the program PARGEN, which describe the sizes of various tensors and the time variables for the problem.

The program STAB calls the subroutine XCALC many times in an exhaustive search of the space of possible initial conditions in order to determine the set of initial conditions which are acceptable. These "regions of usefulness" are shown in Chapter V for two of the example problems.

Chapter V contains three examples. These examples show four main results. First of all the Lukes example verifies the software in the sense that the program calculated the same feedback tensors for linear and quadratic feedback as Lukes calculated. Second, since the problems were of different sizes they served as a demonstration of the flexibility of the software. Third, the second and third examples showed that the third order feedback had a larger region of usefulness than linear or quadratic feedback. Finally, the third example showed for one case the compatibility of this technique with an identified model.

There are several promising areas here for future research. First, now that there is a general package set up for calculating nonlinear feedback tensors, a search for more examples to demonstrate the usefulness of the higher order feedback terms should be carried out. The flexibility of the program is ideal for changing problems or constraints in a relatively short period of time. This search should include changing the cost functional to be minimized in order to achieve design goals. Once more examples have been found, software for calculating fourth and fifth order feedback tensors should be implemented. After those feedback tensors can be calculated another search for examples should be carried out to demonstrate the usefulness of those terms.

Another area for which this thesis is a preliminary study is the mixing of identified models with the calculations of feedback tensors. It must be determined whether a model which was chosen to outperform models of like degree can be used in an algorithm based upon using the Taylor series truncation of the actual system. This can be viewed as a robustness problem--can nonlinear feedback perform satisfactorily in the face of modeling errors. This is an area which must be studied before the feedback can be used in practice.

Another area for potential research is the scheduling of nonlinear controls. The problem envisioned is that of controlling a system which has a series of nonlinear models scheduled over an operating line. Each of these models is only locally valid. A nonlinear controller for each of these models would be calculated. Then, when transitioning from the region where one model is valid to the region where the next model is valid, the controller would be changed smoothly. This problem involves checking that the regions of usefulness overlap sufficiently from one model to the next to ensure stability. Other questions such as how nonlinear scheduling should be viewed theoretically also must be entertained. Applications for this research would include flight controls. In particular, solving both the question of compatibility with models and the problem of scheduling would give the ability to use simulations such as HYTESS [13] to design controllers for actual flight systems.

APPENDIX A

This appendix contains the software described in the preceding chapters of this thesis. This software is written in FORTRAN IV-plus on a PDP-11/44.

A flow chart for a representative problem is shown in Figure A.1. The user of the software must calculate the system tensors and the cost functional tensors. These tensors are input into TNSCLC. The program PARGEN is run first. This program accepts as input the number of states, the number of controls, the initial time, the final time, the integration stepsize, and the number of integration steps between stored values of the controller tensors in TNSCLC. PARGEN creates the file PARAM, which contains the set of parameters used by TNSCLC and XCALC to dimension all of the tensors, to perform functions on the tensors, to calculate the feedback tensors over time, to store and retrieve these tensors, and to calculate the state trajectories over time.

TNSCLC is run next. TNSCLC uses the input system and cost functional tensors and PARAM, to calculate $V_{[2]}$, $V_{[3]}$, $V_{[4]}$, K^1_1, $K^1_{[2]}$, and $K^1_{[3]}$ from the equations derived in Chapter IV. The values for each of these tensors is stored over time in data files. TNSCLC uses the subroutines in TLIB, the library of tensor subroutines, to perform the necessary calculations. The list of these subroutines is shown in Figure A.2.

After TNSCLC is run, XCALC may be run to view the behavior of the system with the feedback. XCALC uses PARAM and the actual system equations to integrate the states when the controller tensors are used. This program may be run with as many different sets of initial conditions as desired. If the behavior of the system is acceptable the program STAB may be run.

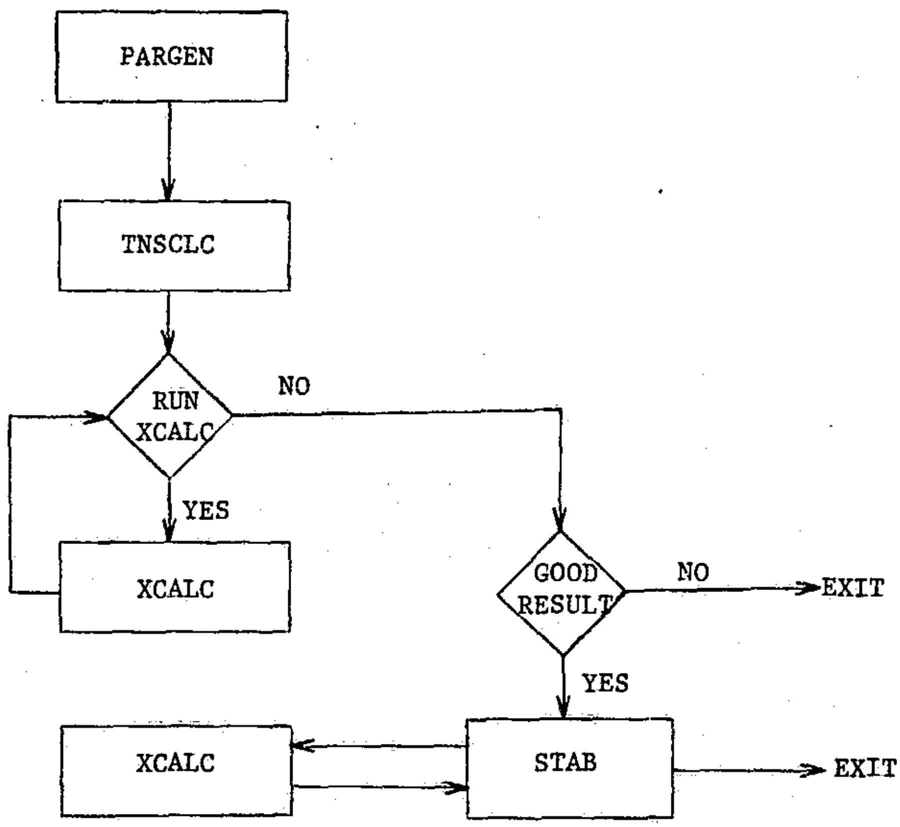


Figure A.1 Flow Chart for One Problem

TLIB

RUNGE	SMULT
TCONT	TMULT
TCONT1	TADD
TRANS	DPROD
PERM	DELTA
RAISE	

Figure A.2. The List of Subroutines in TLIB.

TLIB is used by TNSCLC and XCALC to perform all of the tensor functions and integrations.

STAB calculates the regions of usefulness discussed in Chapter V, by repeatedly calling XCALC with different initial conditions. The program STAB outputs the regions shown in Chapter V. Printouts of each of the subroutines in TLIB and each of the programs discussed above are contained in the Appendix.

```
C************************************************************************
C************************************************************************
C************************************************************************
C***                                                                 ***
C***             TITLE:  PARGEN                                      ***
C***             AUTHOR:  JOSEPH A. O'SULLIVAN                       ***
C***                                                                 ***
C***             THIS PROGRAM GENERATES THE PARAMETERS USED IN CALCULA- ***
C***      TING THE NONLINEAR FEEDBACK TENSOR COEFFICIENTS IN THE SOLUTION ***
C***      TO THE NONLINEAR OPTIMAL CONTROL PROBLEM.  THESE PARAMETERS ARE ***
C***      USED TO DETERMINE THE DIMENSIONS OF THE DIFFERENT TENSORS IN   ***
C***      BOTH SYMMETRIC AND UNSYMMETRIC FORMS.  THIS PROGRAM ALSO FEEDS ***
C***      INFORMATION ABOUT THE NUMBER OF INTEGRATION STEPS AND THE NUMBER***
C***      OF STORED VALUES TO THE MAIN PROGRAM.  PROGRAMS WHICH USE THE  ***
C***      RESULTS OF THIS PRGRAM DO SO BY HAVING THE FIRST LINE OF THEIR ***
C***      CODE BE                                                    ***
C***             INCLUDE 'PARAM1.FTN'                                ***
C***      AFTER THIS LINE, THE PARAMETERS CAN BE USED AS IF THEY WERE THE ***
C***      NUMBERS THEMSELVES.  THIS IS MOST USEFUL IN PROGRAMS WHICH ONE ***
C***      WISHES TO USE DIFFERENT TIMES WITH DIFFERENT SIZE TENSORS.     ***
C***      ALL ONE HAS TO DO IS RUN THIS PROGRAM THEN RECOMPILE THE ORIGI- ***
C***      NAL PROGRAM.  THUS IF ONE CONSISTENTLY USES THE PARAMETERS IN  ***
C***      THE ORIGINAL PROGRAM INSTEAD OF THEIR NUMERIC VALUES, THEN THE ***
C***      PROGRAM WILL NOT HAVE TO BE CHANGED LINE BY LINE.  SO, WHILE   ***
C***      THE TIME NECESSARY TO RECOMPILE THE PROGRAM WILL HAVE TO BE    ***
C***      SPENT, THE ARDUOUS TASK OF COMPLETELY REPROGRAMMING A WORKING  ***
C***      PROGRAM WILL NOT BE REQUIRED.                              ***
C***                                                                 ***
C***      IMPORTANT VARIABLES:                                       ***
C***             IDIMX:  THE DIMENSION OF THE INDEPENDENT VARIABLE   ***
C***             IDIMU:  THE DIMENSION OF THE CONTROL VARIABLE       ***
C***             I2,I3,I4,I5,I6,...:  IDIMX RAISED TO THE APPROPRIATE ***
C***                    POWER                                        ***
C***             I11,I20,I12,I21,I30,I13,I22,I31,I40,...:  IDIMU RAISED ***
C***                    TO THE POWER OF THE FIRST DIGIT TIMES IDIMX  ***
C***                    RAISED TO THE POWER OF THE SECOND DIGIT
C***             IS2,IS3,IS4,...,IS20,IS12,IS21,IS30,IS13,...:  THE  ***
C***                    DIMENSION OF A TENSOR WHICH IS THE SYMMETRIC ***
C***                    VERSION OF A TENSOR WHOSE DIMENSION IS THE SAME ***
C***                    AS THIS PARAMETER WITHOUT THE 'S'
C***      E.G.,                                                      ***
C***             I4=IDIMX**4                                         ***
C***             I13=(IDIMU**1)*(IDIMX**3)                           ***
C***             IS32=IS30*IS2=(IDIMU+2)*(IDIMU+1)*IDIMU*(IDIMX+1)   ***
C***                    *IDIMX/12                                    ***
C***             IS32 IS THE DIMENSION OF THE SYMMETRIZED VERSION OF I32 ***
C***                                                                 ***
C***             T0:  THE STARTING TIME FOR THE INTEGRATIION         ***
C***             T1:  THE FINAL TIME FOR THE INTEGRATION             ***
C***             DT:  THE INTEGRATION STEPSIZE                       ***
C***             IIT:  THE TOTAL NUMBER OF INTEGRATION STEPS         ***
C***             J11:  THE NUMBER OF STEPS BETWEEN STORED VALUES     ***
C***             IIIT:  THE NUMBER OF SECONDARY INTEGRATION STEPS    ***
C***             I0=I2                                               ***
```

```
C***            IQ1D1=I11                                          ***
C***            IR=I20                                             ***
C***            IS112=IDIMX*IDIMU*IS2:  THE DIMENSION OF A SYMMETRIC ***
C***                    TENSOR WHICH HAS ONE CONTRAVARIANT POWER OF ***
C***                    THE DEPENDENT VARIABLE, ONE COVARIANT POWER OF ***
C***                    THE CONTROL VARIABLE, AND TWO COVARIANT POWERS ***
C***                    OF THE INDEPENDENT VARIABLE                ***
C***            IS102=IDIMX*IS2                                    ***
C***                                                               ***
C***    INPUT:                                                     ***
C***            IDIMX, IDIMU, TO, T1, DT, J11 ARE ACCEPTED AS INPUT ***
C***            VARIABLES BY THIS PROGRAM.  FROM THEN ALL OTHER VARI- ***
C***            ABLES CALCULATED.                                  ***
C***                                                               ***
C***    OUPUT:                                                     ***
C***            THE OUTPUT IS INTO 'PARAM1.FTN', A FILE WHICH CONSISTS ***
C***            OF ONE PARAMETER STATEMENT WHICH IS ABOUT FIFTEEN LINES ***
C***            LONG.  THUS IF ONE USES THIS FILE VIA AN INCLUDE STATE- ***
C***            MENT, ONE MUST USE THE '/CO:15' SWITCH WHEN THE PROGRAM ***
C***            IS COMPILED.  E.G., SAY THE FIRST LINE OF 'MAIN.FTN' IS ***
C***                                                               ***
C***            INCLUDE 'PARAM1.FTN'                               ***
C***                                                               ***
C***            THEN THE COMPILATION LINE MUST BE                  ***
C***                                                               ***
C***            >FOR MAIN,MAIN=MAIN/CO:15                          ***
C***                                                               ***
C***            OR ELSE THE PROGRAM WILL NOT COMPILE.              ***
C***                                                               ***
C**********************************************************************
C**********************************************************************
C**********************************************************************
        TYPE 1
1       FORMAT(' ENTER THE DIMENSION OF THE X VARIABLE')
        ACCEPT *, IDIMX
        TYPE 20
20      FORMAT(' ENTER THE DIMENSION OF THE U VARIABLE')
        ACCEPT *, IDIMU
        IQ=IDIMX**2
        I2=IQ
        IQ1D1=IDIMX*IDIMU
        I11=IQ1D1
        IR=IDIMU**2
        I20=IR
        I3=IDIMX**3
        I12=IDIMU*IDIMX**2
        I21=IDIMX*IDIMU**2
        I30=IDIMU**3
        I4=IDIMX**4
        I13=IDIMU*IDIMX**3
        I22=IDIMU**2*IDIMX**2
        I31=IDIMX*IDIMU**3
        I40=IDIMU**4
C***
C***    THE SYMMETRIC PARAMETERS ARE CALCULATED USING COMINATORIAL THEORY.
```

```
C***     FOR EXAMPLE, IF ONE HAS TWO DIGITS, EACH OF WHICH CAN RANGE FROM
C***     ZERO TO IDIMX, AND THE ORDER OF THE DIGITS DOESN'T MATTER, THEN
C***     THE NUMBER OF POSSIBLE PAIRS OF DIGITS IS
C***                        IS2=(IDIMX+1)*IDIMX/2.
C***
         IS2=(IDIMX+1)*IDIMX/2
         IS102=IDIMX*IS2
         IS3=(IDIMX+2)*(IDIMX+1)*IDIMX/6
         IS11=IQ1D1
         IS20=(IDIMU+1)*IDIMU/2
         IS30=(IDIMU+2)*(IDIMU+1)*IDIMU/6
         IS12=IDIMU*IS2
         IS21=IDIMX*IS20
         IS4=(IDIMX+3)*IDIMX*(IDIMX+2)*(IDIMX+1)/24
         IS13=IDIMU*IS3
         IS22=IS2*IS20
         IS31=IS30*IDIMX
         IS40=(IDIMU+3)*(IDIMU+2)*(IDIMU+1)*IDIMU/24
         I5=IDIMX**5
         I14=IDIMU*I4
         I23=IR*I3
         I32=I30*I2
         I41=IDIMX*I40
         I50=I40*IDIMU
         IS5=IS4*(IDIMX+4)/5
         IS14=IDIMU*IS4
         IS23=IS20*IS3
         IS32=IS30*IS2
         IS41=IS40*IDIMX
         IS50=IS40*(IDIMU+4)/5
         I6=IDIMX**6
         I15=IDIMU*I5
         I24=IR*I4
         I33=I30*I3
         I42=I40*IQ
         I51=I50*IDIMX
         I60=IDIMU**6
         IS6=IS5*(IDIMX+5)/6
         IS15=IS5*IDIMU
         IS24=IS20*IS4
         IS33=IS30*IS3
         IS42=IS40*IS2
         IS51=IS50*IDIMX
         IS60=IS50*(IDIMU+5)/6
C***
C***     AFTER CALCULATING ALL OF THE DESIRED PARAMETERS WHICH DEPEND ONLY
C***     UPON IDIMX AND IDIMU, THE OUTPUT FILE IS SET UP AND THEN THE INFOR-
C***     MATION ABOUT THE INTEGRATION STEPS IS OBTAINED.
C***
         OPEN (UNIT=1,TYPE='NEW',NAME='PARAM1.FTN')
         TYPE *,' ENTER T0,T1,DT,'
         ACCEPT *,T0,T1,DT
         IIT=IFIX((T1-T0)/DT+0.0001)+1
         TYPE *,' ENTER THE NUMBER OF STEPS BETWEEN STORED VALUES.'
         ACCEPT *,J11
```

```
C***
C***    THE NEXT VARIABLE CALCULATED, IIIT, IS USED AS THE NUMBER OF
C***    INTEGRATION STEPS IN THE SECONDARY INTEGRATION.  TO BE USED, THE
C***    MAIN PROGRAM SHOULD ONLY SAVE EVERY J11 VALUES FROM THE PRIMARY
C***    INTEGRATION.  IIIT VALUES WILL THEN HAVE BEEN SAVED (ASSUMING
C***    THAT THE FIRST AND LAST VALUES HAVE BEEN SAVED).  THUS IF THE
C***    FIRST INTEGRATION IS DONE BACKWARDS IN TIME USING IIT INTEGRATION
C***    STEPS AND SAVING EVERY J11 VALUES OF THE DESIRED TENSORS, THEN
C***    BY INTEGRATING FORWARD IN TIME AND USING IIIT INTEGRATION STEPS,
C***    WITH A STEPSIZE OF (T1-T0)/IIIT, ONE WILL GET BACK TO T1 AND USE
C***    ALL OF THE SAVED TENSOR VALUES.
C***
        IIIT=IFIX(FLOAT(IIT-1)/FLOAT(J11)+0.0001)+1
        WRITE(1,10)IQ,I2,I0101,I11,IR,I20,I3,I12,I21,I30,I4,I13,I22,
     & I31,I40,I5,I14,I23,I32,I41,I50,I6,I15,I24,I33,I42,I51,I60,IS2,
     & IS11,IS20,IS3,IS12,IS21,IS30,IS4,IS13,IS22,IS31,IS40,IS5,IS14,
     & IS23,IS32,IS41,IS50,IS6,IS15,IS24,IS33,IS42,IS51,IS60,IS102,IIT,
     & IDINX,IDINU,T0,T1,DT,IIIT,J11
        CLOSE (UNIT=1)
        STOP
C***
C***    THE FOLLOWING IS THE FORMAT STATEMENT WHICH ENSURES THAT THE
C***    PARAMETERS ARE STORED IN 'PARAM1.FTN' IN ONE PROPER PARAMETER
C***    STATEMENT.  EVERY NEW LINE MUST HAVE A CHARACTER IN THE SIXTH
C***    COLUMN TO BE A CONTINUATION OF THE LINE PRECEDING IT.  IT SHOULD
C***    BE NOTED THAT IF THE DIMENSIONS GET TO BE SO LARGE THAT IDINX
C***    RAISED TO THE POWER OF SIX IS MORE THAN FIVE DIGITS LONG, AN
C***    OUTPUT ERROR WILL OCCUR.  ALSO, IF THE PARAMETERS GET TO BE TOO
C***    LARGE, THEN THE TENSORS WHICH USE THESE PARAMETERS FOR THEIR
C***    DIMENSIONS MAY GET TOO LARGE TO BE USEFUL.
C***
10      FORMAT(6X,'PARAMETER IQ=',I4,',I2=',I4,',I01D1=',I4,',I11=',I4/
     &  5X,'& ,IR=',I4,',I20=',I4,',I3=',I4,',I12=',I4,',I21=',I4,','/
     &  5X,'& I30=',I5,',I4=',I5,',I13=',I5,',I22=',I5,',I31=',I5,','/
     &  5X,'& I40=',I5,',I5=',I5,',I14=',I5,',I23=',I5,',I32=',I5,','/
     &  5X,'& I41=',I5,',I50=',I5,',I6=',I5,',I15=',I5,',I24=',I5,','/
     &  5X,'& I33=',I5,',I42=',I5,',I51=',I5,',I60=',I5,',IS2=',I5,','/
     &  5X,'& IS11=',I5,',IS20=',I5,',IS3=',I5,',IS12=',I5,',IS21=',I5/
     &  5X,'& ,IS30=',I5,',IS4=',I5,',IS13=',I5,',IS22=',I5,',IS31=',I5,
     &  ','/5X,'& IS40=',I5,',IS5=',I5,',IS14=',I5,',IS23=',I5,','/
     &  5X,'& IS32=',I5,',IS41=',I5,',IS50=',I5,',IS6=',I5,',IS15=',I5,/
     &  5X,'* ,IS24=',I5,',IS33=',I5,',IS42=',I5,',IS51=',I5,','/
     &  5X,'& IS60=',I5,',IS102=',I5,',IIT=',I5,',IDINX=',I5,','
     &  /5X,'& IDINU=',I5,',T0=',F13.6,',T1=',F13.6,',DT=',F13.6,
     &  /5X,'& ,IIIT=',I5,',J11=',I5)
        END
```

```
C*****************************************************************************
C*****************************************************************************
C*****************************************************************************
C***                                                                       ***
C***            TITLE:  TNSCLC                                             ***
C***            AUTHOR:  JOSEPH A. O'SULLIVAN                              ***
C***                                                                       ***
C***            SUBROUTINE TNSCLC CALCULATES THE OPTIMAL FEEDBACK          ***
C***    TENSORS.                                                           ***
C***            THIS SUBROUTINE USES THE FILE 'PARAM1.FTN' WHICH IS        ***
C***    GENERATED BY 'DK2:PARGEN.TSK'.  THIS FILE CONTAINS ALL OF THE      ***
C***    PARAMETERS USED FOR DIMENSIONS AND FOR THE INTEGRATIONS.           ***
C***    THIS SUBROUTINE ALSO USES THE SUBROUTINES IN 'TLIB' THE LIBRARY   ***
C***    OF TENSOR FUNCTIONS.                                               ***
C***                                                                       ***
C***    IMPORTANT VARIABLES:                                               ***
C***            A, B, A20, A11, A02, A30, A21, A12, A03,...:  THE          ***
C***                    COEFFICIENTS OF THE POWERS OF THE CONTROL AND      ***
C***                    STATE VARIABLES IN THE DIFFERENTIAL EQUATION       ***
C***                    DESCRIBING THE SYSTEM.  IF DX IS THE DERIVATIVE   ***
C***                    OF THE STATE VARIABLE, AND X IS THE STATE VARI-   ***
C***                    ABLE, U THE CONTROL VARIABLE, AND X2, U2, X3,      ***
C***                    U3, ARE THE TENSOR PRODUCTS OF U AND X WITH        ***
C***                    THEMSELVES, THEN                                   ***
C***                                                                       ***
C***                              2       2        2      3       3        ***
C***                DX=A@X+B@U+A20@U2+A11@U*X+A02@X2+A30@U3+A21@U2*X+...  ***
C***                                                                       ***
C***                    WHERE '*' STANDS FOR THE TENSOR PRODUCT, AND       ***
C***                    '@' STANDS FOR THE CONTRACTION OPERATOR.           ***
C***            Q, R, Q30, Q21, Q12, Q03, Q40, Q31,...:  COEFFICIENTS      ***
C***                    OF THE POWERS OF THE CONTROL AND STATE VARI-       ***
C***                    ABLES IN THE INTEGRAL EQUATION FOR THE COST        ***
C***                    WHICH MUST BE MINIMIZED.                           ***
C***            V, V3, V4,...:  COEFFICIENTS FOR THE OPTIMAL COST FOR      ***
C***                    THE PROBLEM.                                       ***
C***            K, K2, K3,...:  OPTIMAL FEEDBACK COEFFICIENTS              ***
C***            K2S, K3S,...:  SYMMETRIC VERSIONS OF K2, K3,...            ***
C***            QS, RS, Q30S, Q21S,...:  SYMMETRIC VERSIONS OF Q, R,...   ***
C***            A20S, A02S, A30S, A21S,...:  SYMMETRIC VERSIONS OF A20,   ***
C***                    A02, A30,...                                       ***
C***            DV, DV3, DV4, DVS, DV3S, DV4S,...:  DERIVATIVES AND THE   ***
C***                    SYMMETRIC VERSIONS OF THE DERIVATIVES OF V, V3,   ***
C***                    V4,...                                             ***
C***            IDIMX, IDIMU, I2, I11, I20, I3, I21,...:  DIMENSIONS       ***
C***                    OF THE CONTROL AND STATE VECTORS AND HIGHER        ***
C***                    ORDER TENSOR PRODUCTS OF THOSE VARIABLES.          ***
C***                    E.G.,    I32=IDIMU**3*IDIMX**2.                    ***
C***            IS2, IS20, IS3, IS21,...:  DIMENSION OF SYMMETRIC          ***
C***                    TENSORS WHOSE UNSYMMETRIC DIMENSIONS ARE I2,       ***
C***                    I20, I3,...                                        ***
C***            IIT:  THE NUMBER OF INTEGRATION STEPS                      ***
C***            DT:  INTEGRATION STEPSIZE                                  ***
C***            DDT: -DT.  THE INTEGRATION STEPSIZE USED BECAUSE THE       ***
```

```
C***                INTEGRATION IS DONE IN NEGATIVE TIME                ***
C***          TO,T1:  INITIAL AND FINAL TIMES FOR THE INTEGRATION       ***
C***          J11:  NUMBER OF INTEGRATION STEPS BETWEEN STORED VALUES   ***
C***                OF V2S,K1,V3S,K2S,...                               ***
C***          IIIT:  NUMBER OF STORED VALUES OF V2S, K1, V3S,...        ***
C***                                                                    ***
C***          THERE ARE FIVE MAIN SECTIONS TO THIS PROGRAM:             ***
C***          1. THIS SECTION CONTAINS THE PARAMETERS, THE DIMENSION    ***
C***                STATEMENTS, AND THE DATA STATEMENTS.  THESE         ***
C***                DATA STATEMENTS LOAD IN THE VALUES OF THE SYS-      ***
C***                TEM AND COST FUNCTIONAL TENSORS.                    ***
C***          2. THIS SECTION CALCULATES THE FINAL VALUES OF THE        ***
C***                CONTROLLER TENSORS AND THE OPTIMAL VALUE            ***
C***                TENSORS.  THE STATEMENTS WHICH SETUP THE DATA       ***
C***                FILES FOR STORING THE CALCULATED TENSORS ARE        ***
C***                ALSO IN THIS SECTION.  SEVERAL TENSORS WHOSE        ***
C***                VALUES REMAIN CONSTANT FOR THE DURATION OF THE      ***
C***                CALCULATIONS ARE COMPUTED IN THIS SECTION.          ***
C***          3. THE CALCULATIONS OF V2S AND K1 ARE IN THIS SECTION.    ***
C***                THE RICCATI EQUATION IS INTEGRATED HERE USING       ***
C***                SEVERAL TENSORS WHICH WERE CALCULATED IN SEC-       ***
C***                TION 2.  THE RICCATI EQUATION IS DISCUSSED IN       ***
C***                THE COMMENTS IN SECTION 2.                          ***
C***          4. THE CALCULATIONS OF V3S AND K2S ARE IN THIS SECTION.   ***
C***                THE LINEAR DIFFERENTIAL EQUATION IS INTEGRATED      ***
C***                FOR V3S AND K2S IS FOUND AS AN AFFINE FUNCTION      ***
C***                OF V3S.                                             ***
C***          5. V4S AND K3S ARE CALCULATED IN THIS SECTION.  THESE     ***
C***                CALCULATIONS MIRROR THOSE FOR V3S AND K2S.          ***
C***                                                                    ***
C***********************************************************************
C***********************************************************************
C***********************************************************************
      SUBROUTINE TNSCLC
      INCLUDE 'PARAM1.FTN'
      DIMENSION V(IQ),A(IQ),B(IQ1D1),R(1R),RINV(1R),QUP(IQ1D1)
      DIMENSION QR(IQ),RIQ(IQ1D1),BRIQ(IQ),BRI(IQ1D1),Q1D1(IQ1D1)
      DIMENSION Q(IQ),BT(IQ1D1),BRB(IQ),QUPT(IQ1D1),QRQ(I2)
      DIMENSION QCONS(IQ),AV(IQ),VAV1(IQ),VAV2(IQ),VBRB(IQ),VBV(IQ)
      DIMENSION VAV3(IQ),VAV4(IQ),DV(IQ),BTV(I11),AVT(IQ)
      DIMENSION DVS(IS2),VS(IS2),QS(IS2),RS(IS20)
      REAL K(I11),K3(I3),KK(I22),KKK(I33),K3S(IS3),K2(I12),
     & K2S(IS12)
      DIMENSION A20(I21),A11(I12),A02(I3),Q30(I30),Q21(I21)
      DIMENSION Q03(I3),Q12(I12),F3(I3),BK(I2),ATAR(I2)
      DIMENSION AT(I2),V3(I3),DV3(I3),ATAT(I4),AT3(I6)
      DIMENSION Q3K3(I3),A11P(I12),A1K(I3),BTV3S(IS12)
      DIMENSION A2KK(I3),VA2K(I3),Q21P(I21),Q12P(I12)
      DIMENSION Q2KK(I3),Q1K(I3),F3S(IS3),DV4(I4)
      DIMENSION V3S(IS3),A30T(I31),A21T(I22),A21TP(I22),F4(I4)
      DIMENSION DV3S(IS3),A20V2K(I12),A2KKR(I3)
      DIMENSION BTV3(I12),A11T(I12),A11TP(I12),A11V(I12)
      DIMENSION Q30KK(I12),Q111P(I21),Q111K(I12),A20T(I21),V2K(I13)
      DIMENSION A21(I22),A21P(I22),A12(I13),A12P(I13),A30(I31),A03(I4)
      DIMENSION Q04(I4),Q13(I13),Q22(I22),Q31(I31),Q40(I40),Q13P(I13)
      DIMENSION Q22P(I22),Q31P(I31),AT4(I4*I4),A12T(I13),A12TP(I13)
```

```
      DIMENSION Q21PR(I21),Q31PR(I31),Q22PR(I22),V3B(I12)
      DIMENSION V3BK2(I4),A20K23(I4),A30KKK(I4),A20KK(I3),A11FK(I3)
      DIMENSION F4S(IS4),V4S(IS4)
      DIMENSION V4(I4),BTV4(I13),A12V2(I13),V2K2(I14),V3K(I14)
      DIMENSION BTV4S(IS13),AT41S(I4*IS4),X12(IDIMX)
      DIMENSION V2KK(I24),BK2(I3)
      REAL K3(I13),K3S(IS13),K34(I30*I4),K44(I40*I4)
      REAL K23(I23),K24(I24)
      DATA R/1./
      DATA Q1D1/0.,0./
      DATA Q/1,0,0,1/
      DATA V/0,0,0,0/
      DATA VS/0,0,0/
      DATA Q21,Q12,Q30,Q03/I21*0.0,I12*0.0,I30*0.0,I3*0.0/
      DATA RINV/1./
      DATA A/0,1,0,0/
      DATA B/0.,1./
      DATA A11/4*0./
      DATA A02/4*0.,1.,0.,0.,-1./
      DATA A20/2*0./
      DATA A30/2*0./
      DATA A12,A21/I13*0.,0.,0.,1.,-1./
      DATA A03/I4*0./
      DATA Q31,Q22,Q13,Q40,Q04/I31*0.,I22*0.,I13*0.,-.33333333,I4*0./
      DATA V3S,V3/IS3*0.,I3*0./
      DATA V4S,V4/IS4*0.,I4*0./
C***
C***  J9 IS A COUNTER WHICH DETERMINES WHICH VALUES OF THE OPTIMAL COST
C***  TENSORS AND OPTIMAL FEEDBACK TENSORS WILL BE SAVED BY THE PROGRAM.
C***  J9 RANGES FROM 1 TO J11.  WHEN IT REACHES J11, J10 IS INCREMENTED.
C***  J10 IS A COUNTER WHICH KEEPS TRACK OF WHICH RECORD THE NEXT VALUES
C***  OF THE CALCULATED TENSORS WILL BE STORED ON.  I34, I8, I44 ARE
C***  PARAMETERS WHICH ARE NEEDED IN THE PROGRAM BUT WERE NOT CALCULATED
C***  IN 'PARGEN' AND ARE THUS NOT IN 'PARAM1.FTR.'
C***
      J9=0
      J10=1
      I34=I30*I4
      I8=I4*I4
      I44=I40*I4
C***
C***  THE FOLLOWING SECTION CALCULATES SEVERAL TENSORS WHICH ARE NEEDED
C***  TO SOLVE THE RICCATI EQUATION FOR V.  THE RICCATI EQUATION IS:
C***
C***          DV+V@AV+AVT@V-V@BRB@V+QCONS=0
C***
C***  WHERE:
C***          DV IS THE DERIVATIVE OF V WITH RESPECT TO TIME
C***          AV=A-0.5*B@(R**(-1))@QUP
C***          QUP IS Q1D1 WITH THE COVARIANT POWER OF THE CONTROL RAISED
C***          AVT IS AV TRANSPOSED
C***          BRB=B@(R**(-1))@BT
C***          BT IS B TRANSPOSED
C***          QCONS=QR-0.25*QUPT@(R**(-1))@QUP
C***          QR IS Q WITH ONE COVARIANT POWER OF THE STATE VARIABLE RAISED
C***          QUPT IS QUP TRANSPOSED
```

```
C***            '@' IS THE CONTRACTION OPERATOR
C***            '*' IS SCALAR MULTIPLICATION
C***            R**(-1) IS THE TENSOR WHICH WHEN CONTRACTED WITH R, AFTER
C***                    RAISING ONE OF THE COVARIANT POWERS OF THE CONTROL
C***                    VARIABLE OF R, GIVES AN 'IDENTITY TENSOR.'  R**(-1)
C***                    WILL HAVE ONE CONTRAVARIANT AND ONE COVARIANT POWER
C***                    OF THE CONTROL VARIABLE.  THE RESULT OF CONTRACTING
C***                    R**(-1) AND R WITH A RAISED POWER IS A TENSOR OF THE
C***                    SAME TYPE AS R**(-1), AND THE IDENTITY TENSOR IS THUS
C***                    DEFINED AS A TENSOR WHOSE VALUE IS ONE IF THE COVARIANT
C***                    AND CONTRAVARIANT BASIS ELEMENTS HAVE THE SAME INDEX
C***                    AND ZERO OTHERWISE.
C***
C***    NOTE THAT THIS EQUATION IS CALLED A RICCATI EQUATION BECAUSE OF THE
C***    TERM 'V@BRB@V.'
C***
1       CALL SYM(Q,QS,0,2,0,0,2,IQ,IS2,IDIMU,IDIMX,1,1)
        CALL RAISE(Q,QR,0,0,2,1,0,1,IDIMU,IDIMX,IQ,2)
        CALL RAISE(Q1D1,QUP,0,1,1,1,0,1,IDIMU,IDIMX,IQ1D1,1)
        CALL TCONT(RINV,QUP,RIQ,1,1,0,1,0,1,1,IR,I11,I11,IDIMU,IDIMX,4)
        CALL TCONT(B,RIQ,BRIQ,1,1,0,1,0,1,1,I11,I11,I2,IDIMU,IDIMX,3)
        CALL SMULT(BRIQ,I2,-0.5)
        CALL TADD(A,BRIQ,AV,I2)
        CALL TRANS(AV,AVT,1,1,0,IQ,IDIMX,IDIMX,1)
        CALL TCONT(B,RINV,BRI,1,1,0,1,1,0,1,I11,IR,I11,IDIMU,IDIMX,3)
        CALL TRANS(B,BT,1,1,0,I11,IDIMU,IDIMX,1)
        CALL TCONT(BRI,BT,BRB,1,1,0,1,0,1,1,I11,I11,I2,IDIMU,IDIMX,3)
        CALL TRANS(QUP,QUPT,1,0,1,I11,IDIMU,IDIMX,2)
        CALL TCONT(QUPT,RIQ,QRQ,1,1,0,1,0,1,1,I11,I11,I2,IDIMU,IDIMX,3)
        CALL SMULT(QRQ,I2,-0.25)
        CALL TADD(QRQ,QR,QCONS,IQ)
C***
C***    NOTE THAT THE INTEGRATION IS BACKWARD IN TIME FOR CALCULATING V
C***    AND V3, V4,,,, BECAUSE THE FINAL VALUE IS KNOWN.  THUS INITIALLY
C***    T=T1 AND DDT=-DT, WHERE DDT IS THE INTEGRATION TIMESTEP USED.
C***
        T=T1
        DDT=-DT
C***
C***    THE NEXT FEW LINES CALCULATE THE FINAL VALUE FOR K.  K IS CALCULATED
C***    FROM V VIA THE FOLLOWING EQUATION:
C***
C***    K=-1*R**(-1)@(BT@V+0.5*QUP)
C***
C***    WHERE EACH OF THESE SYMBOLS HAS BEEN DEFINED ABOVE.
C***
        CALL TCONT(BT,V,BTV,1,0,1,1,0,1,1,I11,I2,I11,IDIMU,IDIMX,2)
        CALL SMULT(QUP,I11,0.5)
        CALL TADD(BTV,QUP,BTV,I11)
        CALL SMULT(QUP,I11,2.)
        CALL TCONT(RINV,BTV,K,1,1,0,1,0,1,1,I20,I11,I11,IDIMU,IDIMX,4)
        CALL SMULT(K,I11,-1.)
C***
C***    THE VALUES FOR VS, K, V3S, K2S, V4S, K3S, ARE STORED IN DIRECT
C***    ACCESS FILES.  DIRECT ACCESS FILES MUST BE USED FOR K, K2S, AND
C***    K3S BECAUSE OF THE NEED TO BE ABLE TO READ OUT THE VALUES FROM
```

```
C***    THE END OF THE FILES BEFORE THE VALUES IN THE BEGINNING OF THE
C***    FILES.  THIS MUST BE DONE WHEN THE VALUES OF THE STATES ARE
C***    CALCULATED IN 'XCALC' BECAUSE THAT INTEGRATION IS DONE FORWARD IN
C***    TIME.
C***
        OPEN(UNIT=1,NAME='VSI.DAT',TYPE='NEW',ACCESS='DIRECT',FORM=
     &'FORMATTED',RECORDSIZE=IS2*11,CARRIAGECONTROL='LIST')
        OPEN(UNIT=2,NAME='K1I.DAT',TYPE='NEW',ACCESS='DIRECT',FORM=
     &'FORMATTED',RECORDSIZE=23,CARRIAGECONTROL='LIST')
        OPEN(UNIT=3,NAME='V3SI.DAT',TYPE='NEW',ACCESS='DIRECT',FORM=
     &'FORMATTED',RECORDSIZE=IS3*11,CARRIAGECONTROL='LIST')
        OPEN(UNIT=4,NAME='K2SI.DAT',TYPE='NEW',ACCESS='DIRECT',FORM=
     &'FORMATTED',RECORDSIZE=34,CARRIAGECONTROL='LIST')
        OPEN(UNIT=7,NAME='V4SI.DAT',TYPE='NEW',ACCESS='DIRECT',FORM=
     &'FORMATTED',RECORDSIZE=IS4*11,CARRIAGECONTROL='LIST')
        OPEN(UNIT=8,NAME='K3SI.DAT',TYPE='NEW',ACCESS='DIRECT',FORM=
     &'FORMATTED',RECORDSIZE=45,CARRIAGECONTROL='LIST')
        WRITE(1'1,491)VS
        WRITE(2'1,492)K
        WRITE(3'1,493)V3S
C***
C***    AFTER STORING VS, K, AND V3S, THE FIRST VALUE OF K2 IS CALCULATED.
C***    THIS CALCULATION IS THE SAME AS THAT WITHIN THE INTEGRATION LOOP.
C***
        CALL TRANS(A11,A11T,1,1,1,I12,IDIMU,IDIMX,1)
        CALL PERM(A11T,A11TP,1,1,1,I12,IDIMU,IDIMX,10,4)
        CALL TCONT(A11TP,V,A11V,1,0,2,1,0,1,1,I12,12,I12,IDIMU,IDIMX,2)
        CALL SMULT(A11V,I12,2,0)
        CALL TMULT(K,K,KK,0,1,1,0,1,1,I11,I11,I22,IDIMU,IDIMX,4)
        CALL TCONT(Q30,KK,Q30KK,1,2,0,2,0,2,2,I30,I22,I12,IDIMU,IDIMX,4)
        CALL SMULT(Q30KK,I12,3,0)
        CALL TADD(A11V,Q30KK,BTV3,I12)
        CALL PERM(Q21,Q111P,1,1,1,I21,IDIMU,IDIMX,2,5)
        CALL TCONT(Q111P,K,Q111K,1,1,1,1,1,0,1,I21,I11,I12,IDIMX,IDIMU,1
     &)
        CALL SMULT(Q111K,I12,2,0)
        CALL TADD(BTV3,Q111K,BTV3,I12)
        CALL TADD(BTV3,Q12,BTV3,I12)
        CALL TRANS(A20,A20T,1,2,0,I21,IDIMU,IDIMX,1)
        CALL TMULT(V,K,V2K,1,0,1,0,1,1,12,I11,I13,IDIMU,IDIMX,3)
        CALL TCONT(A20T,V2K,A20V2K,IDIMU,0,1,1,0,2,1,I21,I13,I12,I11,IDIMX,10)
        CALL SMULT(A20V2K,I12,4,0)
        CALL TADD(BTV3,A20V2K,BTV3,I12)
        CALL SYM(BTV3,BTV3S,1,2,0,0,2,I12,IS12,IDIMU,IDIMX,1,11)
        CALL TCONT(RINV,BTV3S,K2S,1,1,0,1,0,1,1,I20,IS12,IS12,IDIMU,IS2,4)
        CALL SMULT(K2S,IS12,-0.5)
        WRITE(4'1,494)K2S
        WRITE(7'1,497)V4S
C***
C***    AFTER CALCULATING K2S, K2S AND V4S ARE STORED.  NEXT, SEVERAL
C***    CALCULATIONS WHICH ARE PERFORMED ON TIME-INDEPENDENT TENSORS
C***    AND WHICH ARE NEEDED FOR OTHER CALCULATIONS WITHIN THE INTEGRATION
C***    LOOP ARE CARRIED OUT.  OBVIOUSLY, DOING THESE CALCULATIONS
C***    REPEATEDLY WITHIN THE INTEGRATION LOOP WOULD BE A GREAT WASTE OF
C***    TIME.
C***
```

```
      CALL PERM(A11,A11P,1,1,1,I12,IDIMU,IDIMX,1,1)
      CALL PERM(Q21,Q21P,0,2,1,I21,IDIMU,IDIMX,3,2)
      CALL PERM(Q12,Q12P,0,1,2,I12,IDIMU,IDIMX,3,3)
      CALL PERM(A21,A21P,1,2,1,I22,IDIMU,IDIMX,1,11)
      CALL PERM(A12,A12P,1,1,2,I13,IDIMU,IDIMX,1,12)
      CALL PERM(Q31,Q31P,0,3,1,I31,IDIMU,IDIMX,3,15)
      CALL PERM(Q22,Q22P,0,2,2,I22,IDIMU,IDIMX,3,16)
      CALL PERM(Q13,Q13P,0,1,3,I13,IDIMU,IDIMX,3,17)
      CALL TRANS(A12,A12T,1,1,2,I13,IDIMU,IDIMX,1)
      CALL PERM(A12T,A12TP,1,1,2,I13,IDIMU,IDIMX,10,18)
      CALL PERM(Q21,Q21PR,1,1,1,I21,IDIMU,IDIMX,2,20)
      CALL PERM(Q31,Q31PR,1,2,1,I31,IDIMU,IDIMX,2,21)
      CALL PERM(Q22,Q22PR,1,1,2,I22,IDIMU,IDIMX,2,22)
      CALL TRANS(A21,A21T,1,2,1,I22,IDIMU,IDIMX,1)
      CALL PERM(A21T,A21TP,IDIMU,1,1,I22,I11,IDIMX,11,23)
      CALL TRANS(A30,A30T,1,3,0,I31,IDIMU,IDIMX,1)
C***
C***  NEXT THE FIRST VALUE OF K3S IS CALCULATED.  THIS IS THE SAME
C***  CALCULATION DONE WITHIN THE INTEGRATION LOOP AFTER EACH CALCULATION
C***  OF V4.
C***
      CALL TCONT(BT,V4,BTV4,1,0,1,1,0,3,1,I11,I4,I13,IDIMU,IDIMX,2)
      CALL SMULT(BTV4,I13,4,0)
      CALL TCONT(A12TP,V,A12V2,1,0,3,1,0,1,1,I13,I2,I13,IDIMU,IDIMX,2)
      CALL SMULT(A12V2,I13,2,0)
      CALL TADD(BTV4,A12V2,BTV4,I13)
      CALL SYM(K2,K2S,1,2,0,0,2,I12,IS12,IDIMU,IDIMX,2,19)
      CALL TMULT(V,K2,V2K2,1,0,1,0,1,2,I2,I12,I14,IDIMU,IDIMX,16)
      CALL TCONT(A20T,V2K2,A12V2,IDIMU,0,1,1,0,3,1,I21,I14,I13,I11,IDIMX,10)
      CALL SMULT(A12V2,I13,4,0)
      CALL TADD(BTV4,A12V2,BTV4,I13)
      CALL TMULT(V,KK,V2KK,1,0,1,0,2,2,I2,I22,I24,IDIMU,IDIMX,27)
      CALL TCONT(A30T,V2KK,A12V2,IDIMU,0,1,1,0,3,1,I31,I24,I13,I21,IDIMX,10)
      CALL SMULT(A12V2,I13,6,0)
      CALL TADD(BTV4,A12V2,BTV4,I13)
      CALL TCONT(A21TP,V2K,A12V2,I11,0,1,1,0,2,1,I22,I13,I13,I11,IDIMX,10)
      CALL SMULT(A12V2,I13,4,0)
      CALL TADD(BTV4,A12V2,BTV4,I13)
      CALL TCONT(A11TP,V3,A12V2,1,0,2,1,0,2,1,I12,I3,I13,IDIMU,IDIMX,2)
      CALL SMULT(A12V2,I13,3,0)
      CALL TADD(BTV4,A12V2,BTV4,I13)
      CALL TMULT(V3,K,V3K,1,0,2,0,1,1,I3,I11,I14,IDIMU,IDIMX,17)
      CALL TCONT(A20T,V3K,A12V2,IDIMU,0,1,1,0,3,1,I21,I14,I13,I11,IDIMX,10)
      CALL SMULT(A12V2,I13,6,0)
      CALL TADD(BTV4,A12V2,BTV4,I13)
      CALL TADD(BTV4,Q13,BTV4,I13)
      CALL TMULT(K,K2,K23,0,1,1,0,1,2,I11,I12,I23,IDIMU,IDIMX,20)
      CALL TCONT(Q30,K23,A12V2,1,2,0,2,0,3,2,I30,I23,I13,IDIMU,IDIMX,4)
      CALL SMULT(A12V2,I13,6,0)
      CALL TADD(BTV4,A12V2,BTV4,I13)
      CALL TMULT(K,KK,KKK,0,1,1,0,2,2,I11,I22,I33,IDIMU,IDIMX,21)
      CALL TCONT(Q40,KKK,A12V2,1,3,0,3,0,3,3,I40,I33,I13,IDIMU,IDIMX,4)
      CALL SMULT(A12V2,I13,4,0)
      CALL TADD(BTV4,A12V2,BTV4,I13)
      CALL TCONT(Q21PR,K2,A12V2,1,1,1,1,2,0,1,I21,I12,I13,IDIMX,IDIMU,1)
      CALL SMULT(A12V2,I13,2,0)
```

```
      CALL TADD(BTV4,A12V2,BTV4,I13)
      CALL TCONT(Q31PR,KK,A12V2,1,1,2,2,2,0,2,I31,I22,I13,IDIMX,IDIMU,1)
      CALL SMULT(A12V2,I13,3,0)
      CALL TADD(BTV4,A12V2,BTV4,I13)
      CALL TCONT(Q22PR,K,A12V2,1,2,1,1,1,0,1,I22,I11,I13,IDIMX,IDIMU,1)
      CALL SMULT(A12V2,I13,2,0)
      CALL TADD(BTV4,A12V2,BTV4,I13)
      CALL SYM(BTV4,BTV4S,0,3,1,0,3,I13,IS13,IDIMU,IDIMX,1,26)
      CALL SYM(BTV4,BTV4S,0,3,1,0,3,I13,IS13,IDIMU,IDIMX,2,27)
      CALL TCONT(RINV,BTV4,K3,1,1,0,1,0,3,1,I20,I13,I13,IDIMU,IDIMX,4)
      CALL SMULT(K3,I13,-0.5)
      CALL SYM(K3,K3S,1,3,0,0,3,I13,IS13,IDIMU,IDIMX,1,28)
      WRITE(8'J10,498)K3S
C***
C***  K3S HAS NOW BEEN CALCULATED AND SAVED.  NOW THE REMAINING VALUES
C***  OF V, V3, K, K2S,,,, CAN BE CALCULATED AFTER EACH INTEGRATION STEP.
C***
      DO 100 J=2,IIT
C***
C***  J9 IS INCREMENTED NOW.  IF J9=J11, THEN THIS VALUE OF THE TENSORS
C***  WILL BE SAVED, AND J9 WILL BE SET BACK TO ZERO.  IF J9 IS LESS
C***  THAN J11, THE VALUES CALCULATED THIS TIME AROUND WILL NOT BE SAVED.
C***
      J9=J9+1
C***
C***  THE INTEGRATION OF THE RICCATI EQUATION IS THE FIRST PERFORMED.
C***  THE CONSTANT COEFFICIENTS WERE CALCULATED OUTSIDE OF THE INTEGRATION
C***  LOOP.  NOTE THAT THE TERMS INVOLVING V MUST BE CALCULATED FOUR TIMES
C***  BEFORE THE NEXT VALUE OF V IS CALCULATED BY RUNGE.  THE ENTIRE
C***  RICCATI EQUATION WAS DESCRIBED ABOVE.
C***
2     DO 20 M=1,4
      CALL TCONT(V,AV,VAV1,1,0,1,1,0,1,1,IQ,IQ,IQ,IDIMU,IDIMX,1)
      CALL TCONT(AVT,V,VAV2,1,0,1,1,0,1,1,IQ,IQ,IQ,IDIMU,IDIMX,1)
      CALL TADD(VAV1,VAV2,VAV3,IQ)
      CALL TCONT(V,BRB,VBRB,1,0,1,1,0,1,1,IQ,IQ,IQ,IDIMU,IDIMX,1)
      CALL TCONT(VBRB,V,VBV,1,0,1,1,0,1,1,IQ,IQ,IQ,IDIMU,IDIMX,1)
      CALL SMULT(VBV,IQ,-1.)
      CALL TADD(VAV3,VBV,VAV4,IQ)
            CALL TADD(VAV4,RCONS,DV,IQ)
            CALL SMULT(DV,I2,-1,0)
            CALL RUNGE(I2,V,DV,1,DDT,M)
20    CONTINUE
C***
      IF(J9.NE.J11)GO TO 30
C***
C***  IF J9=J11, VS IS FOUND FROM V AND IS SAVED.
C***
C..         WRITE(5,*)' DV',(DV(J1),J1=1,I2)
            CALL SYM(V,VS,0,2,0,0,2,I2,IS2,IDIMU,IDIMX,1,5)
            J10=J10+1
            WRITE(1'J10,491)VS
C***
C***  NEXT K IS CALCULATED.
C***
30    CALL TCONT(BT,V,BTV,1,0,1,1,0,1,1,I11,I2,I11,IDIMU,IDIMX,2)
```

```
      CALL SMULT(QUP,I11,0.5)
      CALL TADD(BTV,QUP,BTV,I11)
      CALL SMULT(QUP,I2,2.)
      CALL TCONT(RINV,BTV,K,1,1,0,1,0,1,1,I20,I11,I11,IDIMU,IDIMX,4)
      CALL SMULT(K,I11,-1.)
      IF (J9.NE.J11)GOTO 31
            WRITE(2'J10,492)K
C***
C***  RUNGE INCREMENTS T BY DDT, THUS FOR THE NEXT INTEGRATION T MUST
C***  BE CHANGED BACK TO T-DDT.  THE CALCULATION OF V3 BEGINS NOW.
C***
31    T=T-DDT
      CALL TCONT(B,K,BK,1,1,0,1,0,1,1,I11,I11,I2,IDIMU,IDIMX,3)
      CALL TADD(A,BK,ATAR,I2)
      CALL TRANS(ATAR,AT,1,0,1,I2,IDIMX,IDIMX,2)
      CALL TMULT(K,K,KK,0,1,1,0,1,1,I11,I11,I22,IDIMU,IDIMX,1)
      CALL TCONT(A20,KK,A2KK,1,2,0,2,0,2,2,I21,I22,I3,IDIMU,IDIMX,3)
C***
C***  NOTE THIS NEXT CALL IS SPECIALLY ADAPTED TO THIS SITUATION
C***  WHERE A11 HAS BEEN PERMUTED (REORDERED) SUCH THAT THE X'S AND
C***  U'S HAVE SWITCHED PLACES.
C***
      CALL TCONT(A11P,K,A1K,1,1,1,1,1,0,1,I12,I11,I3,IDIMX,IDIMU,2)
      CALL TADD(A1K,A2KK,A2KK,I3)
      CALL TADD(A2KK,A02,A2KK,I3)
      CALL DPROD(AT,AT,ATAT,I2,I2,I4,1)
      CALL DPROD(ATAT,AT,AT3,I4,I2,I6,2)
      CALL TCONT(Q21P,KK,Q2KK,0,1,2,2,2,0,2,I21,I22,I3,IDIMX,IDIMU,2)
      CALL TMULT(K,KK,KKK,0,1,1,0,2,2,I11,I22,I33,IDIMU,IDIMX,2)
      CALL TCONT(Q30,KKK,Q3K3,0,3,0,3,0,3,3,I30,I33,I3,IDIMU,IDIMX,3)
      CALL TADD(Q3K3,Q2KK,Q3K3,I3)
      CALL TCONT(Q12P,K,Q1K,0,2,1,1,1,0,1,I12,I11,I3,IDIMX,IDIMU,2)
      CALL TADD(Q3K3,Q1K,Q3K3,I3)
      CALL TADD(Q3K3,R03,R3K3,I3)
      CALL TCONT(V,A2KK,VA2K,1,0,1,3,0,0,1,I2,I3,I3,IDIMU,IDIMX,1)
      CALL SMULT(VA2K,I3,2.0)
      CALL TADD(VA2K,Q3K3,VA2K,I3)
      CALL SYM(VA2K,F3S,0,3,0,0,3,I3,IS3,IDIMU,IDIMX,1,7)
      CALL SYM(F3,F3S,0,3,0,0,3,I3,IS3,IDIMU,IDIMX,2,8)
C***
C***  THE FOUR STEP RUNGE-KUTTA INTEGRATION STARTS HERE FOR V3S.
C***
119   DO 130 M=1,4
      CALL TCONT(AT3,V3,DV3,1,0,1,1,0,0,1,I6,I3,I3,IDIMU,I3,1)
      CALL TADD(DV3,F3,DV3,I3)
      CALL SMULT(DV3,I3,-1.0)
      CALL RUNGE(I3,V3,DV3,T,DDT,M)
130   CONTINUE
C***
C***  V3 HAS NOW BEEN FOUND.  IT MUST BE SAVED, IF J9=J11, AND V3S MUST
C***  BE DETERMINED.  THEN THE CALCULATION FOR K2 STARTS.
C***
      CALL SYM(V3,V3S,0,3,0,0,3,I3,IS3,IDIMU,IDIMX,1,9)
      IF(J9.NE.J11)GOTO 136
            WRITE(3'J10,493)V3S
136   CALL TCONT(BT,V3,BTV3,1,0,1,1,0,2,1,I11,I3,I12,IDIMU,IDIMX,2)
```

```
      CALL SMULT(BTV3,I12,3.0)
      CALL TCONT(A11TP,V,A11V,1,0,2,1,0,1,1,I12,I2,I12,IDIMU,IDIMX,2)
      CALL SMULT(A11V,I12,2.0)
      CALL TADD(BTV3,A11V,BTV3,I12)
      CALL TCONT(Q30,KK,Q30KK,1,2,0,2,0,2,2,I30,I22,I12,IDIMU,IDIMX,4)
      CALL SMULT(Q30KK,I12,3.0)
      CALL TADD(BTV3,Q30KK,BTV3,I12)
      CALL TCONT(Q111P,K,Q111K,1,1,1,1,1,0,1,I21,I11,I12,IDIMX,IDIMU,1
     1)
      CALL SMULT(Q111K,I12,2.0)
      CALL TADD(BTV3,Q111K,BTV3,I12)
      CALL TADD(BTV3,Q12,BTV3,I12)
      CALL TMULT(V,K,V2K,1,0,1,0,1,1,I2,I11,I13,IDIMU,IDIMX,3)
      CALL TCONT(A20T,V2K,A20V2K,IDIMU,0,1,1,0,2,1,I21,I13,I12,I11,IDIMX,10)
      CALL SMULT(A20V2K,I12,4.0)
      CALL TADD(BTV3,A20V2K,BTV3,I12)
      CALL SYM(BTV3,BTV3S,1,2,0,0,2,I12,IS12,IDIMU,IDIMX,1,11)
      CALL TCONT(RINV,BTV3S,K2S,1,1,0,1,0,1,1,I20,IS12,IS12,IDIMU,IS2,4)
      CALL SMULT(K2S,IS12,-0.5)
      IF(J9.NE.J11)GOTO 146
            WRITE(4'J10,494)K2S
146   CONTINUE
      CALL SYM(K2,K2S,1,2,0,0,2,I12,IS12,IDIMU,IDIMX,2,20)
C***
C***  NOW THAT K2 HAS BEEN FOUND, T MUST BE SET BACK TO T-DDT AND THE
C***  CALCULATION FOR V4 STARTS.
C***
      T=T-DDT
C***
C***  IN THE NEXT SECTION, A20K23 AND V3BK2 ARE USED AS RUNNING SUMS FOR
C***  THE CALCULATIONS BY UTILIZING THE APPROPRIATE OPTION IN TCONT1.
C***  THUS, THE FOLLOWING EQUATION WILL HOLD FOR THE FINAL VALUE OF
C***  V3BK2:
C***
C***              4
C***        DV4+AT4@V4+V3BK2=0
C***
      CALL SPROD(AT,AT3,AT4,I2,I6,I8,10)
      CALL TCONT(B,K2,BK2,1,1,0,1,0,2,1,I11,I12,I3,IDIMU,IDIMX,3)
      CALL TADD(BK2,A2KK,BK2,I3)
      CALL TCONT1(V3,BK2,V3BK2,2,0,1,1,0,2,1,I3,I3,I4,IDIMU,IDIMX,1,3.0)
      CALL TMULT(K,K2,K23,0,1,1,0,1,2,I11,I12,I23,IDIMU,IDIMX,10)
      CALL SMULT(K23,I23,2.0)
      CALL TCONT1(A20,K23,A20K23,1,2,0,2,0,3,2,I21,I23,I4,IDIMU,IDIMX,3,1.)
      CALL TADD(A03,A20K23,A20K23,I4)
      CALL TCONT1(A30,KKK,A20K23,1,3,0,3,0,3,3,I31,I33,I4,IDIMU,IDIMX,-3,1.0)
      CALL TCONT1(A11P,K2,A20K23,1,1,1,1,2,0,1,I12,I12,I4,IDIMX,IDIMU,-2,1.0)
      CALL TCONT1(A21P,KK,A20K23,1,1,2,2,2,0,2,I22,I22,I4,IDIMX,IDIMU,-2,1.0)
      CALL TCONT1(A12P,K,A20K23,1,2,1,1,1,0,1,I13,I11,I4,IDIMX,IDIMU,-2,1.0)
      CALL TCONT1(V,A20K23,V3BK2,1,0,1,4,0,0,1,I2,I4,I4,IDIMU,IDIMX,-1,2.0)
      CALL TMULT(K2,K2,K24,0,1,2,0,1,2,I12,I12,I24,IDIMU,IDIMX,13)
      CALL TCONT1(R,K24,V3BK2,0,2,0,2,0,4,2,I20,I24,I4,IDIMU,IDIMX,-3,1.)
      CALL TADD(V3BK2,Q04,V3BK2,I4)
      CALL TMULT(KK,K2,K34,0,2,2,0,1,2,I22,I12,I34,IDIMU,IDIMX,14)
      CALL SMULT(K34,I34,3.0)
      CALL TCONT1(Q30,K34,V3BK2,0,3,0,3,0,4,3,I30,I34,I4,IDIMU,IDIMX,-3,1.0)
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      CALL TMULT(KK,KK,K44,0,2,2,0,2,2,I22,I22,I44,IDIMU,IDIMX,15)
      CALL TCONT1(Q40,K44,V3BK2,0,4,0,4,0,4,4,I40,I44,I4,IDIMU,IDIMX,-3,1.)
      CALL TCONT1(Q21P,K23,V3BK2,0,1,2,2,3,0,2,I21,I23,I4,IDIMX,IDIMU,-1,1.)
      CALL TCONT1(Q31P,KKK,V3BK2,0,1,3,3,3,0,3,I31,I33,I4,IDIMX,IDIMU,-1,1.)
      CALL TCONT1(Q12P,K2,V3BK2,0,2,1,1,2,0,1,I12,I12,I4,IDIMX,IDIMU,-1,1.)
      CALL TCONT1(Q22P,KK,V3BK2,0,2,2,2,2,0,2,I22,I22,I4,IDIMX,IDIMU,-1,1.)
      CALL TCONT1(Q13P,K,V3BK2,0,3,1,1,1,0,1,I13,I11,I4,IDIMX,IDIMU,-1,1.)
      CALL SYM(V3BK2,F4S,0,4,0,0,4,I4,IS4,IDIMU,IDIMX,1,23)
      CALL SYM(F4,F4S,0,4,0,0,4,I4,IS4,IDIMU,IDIMX,2,24)
      DO 305 M=1,4
      CALL TCONT(AT4,V4,DV4,1,0,1,1,0,0,1,14*I4,I4,I4,IDIMU,I4,1)
      CALL TADD(DV4,F4,DV4,I4)
      CALL SMULT(DV4,I4,-1.0)
      CALL RUNGE(I4,V4,DV4,T,DDT,M)
305   CONTINUE
C***
C***  V4 HAS BEEN DETERMINED.  IF J9=J11, V4S MUST BE SAVED AND K3S MUST
C***  BE CALCULATED.  IN THE CALCULATION OF K3S, BTV4 ACTS AS A RUNNING
C***  SUM IN THE SAME MANNER AS V3BK2 ACTED AS A RUNNING SUM ABOVE.
C***
      IF(J9.NE.J11)GOTO 100
      CALL SYM(V4,V4S,0,4,0,0,4,I4,IS4,IDIMU,IDIMX,1,25)
      WRITE(7'J10,497)V4S
      CALL TCONT1(BT,V4,BTV4,1,0,1,1,0,3,1,I11,I4,I13,IDIMU,IDIMX,2,4.)
      CALL TCONT1(A12TP,V,BTV4,1,0,3,1,0,1,1,I13,I2,I13,IDIMU,IDIMX,-2,2.)
      CALL TMULT(V,K2,V2K2,1,0,1,0,1,2,I2,I12,I14,IDIMU,IDIMX,16)
      CALL TCONT1(A20T,V2K2,BTV4,IDIMU,0,1,1,0,3,1,I21,I14,I13,I11,IDIMX,-10
     &,4.0)
      CALL TMULT(V,KK,V2KK,1,0,1,0,2,2,I2,I22,I24,IDIMU,IDIMX,27)
      CALL TCONT1(A30T,V2KK,BTV4,IDIMU,0,1,1,0,3,1,I31,I24,I13,I21,IDIMX,-10
     &,6.0)
      CALL TCONT1(A21TP,V2K,BTV4,I11,0,1,1,0,2,1,I22,I13,I13,I11,IDIMX,-10
     &,4.0)
      CALL TCONT1(A11TP,V3,BTV4,1,0,2,1,0,2,1,I12,I3,I13,IDIMU,IDIMX,-2,3.)
      CALL TMULT(V3,K,V3K,1,0,2,0,1,1,I3,I11,I14,IDIMU,IDIMX,17)
      CALL TCONT1(A20T,V3K,BTV4,IDIMU,0,1,1,0,3,1,I21,I14,I13,I11,IDIMX,-10
     &,6.0)
      CALL TADD(BTV4,Q13,BTV4,I13)
      CALL TCONT1(Q30,K23,BTV4,1,2,0,2,0,3,2,I30,I23,I13,IDIMU,IDIMX,-4,3.)
      CALL TCONT1(Q40,KKK,BTV4,1,3,0,3,0,3,3,I40,I33,I13,IDIMU,IDIMX,-4,4.)
      CALL TCONT1(Q21PR,K2,BTV4,1,1,1,1,2,0,1,I21,I12,I13,IDIMX,IDIMU,-1,2.)
      CALL TCONT1(Q31PR,KK,BTV4,1,1,2,2,2,0,2,I31,I22,I13,IDIMX,IDIMU,-1,3.)
      CALL TCONT1(Q22PR,K,BTV4,1,2,1,1,1,0,1,I22,I11,I13,IDIMX,IDIMU,-1,2.)
      CALL SYM(BTV4,BTV4S,0,3,1,0,3,I13,IS13,IDIMU,IDIMX,1,26)
      CALL SYM(BTV4,BTV4S,0,3,1,0,3,I13,IS13,IDIMU,IDIMX,2,27)
      CALL TCONT(RINV,BTV4,K3,1,1,0,1,0,3,1,I20,I13,I13,IDIMU,IDIMX,4)
      CALL SMULT(K3,I13,-0.5)
      CALL SYM(K3,K3S,1,3,0,0,3,I13,IS13,IDIMU,IDIMX,1,28)
      WRITE(8'J10,498)K3S
      J9=0
100   CONTINUE
C***
C***  AFTER ALL OF THE CALCULATIONS HAVE BEEN PERFORMED, THE OUTPUT FILES
C***  CLOSED.
C***
      CLOSE(UNIT=1,DISPOSE='SAVE')
```

```
      CLOSE(UNIT=2,DISPOSE='SAVE')
      CLOSE(UNIT=3,DISPOSE='SAVE')
      CLOSE(UNIT=4,DISPOSE='SAVE')
      CLOSE(UNIT=7,DISPOSE='SAVE')
      CLOSE(UNIT=8,DISPOSE='SAVE')
      RETURN
C***
C***  THE FOLLOWING ARE THE FORMAT STATEMENTS.  THESE ARE USED TO OUTPUT
C***  THE VALUES OF V2S, K1,... AND MUST BE CHANGED ACCORDING TO THE
C***  PROBLEM SIZE.  ONE GOAL WHICH IS EASILY ATTAINABLE IS TO HAVE THE
C***  PROGRAM PARGEN WRITE A FILE OF FORMAT STATEMENTS AND TO HAVE THIS
C***  PROGRAM 'INCLUDE' THAT FILE.
C***
491   FORMAT(3E11.4)
492   FORMAT(1X,2E11.4)
493   FORMAT(4E11.4)
494   FORMAT(1X,3E11.4)
497   FORMAT(5E11.4)
498   FORMAT(1X,4E11.4)
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
C***********************************************************************
C***********************************************************************
C***********************************************************************
C***                                                                 ***
C***            TITLE:  XCALC                                        ***
C***            AUTHOR:  JOSEPH A. O'SULLIVAN                        ***
C***                                                                 ***
C***            SUBROUTINE XCALC CALCULATES THE TRAJECTORIES OF THE  ***
C***    STATES OF A SYSTEM AFTER THE FEEDBACK TENSORS HAVE BEEN SAVED ***
C***    THE VALUES OF THE STATES ARE SAVED IN SEQUENTIAL FILES FOR   ***
C***    THREE DIFFERENT CASES:  LINEAR FEEDBACK, QUADRATIC FEEDBACK, ***
C***    AND THIRD ORDER FEEDBACK.  THIS SUBROUTINE IS NORMALLY USED IN ***
C***    THE SAME PROGRAM AS TNSCLC WHICH CALCULATES AND SAVES THE    ***
C***    FEEDBACK TENSORS.  IT CAN, HOWEVER, BE USED IN A PROGRAM WHICH ***
C***    USES FEEDBACK TENSORS OTHER THAN THE OPTIMAL ONES, UNDER THE ***
C***    PROVISION THAT THEY ARE STORED IN FILES OF THE SAME TYPE AND ***
C***    NAME AS USED IN THIS SUBROUTINE.                             ***
C***            THIS SUBROUTINE USES THE FILE 'PARAM1.FTN' WHICH MUST ***
C***    BE SET UP BY 'DK2:PARGEN.TSK'.  THIS SAME 'PARAM1.FTN' MUST BE ***
C***    USED BY 'TNSCLC' OR ELSE THE FILES WITH THE FEEDBACK TENSORS ***
C***    WILL BE OF A DIFFERENT SIZE, AND THE COEFFICIENT TENSORS IN THE ***
C***    DIFFERENTIAL EQUATION DESCRIBING THE SYSTEM WILL BE DIFFERENT ***
C***    SIZES AS WELL.                                               ***
C***                                                                 ***
C***    IMPORTANT VARIABLES:                                         ***
C***            IDIMX, IDIMU, I2, I11, I20, I3, I21, I12, I30,...:   ***
C***                    DIMENSION PARAMETERS CALCULATED BY PARGEN AND ***
C***                    USED IN THIS PROGRAM VIA THE FILE 'PARAM1.FTN' ***
C***            IIIT:  THE NUMBER OF INTEGRATION STEPS IN THIS SUBROU- ***
C***                    TINE.  THIS IS ALSO FROM 'PARAM1.FTN'        ***
C***            X:  THE STATE VECTOR WHICH IS BEING INTEGRATED       ***
C***            U:  CONTROL VECTOR WHICH IS A POLYNOMIAL FUNCTION OF ***
C***                    THE STATES                                   ***
C***            DX:  DERIVATIVE OF X                                 ***
C***            X1:  STATE VECTOR CALCULATED USING QUADRATIC FEEDBACK ***
C***            X12:  STATE VECTOR CALCULATED USING THIRD ORDER FEED- ***
C***                    BACK                                         ***
C***            U2:  TENSOR PRODUCT OF U WITH ITSELF                 ***
C***            X2:  TENSOR PRODUCT OF X WITH ITSELF                 ***
C***            X2S:  SYMMETRIC TENSOR PRODUCT OF X WITH ITSELF      ***
C***            BU:  DUMMY VARIABLE USED IN THE CALCULATION OF DX    ***
C***            UX:  TENSOR PRODUCT OF U WITH X                      ***
C***            X3, X3S:  THIRD ORDER TENSOR PRODUCT OF X; AND ITS SYM- ***
C***                    METRIC VERSION, RESPECTIVELY                 ***
C***            K, K2S, K3S:  SYMMETRIC FEEDBACK TENSORS             ***
C***            K2X2S:  INTERMEDIATE VALUE USED TO CALCULATE U       ***
C***            M:  INDEX WHICH KEEPS TRACK OF THE NUMBER OF THE PASS ***
C***                    THROUGH SUBROUTINE RUNGE THAT IS BEING MADE  ***
C***                                                                 ***
C***    PROGRAM SECTIONS:                                            ***
C***            1) THIS PART SETS UP THE DIMENSIONS OF THE VARIABLES ***
C***                    GETS THE VALUES OF THE PARAMETERS FROM 'PARAM1' ***
C***                    OPENS UP THE FILES WHERE THE FEEDBACK TENSORS ***
C***                    ARE STORED, AND SETS UP THE FILES FOR STORING ***
```

```
C***                   THE VALUES OF THE STATES.  IT ALSO ACCEPTS THE   ***
C***                   INITIAL VALUES FOR THE STATES AND READS IN THE   ***
C***                   INITIAL VALUES FOR THE FEEDBACK TENSORS.         ***
C***              2) THIS SECTION IS THE FIRST OF THREE SECTIONS INSIDE ***
C***                   THE MAIN DO LOOP.  IT INTEGRATES THE STATES      ***
C***                   USING ONLY LINEAR FEEDBACK.                      ***
C***              3) INTEGRATION USING QUADRATIC FEEDBACK               ***
C***              4) INTEGRATION USING THIRD ORDER FEEDBACK             ***
C***              5) THIS IS THE OUTPUT SECTION.  THE VALUES OF THE     ***
C***                   STATES ARE PRINTED, ALONG WITH THE STANDARD      ***
C***                   (EUCLIDEAN) NORM OF THE STATES.                  ***
C***              6) THE FILES STORING THE VALUES OF THE STATES ARE     ***
C***                   DELETED AND THEN THE PROGRAM CAN BE REPEATED     ***
C***                   IF IT IS DESIRED TO RUN IT AGAIN WITH A DIF-     ***
C***                   RENT STARTING VALUE.  THE VALUES FOR THE         ***
C***                   FEEDBACK TENSORS ARE THEN PRINTED.               ***
C***              7) THIS IS THE FORMAT SECTION.                        ***
C***                                                                    ***
C***********************************************************************
C***********************************************************************
C***********************************************************************
      SUBROUTINE XCALC
      INCLUDE 'PARAM1.FTN'
      DIMENSION DX(IDIMX),X(IDIMX),U(IDIMU),X1(IDIMX)
      DIMENSION U2(I20),X2(I2),X2S(IS2),BU(IDIMX),UX(I11)
      DIMENSION X12(IDIMX),X3(I3),X3S(IS3)
      REAL K(I11),K2S(IS12),K3S(IS13),K2X2S(IDIMU)
C***
C***  THESE ARE EXACTLY THE SAME OPEN STATEMENTS AS IN TNSCLC.
C***
      OPEN(UNIT=2,NAME='K1I.DAT',TYPE='OLD',ACCESS='DIRECT',FORM=
     &'FORMATTED',RECORDSIZE=45,CARRIAGECONTROL='LIST')
      OPEN(UNIT=4,NAME='K2SI.DAT',TYPE='OLD',ACCESS='DIRECT',FORM=
     &'FORMATTED',RECORDSIZE=67,CARRIAGECONTROL='LIST')
      OPEN(UNIT=8,NAME='K3SI.DAT',TYPE='OLD',ACCESS='DIRECT',FORM=
     &'FORMATTED',RECORDSIZE=89,CARRIAGECONTROL='LIST')
      DDT=DT*FLOAT(J11)
1     TYPE *,' ENTER THE STARTING VALUES FOR X.'
      ACCEPT *,X(1),X(2)
      DO 201 J1=1,IDIMX
      X12(J1)=X(J1)
201   X1(J1)=X(J1)
C***
C***  THE FILES WHERE THE VALUES OF THE STATES WILL BE STORED FOLLOW.
C***  THESE FILES NEED NOT BE DIRECT ACCESS SINCE THERE IS NO NEED TO
C***  BE ABLE TO ACCESS THE STORED VALUES IN ANY MANNER OTHER THAN
C***  SEQUENTIAL.
C***
      OPEN(UNIT=9,NAME='XI.DAT',TYPE='NEW',ACCESS='SEQUENTIAL')
      OPEN(UNIT=10,NAME='X1I.DAT',TYPE='NEW',ACCESS='SEQUENTIAL')
      OPEN(UNIT=11,NAME='X12I.DAT',TYPE='NEW',ACCESS='SEQUENTIAL')
      WRITE(9,*)X
      WRITE(10,*)X1
      WRITE(11,*)X12
      DO 260 J=1,IIIT-1
```

```
      IK=IIIT-J
      READ(2'IK,492)(K(J1),J1=1,I11)
      READ(4'IK,494)(K2S(J1),J1=1,IS12)
      READ(8'IK,498)(K3S(J1),J1=1,IS13)
C***
C***  SECTION 2.  THE CONTROL VARIABLE IS JUST A LINEAR FUNCTION OF THE
C***  STATES.  THE SUBROUTINE RUNGE MUST BE CALLED FOUR TIMES IN ORDER
C***  TO CALCULATE THE RUNGE-KUTTA COEFFICIENTS.
C***
      DO 210 M=1,4
      CALL TCONT(K,X,U,1,0,1,1,0,0,1,I11,IDIMX,IDIMU,IDIMU,IDIMX,2)
      DX(1)=U(2)*COSH(X(1)*X(2))-EXP(2.*U(1))*SINH(2.*X(1))
     &-3.*SINH(X(2))
      DX(2)=EXP(U(1)*U(2))*SINH(X(1))-EXP(U(1))*U(1)*COSH(X(1)**2)
     &+SINH(X(2))
      CALL RUNGE(IDIMX,X,DX,T,DDT,M)
210   CONTINUE
      WRITE(9,*)X
C***
C***  SECTION 3.  HERE THE CONTROL IS A QUADRATIC FUNCTION OF THE STATES.
C***  THE SECOND ORDER TENSOR POWER OF THE STATES MUST BE COMPUTED AND
C***  THEN SYMMETRIZED.  NOTE THAT SINCE THESE TENSOR POWERS OF THE
C***  STATES ARE CONTRAVARIANT POWERS THE SYMMETRIZATION IS DIFFERENT THAN
C***  FOR THE COVARIANT SYMMETRIZATION.
C***
      T=T-DDT
      DO 220 M=1,4
      CALL TCONT(K,X1,U,1,0,1,1,0,0,1,I11,IDIMX,IDIMU,IDIMU,IDIMX,2)
      CALL TMULT(X1,X1,X2,0,0,1,0,0,1,IDIMX,IDIMX,I2,IDIMU,IDIMX,12)
      X2S(1)=X2(1)
      X2S(2)=(X2(2)+X2(3))/2.
      X2S(3)=X2(4)
      CALL TCONT(K2S,X2S,K2X2S,1,0,1,1,0,0,1,IS12,IS2,IDIMU,IDIMU,IS2,2)
      CALL TADD(U,K2X2S,U,IDIMU)
      DX(1)=U(2)*COSH(X1(1)*X1(2))-EXP(2.*U(1))*SINH(2.*X1(1))
     &-3.*SINH(X1(2))
      DX(2)=EXP(U(1)*U(2))*SINH(X1(1))-EXP(U(1))*U(1)*COSH(X1(1)**2)
     &+SINH(X1(2))
      CALL RUNGE(IDIMX,X1,DX,T,DDT,M)
220   CONTINUE
      T=T-DDT
      WRITE(10,*)X1
C***
C***  SECTION 4.  THIS IS THE THIRD ORDER INTEGRATION.
C***
      DO 230 M=1,4
      CALL TCONT(K,X12,U,1,0,1,1,0,0,1,I11,IDIMX,IDIMU,IDIMU,IDIMX,2)
      CALL TMULT(X12,X12,X2,0,0,1,0,0,1,2,2,4,2,2,13)
      X2S(1)=X2(1)
      X2S(2)=(X2(2)+X2(3))/2.
      X2S(3)=X2(4)
      CALL TCONT(K2S,X2S,K2X2S,1,0,1,1,0,0,1,IS12,IS2,IDIMU,IDIMU,IS2,2)
      CALL TADD(U,K2X2S,U,IDIMU)
      CALL TMULT(X12,X2,X3,0,0,1,0,0,2,IDIMX,I2,I3,IDIMU,IDIMX,13)
      X3S(1)=X3(1)
```

```
      X3S(2)=(X3(2)+X3(3)+X3(5))/3.
      X3S(3)=(X3(4)+X3(6)+X3(7))/3.
      X3S(4)=X3(8)
      CALL TCONT(K3S,X3S,K2X2S,1,0,1,1,0,0,1,IS13,IS3,IDIMU,IDIMU,IS3,2)
      CALL TADD(U,K2X2S,U,IDIMU)
      DX(1)=U(2)*COSH(X12(1)*X12(2))-EXP(2.*U(1))*SINH(2.*X12(1))
     &-3.*SINH(X12(2))
      DX(2)=EXP(U(1)*U(2))*SINH(X12(1))-EXP(U(1))*U(1)*COSH(X12(1)**2)
     &+SINH(X12(2))
      CALL RUNGE(IDIMX,X12,DX,T,DDT,M)
230   CONTINUE
      WRITE(11,*)X12
260   CONTINUE
C***
C***  SECTION 5.  NEXT COMES THE OUTPUT SECTION.  ALL STORED VALUES
C***  FOR THE STATES AND THEIR NORMS ARE PRINTED.
C***
      WRITE(5,*)' FIRST ORDER VALUES FOR X.'
      WRITE(5,*)' '
      REWIND 9
      REWIND 10
      REWIND 11
      DO 262 J1=1,IIIT
      READ(9,*)X(1),X(2)
262   WRITE(5,*)J1,(X(J2),J2=1,IDIMX),'NORM X',SQRT(X(1)**2+X(2)**2)
      WRITE(5,*)' '
      WRITE(5,*)' SECOND ORDER VALUES FOR X.'
      WRITE(5,*)' '
      DO 263 J1=1,IIIT
      READ(10,*)X(1),X(2)
263   WRITE(5,*)J1,(X(J2),J2=1,IDIMX),'NORM X',SQRT(X(1)**2+X(2)**2)
      WRITE(5,*)' '
      WRITE(5,*)' THIRD ORDER VALUES FOR X.'
      WRITE(5,*)' '
      DO 264 J1=1,IIIT
      READ(11,*)X(1),X(2)
264   WRITE(5,*)J1,(X(J2),J2=1,IDIMX),'NORM X',SQRT(X(1)**2+X(2)**2)
C***
C***  SECTION 6.  THE FILES WHERE THE VALUES OF X ARE STORED ARE DELETED.
C***  IF THE PERSON RUNNING THE PROGRAM WISHES TO RUN IT AGAIN WITH A
C***  DIFFERENT STARTING VALUE FOR THE STATE VECTOR, HE ENTERS '1' FOR
C***  IAGIN.
C***
      CLOSE(UNIT=9,DISPOSE='PRINT')
      WRITE(6,*)' '
      CLOSE(UNIT=10,DISPOSE='PRINT')
      WRITE(6,*)' '
      CLOSE(UNIT=11,DISPOSE='PRINT')
      TYPE *,' IF YOU WANT TO RECALCULATE X WITH A DIFFERENT STARTING'
      TYPE *,' VALUE THEN ENTER 1.'
      ACCEPT *,IAGIN
      IF(IAGIN.EQ.1)GOTO 1
      CLOSE(UNIT=2,DISPOSE='SAVE')
      CLOSE(UNIT=4,DISPOSE='SAVE')
      CLOSE(UNIT=8,DISPOSE='SAVE')
```

```
      RETURN
C***
C***  SECTION 7,  THIS IS THE FORMAT SECTION,  THESE ARE EXACTLY THE SAME
C***  FORMAT STATEMENTS AS IN THE SUBROUTINE 'TNSCLC'.
C***
494   FORMAT(1X,6E11.4)
492   FORMAT(1X,4E11.4)
498   FORMAT(1X,8E11.4)
      END
```

```
C*****************************************************************************
C*****************************************************************************
C*****************************************************************************
C***                                                                       ***
C***            TITLE:  STAB                                               ***
C***            AUTHOR:  JOSEPH A. O'SULLIVAN                              ***
C***            DATE:  DECEMBER, 1983                                      ***
C***                                                                       ***
C***            STAB PERFORMS AN EXHAUSTIVE SEARCH OF THE PHASE PLANE      ***
C***      OF POTENTIAL INITIAL CONDITIONS TO DETERMINE THE REGION OF       ***
C***      OF USEFULNESS OF THE THREE CALCULATED FEEDBACKS:  LINEAR,        ***
C***      QUADRATIC, AND THIRD ORDER.  THE PROGRAM FIRST READS IN THE      ***
C***      STORED VALUES OF THE CONTROLLER TERMS FROM THE DATA FILES        ***
C***      WHERE THEY HAVE BEEN STORED AFTER BEING GENERATED BY THE         ***
C***      PROGRAM TNSCLC.  THE DATA FILES FROM WHICH THESE VALUES ARE      ***
C***      READ ARE THEN CLOSED (IT WAS FOUND THAT THE FEWER FILES OPEN     ***
C***      AT ANY ONE TIME THE FASTER THE COMPUTATIONS ARE PERFORMED.).     ***
C***      DATA FILES ARE THEN OPENED IN WHICH THE REGIONS WILL BE STORED.  ***
C***      THE PROCESS OF CHECKING PERFORMANCE FOR VARIOUS INITIAL CON-     ***
C***      DITIONS IS THEN BEGUN.  FOR EACH INITIAL CONDITION VECTOR,       ***
C***      A MODIFIED VERSION OF THE SUBROUTINE XCALC IS RUN.  THIS         ***
C***      VERSION OF XCALC HAS SEVERAL FEATURES ADAPTED ESPECIALLY FOR     ***
C***      THIS USE.  IF THERE IS AN OVERFLOW IN A CALCULATION FOR ONE OF   ***
C***      THE STATES THEN THE LOGICAL VECTOR INDDX IS SET TO FALSE FOR     ***
C***      THE ORDER OF THE FEEDBACK IN WHICH THE OVERFLOW OCCURRED.        ***
C***      THIS POINT IS DEEMED OUT OF THE REGION OF USEFULNESS.  SIMI-     ***
C***      LARLY, IF THE FINAL VALUE OF THE STATE IS LARGER THAN SOME       ***
C***      THRESHHOLD VALUE, INDDX IS SET TO FALSE FOR THAT FEEDBACK ORDER  ***
C***      AND THE POINT IS DEEMED OUT OF THE REGION OF USEFULNESS.  AFTER  ***
C***      RETURNING TO THIS PROGRAM, THE ELEMENTS OF THE THREE VECTORS     ***
C***      LIN, QUAD, AND THRD ARE SET TO '-' OR '*' ACCORDING TO WHETHER   ***
C***      THE INITIAL CONDITION IS OUT OF OR IS IN THE REGION OF USEFUL-   ***
C***      NESS, RESPECTIVELY.  THESE VECTORS ARE THEN STORED IN THE RES-   ***
C***      PECTIVE DATA FILES, AND THE FILES ARE SAVED.                     ***
C***                                                                       ***
C***      IMPORTANT VARIABLES:                                             ***
C***            K11:  VECTOR OF LINEAR FEEDBACK TENSORS.  THE VALUES       ***
C***                  FOR THIS ARE READ FROM K1I.DAT, A FILE CREATED       ***
C***                  BY TNSCLC.                                           ***
C***            K2SI:  VECTOR OF QUADRATIC FEEDBACK TENSORS.  ITS VAL-     ***
C***                  UES ARE READ FROM K2SI.DAT; COMPUTED BY TNSCLC.      ***
C***            K3SI:  THIRD ORDER FEEDBACK TENSORS FROM K3SI.DAT; COM-    ***
C***                  PUTED BY TNSCLC.                                     ***
C***            LIN, QUAD, THRD:  THE DATA PUT INTO LINX.DAT, QUADX.DAT,   ***
C***                  AND THRDX.DAT, FOR THE LINEAR, QUADRATIC, AND        ***
C***                  THIRD ORDER REGIONS OF USEFULNESS, RESPEC-           ***
C***                  TIVELY.  EACH VECTOR CORRESPONDS TO ALL POS-         ***
C***                  SIBLE VALUES OF X(1) FOR FIXED VALUES OF X(2).       ***
C***            X:  THE STATE VECTOR.  THIS IS USED TO INPUT DIFFERENT     ***
C***                  INITIAL CONDITIONS INTO XCALC.                       ***
C***            INDDX:  THREE DIMENSIONAL LOGICAL VARIABLE.  THE VALUES    ***
C***                  OF THIS VECTOR ARE SET TO TRUE OR FALSE DEPEN-       ***
C***                  DING UPON WHETHER OR NOT EACH INITIAL CONDITION      ***
C***                  IS IN THE REGION OF USEFULNESS FOR LINEAR,           ***
```

```
C***                    QUADRATIC, AND THIRD ORDER FEEDBACK, RESPEC-     ***
C***                    TIVELY.                                          ***
C***                                                                     ***
C***********************************************************************
C***********************************************************************
C***********************************************************************
      DIMENSION INDERR(3),X(2)
      LOGICAL INDDX(3)
      INTEGER LIN(71),QUAD(71),THRD(71)
      REAL K1I(400,4),K2SI(400,6),K3SI(400,8)
      OPEN(UNIT=1,NAME='K1I.DAT',TYPE='OLD',ACCESS='DIRECT',FORM=
     &'FORMATTED',RECORDSIZE=45,CARRIAGECONTROL='LIST')
      OPEN(UNIT=2,NAME='K2SI.DAT',TYPE='OLD',ACCESS='DIRECT',FORM=
     &'FORMATTED',RECORDSIZE=67,CARRIAGECONTROL='LIST')
      OPEN(UNIT=3,NAME='K3SI.DAT',TYPE='OLD',ACCESS='DIRECT',FORM=
     &'FORMATTED',RECORDSIZE=89,CARRIAGECONTROL='LIST')
      DO 10 J1=1,400
      J3=501-J1
      READ(1'J3,105)(K1I(J1,J2),J2=1,4)
      READ(2'J3,106)(K2SI(J1,J2),J2=1,6)
      READ(3'J3,107)(K3SI(J1,J2),J2=1,8)
10    CONTINUE
      CLOSE(UNIT=1,DISPOSE='SAVE')
      CLOSE(UNIT=2,DISPOSE='SAVE')
      CLOSE(UNIT=3,DISPOSE='SAVE')
      OPEN(UNIT=1,NAME='LINX.DAT',TYPE='NEW',ACCESS='DIRECT',
     &RECORDSIZE=142,CARRIAGECONTROL='LIST',FORM='FORMATTED')
      OPEN(UNIT=2,NAME='QUADX.DAT',TYPE='NEW',ACCESS='DIRECT',
     &RECORDSIZE=142,CARRIAGECONTROL='LIST',FORM='FORMATTED')
      OPEN(UNIT=3,NAME='THRDX.DAT',TYPE='NEW',ACCESS='DIRECT',
     &RECORDSIZE=142,CARRIAGECONTROL='LIST',FORM='FORMATTED')
1     DO 50 J1=1,61
      DO 45 J2=1,71
      X(1)=FLOAT(J2-41)/10.
      X(2)=FLOAT(31-J1)/10.
      DO 2 I=1,3
2     INDDX(I)=.FALSE.
      CALL XCALC(X,INDDX,INDERR,K1I,K2SI,K3SI)
      IF(INDDX(1))GO TO 30
      LIN(J2)='*'
      GO TO 31
30    LIN(J2)='-'
31    IF(INDDX(2))GO TO 35
      QUAD(J2)='*'
      GO TO 36
35    QUAD(J2)='-'
36    IF(INDDX(3))GO TO 40
      THRD(J2)='*'
      GO TO 41
40    THRD(J2)='-'
41    CONTINUE
45    CONTINUE
      WRITE(1'J1,101)LIN
      WRITE(2'J1,101)QUAD
      WRITE(3'J1,101)THRD
```

```
50      CONTINUE
        CLOSE(UNIT=1,DISPOSE='SAVE')
        CLOSE(UNIT=2,DISPOSE='SAVE')
        CLOSE(UNIT=3,DISPOSE='SAVE')
        STOP
101     FORMAT(71A2)
105     FORMAT(1X,4E11.4)
106     FORMAT(1X,6E11.4)
107     FORMAT(1X,8E11.4)
        END
```

```
C***********************************************************************
C***********************************************************************
C***********************************************************************
C***                                                                 ***
C***            TITLE:  TMULT                                        ***
C***            AUTHOR:  JOSEPH A. O'SULLIVAN                        ***
C***                                                                 ***
C***            SUBROUTINE TMULT GENERATES THE TENSOR PRODUCT OF TWO ***
C***    TENSORS.  THE RESULTANT TENSOR HAS DIMENSION EQUAL TO THE PRO-***
C***    DUCT OF THE TWO GENERATING TENSORS.  IT IS ASSUMED THAT THE  ***
C***    TENSORS HAVE CONTRAVARIANT COMPONENTS OF THE STATE           ***
C***    VARIABLE ONLY.  IT IS ALSO ASSUMED THAT THE TENSORS ARE NOT  ***
C***    IN REDUCED SYMMETRIC FORM.                                   ***
C***            EACH ELEMENT IN C, THE RESULTANT TENSOR, IS A PRODUCT ***
C***    OF AN ELEMENT FROM A TIMES AN ELEMENT FROM B.  ALL POSSIBLE  ***
C***    PRODUCTS OF AN ELEMENT FROM A AND ONE FROM B APPEAR IN C, EACH ***
C***    PRODUCT APPEARING ONCE.  LEXICOGRAPHIC ORDERING DETERMINES THE ***
C***    METHOD OF CHOOSING WHICH PRODUCT IS IN WHICH ENTRY OF C.  FIRST ***
C***    THE COVARIANT POWERS OF THE STATE VARIABLE IN B ARE          ***
C***    INCREMENTED, THEN THE COVARIANT POWERS OF THE STATE VARIABLE ***
C***    IN A, THEN THE COVARIANT POWERS OF THE CONTROL VARIABLE--    ***
C***    FIRST IN B THEN A--THEN THE CONTRAVARIANT POWERS IN B AND A. ***
C***                                                                 ***
C***    CALLING VARIABLES:                                           ***
C***            A:  INPUT TENSOR                                     ***
C***            B:  INPUT TENSOR                                     ***
C***            C:  OUTPUT TENSOR, THE TENSOR PRODUCT OF A AND B     ***
C***            A1:  THE CONTRAVARIANT POWER OF THE STATE            ***
C***                    VARIABLE IN A                                ***
C***            AP:  THE COVARIANT POWER OF THE CONTROL VARIABLE IN A ***
C***            AQ:  THE COVARIANT POWER OF THE STATE VARIABLE IN A  ***
C***            B1:  THE CONTRAVARIANT POWER OF THE STATE            ***
C***                    VARIABLE IN B                                ***
C***            BP:  THE COVARIANT POWER OF THE CONTROL VARIABLE IN B ***
C***            BQ:  THE COVARIANT POWER OF THE STATE VARIABLE IN B  ***
C***            DIMA:  THE DIMENSION OF A                            ***
C***            DIMB:  THE DIMENSION OF B                            ***
C***            DIMC:  THE DIMENSION OF C                            ***
C***            DIMU:  THE DIMENSION OF THE CONTROL VARIABLE         ***
C***            DIMX:  THE DIMENSION OF THE STATE VARIABLE           ***
C***            NCALL:  A DEBUGGING AID; IF THIS SUBROUTINE IS CALLED ***
C***                    MANY TIMES OVER THE COURSE OF A PROGRAM, THEN ***
C***                    ONE WITH AN ERROR CAN BE UNIQUELY IDENTIFIED ***
C***                    USING THIS VARIABLE.                         ***
C***                                                                 ***
C***    IMPORTANT VARIABLES:                                         ***
C***            NA, NB, NC:  THE DIMENSIONS OF A, B, AND C, RESPEC-  ***
C***                    TIVELY, AS CALCULATED FROM THE OTHER INFOR-  ***
C***                    MATION (DIMU, DIMX, A1, AP, AQ, B1, BP, BQ)  ***
C***            J:  THE ELEMENT OF C WHICH IS BEING CALCULATED AT EACH ***
C***                    STEP                                         ***
C***            JA, JB:  THE ELEMENTS OF A AND B, RESPECTIVELY, WHICH ***
C***                    ARE BEING MULTIPLIED TOGETHER AT EACH STEP   ***
```

```
C***             N1, N2:  THE DIMENSIONS OF THE CONTRAVARIANT PARTS OF    ***
C***                  A AND B, RESPECTIVELY                               ***
C***             N3, N4:  THE DIMENSIONS OF THE PARTS OF A AND B,         ***
C***                  RESPECTIVELY, WHICH ARE COVARIANT IN THE            ***
C***                  CONTROL VARIABLE                                    ***
C***             N5, N6:  THE DIMENSIONS OF THE PARTS OF A AND B,         ***
C***                  RESPECTIVELY, WHICH ARE COVARIANT IN THE            ***
C***                  STATE VARIABLE                                      ***
C***                                                                      ***
C*************************************************************************
C*************************************************************************
C*************************************************************************
      SUBROUTINE TMULT(A,B,C,A1,AP,AQ,B1,BP,BQ,DIMA,DIMB,DIMC,
     1 DIMU,DIMX,NCALL)
      INTEGER A1,AP,AQ,BQ,B1,BP,DIMA,DIMB,DIMC,DIMU,DIMX
      DIMENSION A(DIMA),B(DIMB),C(DIMC)
C***
C***  FIRST THE PROGRAM CHECKS TO MAKE SURE THAT THE DIMENSIONS AND
C***  POWERS GIVEN AGREE WITH EACH OTHER.  IF THEY DISAGREE, THE
C***  PROGRAM IS TERMINATED.
C***
      NA=DIMX**(A1+AQ)*DIMU**AP
      IF(NA.EQ.DIMA)GO TO 5
2     TYPE *,' ERROR IN TMULT. DIMS GIVEN DO NOT AGREE, NCALL=',NCALL
      STOP
5     NB=DIMX**(B1+BQ)*DIMU**BP
      IF(NB.EQ.DIMB)GO TO 10
      GO TO 2
10    NC=NA*NB
      IF(NC.EQ.DIMC) GO TO 15
      GO TO 2
C***
C***  N1, N2, N3, N4, N5, AND N6 ARE CALCULATED TO GIVE THE CONTRAVARIANT
C***  AND COVARIANT PARTS' DIMENSIONS FOR A AND B SO THAT THE DO LOOPS
C***  CAN BE SET UP IN SUCH A WAY AS TO FACILITATE CALCULATING JA AND JB.
C***  THE SIX DO LOOPS ARE SET UP SO THAT THEY INCREMENT IN THE SAME
C***  MANNER AS C.  THUS, J IS INITIALLY SET TO ZERO AND THEN INCREMENTED
C***  EACH TIME THROUGH THE INNERMOST LOOP TO KEEP TRACK OF WHICH
C***  ELEMENT OF C IS BEING CALCULATED AT EACH STEP.  A IS INCREMENTED
C***  IN THE SAME MANNER AS I1, I3, AND I5, BUT BECAUSE OF THE OTHER
C***  DO LOOPS IT MUST BE CALCULATED EACH TIME THROUGH THE SECOND LAST
C***  LOOP.  THE LEXICOGRAPHIC ORDERING FOR B IS THE SAME AS THE INCRE-
C***  MENTATION BY THE LOOPS FOR I2, I4, AND I6.  JB MUST ALSO BE CAL-
C***  CULATED EACH TIME THROUGH THE SECOND LAST LOOP, BUT SOME CALCU-
C***  LATIONS ARE SAVED BY NOT PUTTING THAT CALCULATION INSIDE THE FINAL
C***  LOOP.
C***
15    J=0
      N1=DIMX**A1
      N2=DIMX**B1
      N3=DIMU**AP
      N4=DIMU**BP
      N5=DIMX**AQ
      N6=DIMX**BQ
      DO 20 I1=1,N1
```

```
DO 20 I2=1,N2
DO 20 I3=1,N3
DO 20 I4=1,N4
DO 20 I5=1,N5
JA=I5+(I3-1)*N5+(I1-1)*N5*N3
JB=(I4-1)*N6+(I2-1)*N6*N4
DO 20 I6=1,N6
J=J+1
JB=JB+1
C(J)=A(JA)*B(JB)
RETURN
END
```

```
C***********************************************************************
C***********************************************************************
C***********************************************************************
C***                                                                 ***
C***           TITLE:  TCONT                                         ***
C***           AUTHOR:  JOSEPH A. O'SULLIVAN                         ***
C***                                                                 ***
C***           SUBROUTINE TCONT PERFORMS CONTRACTIONS ON TENSORS.    ***
C***     IT TAKES TWO TENSORS, ONE OF WHICH (B) HAS AT LEAST NCONT CON-  ***
C***     TRAVARIANT POWERS OF ONE OF THE VARIABLES (CONTROL OR STATE),   ***
C***     THE OTHER OF WHICH (A) HAS AT LEAST NCONT COVARIANT POWERS  ***
C***     OF THE SAME VARIABLE, CONTRACTS THE TWO OVER THOSE NCONT    ***
C***     POWERS AND THEREBY FORMS C.                                 ***
C***                                                                 ***
C***     CALLING VARIABLES:                                          ***
C***           A:  INPUT TENSOR                                      ***
C***           B:  INPUT TENSOR                                      ***
C***           C:  OUTPUT TENSOR                                     ***
C***           A1:  CONTRAVARIANT POWER OF A                         ***
C***           AP:  COVARIANT POWER OF THE CONTROL VARIABLE IN A     ***
C***           AQ:  COVARIANT POWER OF STATE VARIABLE IN A           ***
C***           B1:  CONTRAVARIANT POWER OF B                         ***
C***           BP:  COVARIANT POWER OF THE CONTROL VARIABLE IN B     ***
C***           BQ:  COVARIANT POWER OF STATE VARIABLE IN B           ***
C***           NCONT:  NUMBER OF CONTRACTIONS TO BE PERFORMED        ***
C***           DIMA:  DIMENSION OF A                                 ***
C***           DIMB:  DIMENSION OF B                                 ***
C***           DIMC:  DIMENSION OF C                                 ***
C***           DIMU:  DIMENSION OF THE CONTROL VARIABLE              ***
C***           DIMX:  DIMENSION OF THE STATE VARIABLE                ***
C***           NTYPE:  TYPE OF CONTRACTION PERFORMED                 ***
C***                                                                 ***
C***     TYPES OF CONTRACTIONS:                                      ***
C***                    CONTRACTION     CONTRAVA-        CONTRAVA-   ***
C***           NTYPE       OVER      RIANT VAR. IN A  RIANT VAR. IN B ***
C***           -----    -----------  ---------------  --------------- ***
C***             1          X               X                X       ***
C***             2          X               U                X       ***
C***             3          U               X                U       ***
C***             4          U               U                U       ***
C***            10      - - - - - - - -  SPECIAL  - - - - - - - - - - ***
C***                                                                 ***
C***           HERE, X=STATE VARIABLE, U=CONTROL VARIABLE            ***
C***                                                                 ***
C***     NTYPE=10 IS GENERALLY USED IN CIRCUMSTANCES WHERE THE CON-  ***
C***     TRACTION IS OVER BOTH CONTROL AND STATE VARIABLES.          ***
C***     THIS IS NECESSARY IN SEVERAL IMPORTANT INSTANCES.  IN THIS  ***
C***     CASE, THE FOLLOWING MUST BE DONE:                           ***
C***           B1=AQ:  THE CONTRACTION IS DONE OVER THIS PART OF THE ***
C***                   TENSORS.  OFTEN, THEY ARE EQUAL TO ONE, AND   ***
C***                   DIMU IS EQUAL TO THE DIMENSION OF THE PART THAT ***
C***                   THE CONTRACTION IS DONE OVER.                 ***
C***           AP=0                                                  ***
C***           BP=0                                                  ***
```

```
C***          A1=DIMA/(AQ**DIMU):  THIS IS THE PART OF A OVER WHICH    ***
C***                  THE CONTRACTION WILL NOT RANGE.  IT INCLUDES     ***
C***                  THE DIMENSION OF WHATEVER CONTRAVARIANT PART     ***
C***                  THAT THERE IS TIMES THE DIMENSION OF THE         ***
C***                  REMAINDER OF THE COVARIANT PART OVER WHICH       ***
C***                  THERE IS NO CONTRACTION.  THEN DIMA=A1*AQ**DIMU.***
C***          DIMX**BQ=DIMB/(DIMU**B1):  THIS IS THE PART OF TENSOR    ***
C***                  B OVER WHICH THERE IS NO CONTRACTION.  MORE      ***
C***                  FREEDOM IS ALLOWED WITH THIS PART IN B THAN A.   ***
C***          DIMU:  THIS IS AS NOTED THE DIMENSION OF THE 'VARIABLE'  ***
C***                  OVER WHICH THE CONTRACTION RUNS.                 ***
C***                                                                   ***
C***    IMPORTANT VARIABLES:                                           ***
C***          IDA, IDB, IDC:  THE DIMENSIONS OF A, B, AND C RESPEC-    ***
C***                  TIVELY WHICH ARE CALCULATED FROM THE OTHER       ***
C***                  INFORMATION GIVEN (DIMU, DIMX, NCONT, A1, AP,    ***
C***                  AQ, B1, BP, BQ)                                  ***
C***          I7:  DIMENSION OF THE PART OF THE TENSORS OVER WHICH     ***
C***                  THE CONTRACTION IS DONE                          ***
C***          I1*I3*I5:  DIMENSION OF THE PART OF A OVER WHICH THERE   ***
C***                  IS NO CONTRACTION                                ***
C***          I2*I4*I6:  DIMENSION OF THE PART OF B OVER WHICH THERE   ***
C***                  IS NO CONTRACTION                                ***
C***          M1:  CALCULATED VALUE FOR A1, NTYPE=10                   ***
C***              COMPONENT OF C WHICH IS BEING CALCULATED AT EACH     ***
C***                  STEP                                             ***
C***          JA:  ELEMENT IN A IN CALCULATION AT EACH STEP            ***
C***          JB:  ELEMENT IN B IN CALCULATION AT EACH STEP            ***
C***                                                                   ***
C***    TCONT IS A BILINEAR MAPPING FROM (A,B) TO C:                   ***
C***           A1      B1      -----\     (A1+B1-NCONT)                ***
C***           A       *B      -----/    C                             ***
C***           AP,AQ   BP,BQ              (.),(.)                      ***
C***                                                                   ***
C***          WHERE THE COVARIANT POWERS IN C DEPEND ON NTYPE.         ***
C***          * JUST STANDS FOR THE TENSOR PRODUCT.                    ***
C***                                                                   ***
C**********************************************************************
C**********************************************************************
C**********************************************************************
      SUBROUTINE TCONT(A,B,C,A1,AP,AQ,B1,BP,BQ,NCONT,DIMA,DIMB,DIMC,DI
     1MU,DIMX,NTYPE)
      INTEGER A1,AP,AQ,B1,BP,BQ,DIMA,DIMB,DIMC,DIMU,DIMX
      DIMENSION A(DIMA),B(DIMB),C(DIMC)
      GO TO (1,2,3,4),NTYPE
      IF(NTYPE.EQ.10)GO TO 1000
      TYPE *,' NTYPE WAS TOO BIG, SUB TCONT. NTYPE WAS ',NTYPE
      STOP
C***
C***    NTYPE=1.  DIMA AND DIMB ARE CHECKED AGAINST IDA AND IDB,
C***    RESPECTIVELY; IF EITHER DISAGREES, THE PROGRAM IS TERMINATED
C***
1       IDA=DIMX**(A1+AQ)*DIMU**AP
        IDB=DIMX**(B1+BQ)*DIMU**BP
        IF(IDA.EQ.DIMA)GOTO 101
```

```
10      TYPE *,' ERROR IN TCONT. DIMA WRONG, NTYPE=',NTYPE
        STOP
101     IF(IDB.EQ.DIMB)GO TO 102
11      TYPE *,' ERROR TCONT. DIMB WRONG, NTYPE=',NTYPE
        STOP
102     IF(NCONT.LE.AQ.AND.NCONT.LE.B1)GOTO 103
12      TYPE *,' ERROR IN SUB TCONT. NCONT TOO LARGE, NTYPE=',NTYPE
        STOP
103     I7=DIMX**NCONT
        I5=DIMX**(AQ-NCONT)
        I2=DIMX**(B1-NCONT)
        I4=DIMU**BP
        I1=DIMX**A1
        I3=DIMU**AP
        I6=DIMX**BQ
        GO TO 500
2       IDA=DIMX**AQ*DIMU**(AP+A1)
        IDB=DIMX**(B1+BQ)*DIMU**BP
        IF(IDA.EQ.DIMA) GO TO201
        GO TO 10
201     IF(IDB.EQ.DIMB)GOTO 202
        GO TO 11
202     IF(NCONT.LE.AQ.AND.NCONT.LE.B1)GOTO 203
        GO TO 12
203     I1=DIMU**A1
        I2=DIMX**(B1-NCONT)
        I3=DIMU**AP
        I4=DIMU**BP
        I5=DIMX**(AQ-NCONT)
        I6=DIMX**BQ
        I7=DIMX**NCONT
        GOTO 500
3       IDA=DIMX**A1*DIMU**AP
4       IDB=DIMU**(B1+BP)*DIMX**BQ
C***
C***    NTYPE=3 CORRESPONDS TO A CONTRACTION OVER THE CONTROL VARIABLE WITH
C***    A HAVING CONTRAVARIANT POWERS IN THE STATE VARIABLE.  NTYPE=4
C***    CORRESPONDS TO A CONTRACTION OVER THE CONTROL VARIABLE WITH A HAVING
C***    CONTRAVARIANT POWERS IN THE CONTROL VARIABLE.  FOR EITHER OF THESE
C***    TWO TYPES, AQ MUST BE ZERO.  AFTER CHECKING TO SEE IF THE DIMENSIONS
C***    OF A AND B AGREE WITH THE OTHER DIMENSIONS, I1-I7 ARE CALCULATED
C***    AND THEN CONTROL OF THE PROGRAM IS PASSED TO LINE 500.
C***
        IF(NTYPE.EQ.4)IDA=DIMU**(A1+AP)
        IF(IDA.EQ.DIMA)GOTO 301
        GO TO 10
301     IF(IDB.EQ.DIMB)GOTO 302
        GO TO 11
302     IF(NCONT.LE.AP.AND.NCONT.LE.B1)GOTO 303
        GO TO 12
303     I7=DIMU**NCONT
        IF(NTYPE.EQ.3)I1=DIMX**A1
        IF(NTYPE.EQ.4)I1=DIMU**A1
        I2=DIMU**(B1-NCONT)
        I3=DIMU**(AP-NCONT)
```

```
      I4=DIMU**BP
      I5=[illegible]
      I6=DIMX**BQ
      IF(AQ.EQ.0)GOTO 500
      TYPE *,' ERROR SUB TCONT, LINE 303. AQ SHOULD BE 0, NTYPE=',NTYPE
      STOP
C***
C***  AFTER CHECKING THE DIMENSION OF THE THIRD TENSOR AGAINST THE DIMEN-
C***  SIONS GIVEN FOR A AND B, THE OUTPUT TENSOR IS CALCULATED.  THE DO
C***  LOOPS ARE SET UP SO THAT THE FIRST SIX INCREMENT IN THE SAME MANNER
C***  AS C.  THE LAST DO LOOP RANGES OVER THE DIMENSION OF THE PART OF
C***  THE TENSORS WHICH IS BEING CONTRACTED.  J IS INITIALIZED AT ZERO
C***  AND THEN INCREMENTED EACH TIME THROUGH THE SECOND LAST DO LOOP.
C***
500   IDC=I1*I2*I3*I4*I5*I6
      IF(IDC.EQ.DIMC)GOTO 501
      TYPE *,' ERROR SUB TCONT. LINE 500. DIM THIRD TENSOR WRONG, NTYPE=',
     & NTYPE
      STOP
501   J=0
      DO 502 K=1,DIMC
502   C(K)=0.0
      DO 505 J1=1,I1
      DO 505 J2=1,I2
      DO 505 J3=1,I3
      DO 505 J4=1,I4
      DO 505 J5=1,I5
      DO 505 J6=1,I6
      J=J+1
      DO 505 J7=1,I7
      JA=J7+I7*(J5-1+I5*(J3-1+I3*(J1-1)))
      JB=J6+I6*(J4-1+I4*(J2-1+I2*(J7-1)))
505   C(J)=C(J)+A(JA)*B(JB)
      RETURN
C***
C***  NTYPE=10 IS THE SPECIAL TYPE OF CONTRACTION, USED OFTEN WHEN THE
C***  CONTRACTION IS OVER A COMBINATION OF STATE AND CONTROL
C***  VARIABLES.  FIRST, THE CONDITIONS FOR USING NTYPE=10 MUST BE
C***  SATISFIED.  AS NOTED AT THE BEGINNING OF THIS PROGRAM, THESE
C***  INCLUDE:  AP=0, AQ=B1, BP=0, A1=DIMA/DIMU**AQ, BQ**DIMX=DIMB/DIMU**B1

C***  THE DIMENSION OF C IS THEN CHECKED AGAINST THE OTHER INFORMATION
C***  GIVEN TO MAKE SURE IT AGREES.  IF ANY OF THESE CONDITIONS ARE
C***  VIOLATED, THEN THE PROGRAM IS TERMINATED.  THERE ARE SOME THINGS
C***  WHICH THIS PROGRAM ASSUMES WHEN IT CALCULATES C FOR NTYPE=10.
C***  THE BIGGEST ASSUMPTION IS THAT THE METHOD OF INCREMENTING THE
C***  BASIS ELEMENTS IN C WILL BE TO INCREMENT THOSE THAT COME FROM
C***  B FIRST, THEN INCREMENT THOSE THAT COME FROM A.  THIS DOES
C***  NOT POSE A PROBLEM IN GENERAL, BECAUSE USUALLY ALL OF THE CONTRA-
C***  VARIANT POWERS OF B OR ALL OF THE COVARIANT POWERS OF A ARE INVOLVED
C***  IN THE CONTRACTION.  IF BOTH OF THESE ARE NOT TRUE, THEN THE
C***  TENSOR WHICH IS CALCULATED HERE WILL HAVE TO UNDERGO SOME PERMUTATION
C***  TO OBTAIN THE CORRECT TENSOR WHICH WILL HAVE ITS BASIS ELEMENTS
C***  IN THE STANDARD ORDER.
C***
```

```
1000   IF(AP.NE.0)GOTO 1100
       IF(AQ.NE.B1)GOTO 1100
       IF(BP.NE.0)GOTO 1100
       M1=IFIX(FLOAT(DIMA)/FLOAT(DIMU**AQ)+.001)
       IF(A1.NE.M1)GO TO 1200
1002   DO 1001 J=1,DIMC
1001   C(J)=0.0
C***
C***   J WILL NOW KEEP TRACK OF WHICH ELEMENT OF C IS BEING COMPUTED AT
C***   EACH STEP THROUGH THE DO LOOPS.  THE FIRST TWO DO LOOPS ARE SET UP
C***   SO THAT THEY INCREMENT IN THE SAME WAY AS C, THUS J IS SET TO ZERO
C***   AND IS INCREMENTED EACH TIME THROUGH THE SECOND DO LOOP.
C***
       J=0
       DO 1010 J1=1,M1
       DO 1010 J2=1,DIMX**BQ
       J=J+1
       DO 1010 J3=1,DIMU**B1
       JA=(J1-1)*DIMU**AQ+J3
       JB=(J3-1)*DIMX**BQ+J2
1010   C(J)=C(J)+A(JA)*B(JB)
       RETURN
1100   TYPE *,' THE DIMENSIONS IN TCONT ARE NOT CORRECT FOR NTYPE=10.'
       STOP
1200   TYPE *,' ERROR SUB TCONT. A1 IS WRONG, NTYPE=10.'
       STOP
       END
```

```
C*************************************************************************
C*************************************************************************
C*************************************************************************
C***                                                                   ***
C***            TITLE:  TCONT1                                         ***
C***            AUTHOR:  JOSEPH A. O'SULLIVAN                          ***
C***            DATE:  SEPTEMBER, 1983                                 ***
C***                                                                   ***
C***            SUBROUTINE TCONT1 IS ALMOST IDENTICAL TO SUBROUTINE    ***
C***    TCONT.  THE ARGUMENTS FOR THIS SUBROUTINE ARE EXACTLY THE SAME ***
C***    AS THOSE FOR TCONT EXCEPT FOR NTY AND SCAL.  THE DIFFERENCES   ***
C***    ARE:                                                           ***
C***            NTY:  NTY CAN BE LESS THAN ZERO.  THE ABSOLUTE VALUE OF ***
C***                  NTY DETERMINES THE TYPE OF CONTRACTION PERFOR-   ***
C***                  MED.  IF NTY IS LESS THAN ZERO, THEN THE OUTPUT ***
C***                  TENSOR IS NOT SET TO ZERO BEFORE THE CONTRAC-    ***
C***                  TION IS PERFORMED.  THUS, THE RESULTING TENSOR   ***
C***                  WILL BE THE SUM OF THE OUTPUT TENSOR'S ORIGINAL ***
C***                  VALUE PLUS THE VALUE OF THE RESULT OF THE CON-   ***
C***                  TRACTION.                                        ***
C***                                                                   ***
C***            SCAL:  THIS IS A SCALAR WHICH CAN MULTIPLY THE RESULT   ***
C***                   OF THE CONTRACTION.  IT MULTIPLIES THE RESULT   ***
C***                   BEFORE IT IS ADDED TO THE OLD VALUE OF THE      ***
C***                   OUTPUT TENSOR.  SO IF NTY<0 THEN                ***
C***                             C=C+SCAL*A@B.                         ***
C***                   IF SCAL=1 AND NTY>0, THEN THE RESULT OF USING   ***
C***                   TCONT1 WILL BE THE SAME AS USING TCONT.         ***
C***                                                                   ***
C*************************************************************************
C*************************************************************************
C*************************************************************************
        SUBROUTINE TCONT1(A,B,C,A1,AP,AQ,B1,BP,BQ,NCONT,DIMA,DIMB,DIMC,DI
     1MU,DIMX,NTY,SCAL)
        INTEGER A1,AP,AQ,B1,BP,BQ:DIMA,DIMB,DIMC,DIMU,DIMX
        DIMENSION A(DIMA),B(DIMB),C(DIMC)
        NTYPE=IABS(NTY)
        GO TO (1,2,3,4),NTYPE
        IF(NTYPE.EQ.10)GO TO 1000
        TYPE *,' NTYPE WAS TOO BIG, SUB TCONT. NTYPE WAS ',NTYPE
        STOP
C***
C***    NTYPE=1.  DIMA AND DIMB ARE CHECKED AGAINST IDA AND IDB,
C***    RESPECTIVELY; IF EITHER DISAGREES, THE PROGRAM IS TERMINATED
C***
1       IDA=DIMX**(A1+AQ)*DIMU**AP
        IDB=DIMX**(B1+BQ)*DIMU**BP
        IF(IDA.EQ.DIMA)GOTO 101
10      TYPE *,' ERROR IN TCONT. DIMA WRONG, NTYPE=',NTYPE
        STOP
101     IF(IDB.EQ.DIMB)GO TO 102
11      TYPE *,' ERROR TCONT. DIMB WRONG, NTYPE=',NTYPE
        STOP
102     IF(NCONT.LE.AQ.AND.NCONT.LE.B1)GOTO 103
```

```
12      TYPE *,' ERROR IN SUB TCONT, NCONT TOO LARGE, NTYPE=',NTYPE
        STOP
103     I7=DIMX**NCONT
        I5=DIMX**(AQ-NCONT)
        I2=DIMX**(B1-NCONT)
        I4=DIMU**BP
        I1=DIMX**A1
        I3=DIMU**AP
        I6=DIMX**BQ
        GO TO 500
2       IDA=DIMX**AQ*DIMU**(AP+A1)
        IDB=DIMX**(B1+BQ)*DIMU**BP
        IF(IDA.EQ.DIMA) GO TO201
        GO TO 10
201     IF(IDB.EQ.DIMB)GOTO 202
        GO TO 11
202     IF(NCONT.LE.AQ.AND.NCONT.LE.B1)GOTO 203
        GO TO 12
203     I1=DIMU**A1
        I2=DIMX**(B1-NCONT)
        I3=DIMU**AP
        I4=DIMU**BP
        I5=DIMX**(AQ-NCONT)
        I6=DIMX**BQ
        I7=DIMX**NCONT
        GOTO 500
3       IDA=DIMX**A1*DIMU**AP
4       IDB=DIMU**(B1+BP)*DIMX**BQ
C***
C***    NTYPE=3 CORRESPONDS TO A CONTRACTION OVER THE CONTROL VARIABLE WITH
C***    A HAVING CONTRAVARIANT POWERS IN THE INDEPENDENT VARIABLE.  NTYPE=4
C***    CORRESPONDS TO A CONTRACTION OVER THE CONTROL VARIABLE WITH A HAVING
C***    CONTRAVARIANT POWERS IN THE CONTROL VARIABLE.  FOR EITHER OF THESE
C***    TWO TYPES, AQ MUST BE ZERO.  AFTER CHECKING TO SEE IF THE DIMENSIONS
C***    OF A AND B AGREE WITH THE OTHER DIMENSIONS, I1-I7 ARE CALCULATED
C***    AND THEN CONTROL OF THE PROGRAM IS PASSED TO LINE 500.
C***
        IF(NTYPE.EQ.4)IDA=DIMU**(A1+AP)
        IF(IDA.EQ.DIMA)GOTO 301
        GO TO 10
301     IF(IDB.EQ.DIMB)GOTO 302
        GO TO 11
302     IF(NCONT.LE.AP.AND.NCONT.LE.B1)GOTO 303
        GO TO 12
303     I7=DIMU**NCONT
        IF(NTYPE.EQ.3)I1=DIMX**A1
        IF(NTYPE.EQ.4)I1=DIMU**A1
        I2=DIMU**(B1-NCONT)
        I3=DIMU**(AP-NCONT)
        I4=DIMU**BP
        I5=1
        I6=DIMX**BQ
        IF(AQ.EQ.0)GOTO 500
        TYPE *,' ERROR SUB TCONT, LINE 303. AQ SHOULD BE 0, NTYPE=',NTYPE
        STOP
```

```
C***
C***    AFTER CHECKING THE DIMENSION OF THE THIRD TENSOR AGAINST THE DIMEN-
C***    SIONS GIVEN FOR A AND B, THE OUTPUT TENSOR IS CALCULATED.  THE DO
C***    LOOPS ARE SET UP SO THAT THE FIRST SIX INCREMENT IN THE SAME MANNER
C***    AS C.  THE LAST DO LOOP RANGES OVER THE DIMENSION OF THE PART OF
C***    THE TENSORS WHICH IS BEING CONTRACTED.  J IS INITIALIZED AT ZERO
C***    AND THEN INCREMENTED EACH TIME THROUGH THE SECOND LAST DO LOOP.
C***
500     IDC=I1*I2*I3*I4*I5*I6
        IF(IDC.EQ.DIMC)GOTO 501
        TYPE *,' ERROR SUB TCONT. LINE 500. DIM THIRD TENSOR WRONG, NTYPE=',
     & NTYPE
        STOP
501     J=0
        IF(NTY.LT.0)GO TO 503
        DO 502 K=1,DIMC
502     C(K)=0.0
503     DO 505 J1=1,I1
        DO 505 J2=1,I2
        DO 505 J3=1,I3
        DO 505 J4=1,I4
        DO 505 J5=1,I5
        DO 505 J6=1,I6
        J=J+1
        DO 505 J7=1,I7
        JA=J7+I7*(J5-1+I5*(J3-1+I3*(J1-1)))
        JB=J6+I6*(J4-1+I4*(J2-1+I2*(J7-1)))
505     C(J)=C(J)+SCAL*A(JA)*B(JB)
        RETURN
C***
C***    NTYPE=10 IS THE SPECIAL TYPE OF CONTRACTION, USED OFTEN WHEN THE
C***    CONTRACTION IS OVER A COMBINATION OF INDEPENDENT AND CONTROL
C***    VARIABLES.  FIRST, THE CONDITIONS FOR USING NTYPE=10 MUST BE
C***    SATISFIED.  AS NOTED AT THE BEGINNING OF THIS PROGRAM, THESE
C***    INCLUDE:  AP=0, AQ=B1, BP=0, A1=DIMA/DIMU**AQ, BQ**DIMX=DIMB/DIMU**B1
C***    THE DIMENSION OF C IS THEN CHECKED AGAINST THE OTHER INFORMATION
C***    GIVEN TO MAKE SURE IT AGREES.  IF ANY OF THESE CONDITIONS ARE
C***    VIOLATED, THEN THE PROGRAM IS TERMINATED.  THERE ARE SOME THINGS
C***    WHICH THIS PROGRAM ASSUMES WHEN IT CALCULATES C FOR NTYPE=10.
C***    THE BIGGEST ASSUMPTION IS THAT THE METHOD OF INCREMENTING THE
C***    BASIS ELEMENTS IN C WILL BE TO INCREMENT THOSE THAT COME FROM
C***    B FIRST, THEN INCREMENT THOSE THAT COME FROM A.  THIS DOES
C***    NOT POSE A PROBLEM IN GENERAL, BECAUSE USUALLY ALL OF THE CONTRA-
C***    VARIANT POWERS OF B OR ALL OF THE COVARIANT POWERS OF A ARE INVOLVED
C***    IN THE CONTRACTION.  IF BOTH OF THESE ARE NOT TRUE, THEN THE
C***    TENSOR WHICH IS CALCULATED HERE WILL HAVE TO UNDERGO SOME PERMUTATION
C***    TO OBTAIN THE CORRECT TENSOR WHICH WILL HAVE ITS BASIS ELEMENTS
C***    IN THE STANDARD ORDER.
C***
1000    IF(AP.NE.0)GOTO 1100
        IF(AQ.NE.B1)GOTO 1100
        IF(BP.NE.0)GOTO 1100
        M1=IFIX(FLOAT(DIMA)/FLOAT(DIMU**AQ)+.001)
        IF(A1.NE.M1)GO TO 1200
        IF(NTY.LT.0)GOTO 1003
```

```
1002    DO 1001 J=1,DIMC
1001    C(J)=0.0
C***
C***    J WILL NOW KEEP TRACK OF WHICH ELEMENT OF C IS BEING COMPUTED AT
C***    EACH STEP THROUGH THE DO LOOPS.  THE FIRST TWO DO LOOPS ARE SET UP
C***    SO THAT THEY INCREMENT IN THE SAME WAY AS C, THUS J IS SET TO ZERO
C***    AND IS INCREMENTED EACH TIME THROUGH THE SECOND DO LOOP.
C***
1003    J=0
        DO 1010 J1=1,M1
        DO 1010 J2=1,DIMX**B0
        J=J+1
        DO 1010 J3=1,DIMU**B1
        JA=(J1-1)*DIMU**A0+J3
        JB=(J3-1)*DIMX**B0+J2
1010    C(J)=C(J)+SCAL*A(JA)*B(JB)
        RETURN
1100    TYPE *,' THE DIMENSIONS IN TCONT ARE NOT CORRECT FOR NTYPE=10.'
        STOP
1200    TYPE *,' ERROR SUB TCONT. A1 IS WRONG, NTYPE=10.'
        STOP
        END
```

```
C***********************************************************************
C***********************************************************************
C***********************************************************************
C***                                                                 ***
C***           TITLE:  TADD                                          ***
C***                                                                 ***
C***           SUBROUTINE TADD ADDS TENSOR A TO TENSOR B AND GETS C. ***
C***    C CAN BE EITHER TENSOR A OR B OR A THIRD TENSOR.             ***
C***                                                                 ***
C***    VARIABLES:                                                   ***
C***           A AND B:  INPUT TENSORS                               ***
C***           C:  OUTPUT TENSOR                                     ***
C***           DIMA:  THE DIMENSION OF A, B, AND C                   ***
C***                                                                 ***
C***********************************************************************
C***********************************************************************
C***********************************************************************
      SUBROUTINE TADD(A,B,C,DIMA)
      INTEGER DIMA
      DIMENSION A(DIMA),B(DIMA),C(DIMA)
      DO 1 I=1,DIMA
1     C(I)=A(I)+B(I)
      RETURN
      END
```

```
C*********************************************************************
C*********************************************************************
C*********************************************************************
C***                                                               ***
C***           TITLE:  RUNGE                                       ***
C***           ADAPTED BY:  JOSEPH A. O'SULLIVAN                   ***
C***                                                               ***
C***           SUBROUTINE RUNGE IS A FOURTH ORDER RUNGE-KUTTA INTE-  ***
C***      GRATION ROUTINE.  IT MUST BE CALLED FOUR TIMES.  BEFORE EACH ***
C***      CALL, THE DERIVATIVE MUST BE CALCULATED.                 ***
C***                                                               ***
C***      CALLING VARIABLES:                                       ***
C***           N:  DIMENSION OF Y AND F                            ***
C***           Y:  VECTOR WHICH IS BEING INTEGRATED                ***
C***           F:  DERIVATIVE OF Y WITH RESPECT TO X               ***
C***           X:  INDEPENDENT VARIABLE                            ***
C***           H:  INTEGRATION STEPSIZE FOR X                      ***
C***           M:  INTEGER BETWEEN ONE AND FOUR FOR WHICH CALL IT IS ***
C***                                                               ***
C***      IMPORTANT VARIABLES:                                     ***
C***           ZZZZ:  THE LARGEST SIZE VECTOR WHICH CAN BE INTEGRATED ***
C***           PHI:  KUTTA'S COEFFICIENTS                          ***
C***           SAVEY:  THE INITIAL Y IS STORED IN THIS VECTOR      ***
C***                                                               ***
C***      THE RUNGE-KUTTA EQUATION IS:                             ***
C***           Y(X+H)=Y(X)+(F1+2*F2+2*F3+F4)*H/6                   ***
C***           WHERE:                                              ***
C***                F1=F(X,Y(X))                                   ***
C***                F2=F(X+H/2,Y(X)+H*F1/2)                        ***
C***                F3=F(X+H/2,Y(X)+H*F2/2)                        ***
C***                F4=F(X+H,Y(X)+H*F3)                            ***
C***                                                               ***
C*********************************************************************
C*********************************************************************
C*********************************************************************
      SUBROUTINE RUNGE(N,Y,F,X,H,M)
      PARAMETER ZZZZ=125
      DIMENSION PHI(ZZZZ),SAVEY(ZZZZ),Y(N),F(N)
      IF(M.GT.4)GOTO 6
      IF(M.LT.1)GOTO 6
      GOTO (2,3,4,5),M
C***
C***  ON THE FIRST PASS, F=F1.  Y IS SAVED, F1 IS SAVED, X IS
C***  INCREASED BY ONE-HALF OF H, AND Y IS INCREASED BY H*F/2.  THIS
C***  IS NECESSARY TO CALCULATE F2.
C***
2     DO 22 J=1,N
      SAVEY(J)=Y(J)
      PHI(J)=F(J)
22    Y(J)=SAVEY(J)+0.5*H*F(J)
      X=X+0.5*H
      RETURN
C***
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
C***     THE SECOND PASS STARTS HERE.  F=F2, AND Y IS CHANGED TO CALCU-
C***     LATE F3.
C***
3        DO 33 J=1,N
         PHI(J)=PHI(J)+2.0*F(J)
33       Y(J)=SAVEY(J)+0.5*H*F(J)
         RETURN
C***
C***     THE THIRD PASS STARTS HERE.  F=F3, Y AND X ARE BOTH CHANGED TO
C***     CALCULATE F4.
C***
4        DO 44 J=1,N
         PHI(J)=PHI(J)+2.0*F(J)
44       Y(J)=SAVEY(J)+H*F(J)
         X=X+0.5*H
         RETURN
C***
C***     THE FOURTH AND FINAL PASS STARTS HERE.  F=F4, AND THE NEXT
C***     VALUE OF Y IS CALCULATED FROM THE STORED VALUES OF F AND
C***     FROM THE INTEGRATION STEPSIZE, H.
C***
5        DO 55 J=1,N
55       Y(J)=SAVEY(J)+(PHI(J)+F(J))*H/6.0
         RETURN
C***
C***     IF M IS NOT IN THE RANGE ONE TO FOUR, THEN THE FOLLOWING ERROR
C***     IS TYPED.
C***
6        TYPE 66
66       FORMAT(' ERROR IN RUNGE. M IS NOT BETWEEN 1 AND 5')
         STOP
         END
```

```
C***********************************************************************
C***********************************************************************
C***********************************************************************
C***                                                                 ***
C***            TITLE:  SMULT                                        ***
C***                                                                 ***
C***            THIS SUBROUTINE PERFORMS SCALAR MULTIPLICATION ON    ***
C***    TENSORS.  THE RESULT IS RETURNED TO THE CALLING PROGRAM IN   ***
C***    THE ORIGINAL TENSOR.                                         ***
C***                                                                 ***
C***    VARIABLES:                                                   ***
C***            A:  THE TENSOR OF DIMENSION DIMA                     ***
C***            S:  THE SCALAR BY WHICH A IS MULTIPLIED              ***
C***                                                                 ***
C***********************************************************************
C***********************************************************************
C***********************************************************************
        SUBROUTINE SMULT(A,DIMA,S)
        INTEGER DIMA
        DIMENSION A(DIMA)
        DO 1 I=1,DIMA
1       A(I)=A(I)*S
        RETURN
        END
```

```
C*****************************************************************************
C*****************************************************************************
C*****************************************************************************
C***                                                                      ***
C***            TITLE:  TRANS                                             ***
C***            AUTHOR:  JOSEPH A. O'SULLIVAN                             ***
C***                                                                      ***
C***            THE SUBROUTINE TRANS PERFOMS TRANSPOSITION OF TENSORS.    ***
C***    TRANSPOSITION OF TENSORS INVOLVES THE SIMULTANEOUS RAISING OF     ***
C***    A COVARIANT POWER AND LOWERING OF A CONTRAVARIANT POWER.  THE     ***
C***    RESULT OF TWO CONSECUTIVE TRANSPOSITIONS IS THE ORIGINAL TEN-     ***
C***    SOR.  SINCE THIS PROGRAM IS FOR COMPUTATIONAL PURPOSES, IT IS     ***
C***    ASSUMED THAT PRIOR TO THE CALLING OF THIS PROGRAM ALL OF THE      ***
C***    CONTRAVARIANT POWERS ARE OF THE SAME VARIABLE.  THUS, UNLESS      ***
C***    THE ORIGINAL TENSOR HAS NO MORE THAN ONE COVARIANT POWER, THIS    ***
C***    SUBROUTINE CANNOT BE USED TO VERIFY THE FACT THAT TWO TRANSPO-    ***
C***    SITIONS RETURN THE ORIGINAL TENSOR.  THE MAPPING FROM B TO BT     ***
C***    IN GENERAL IS:                                                    ***
C***                    B1               1,B1-1                           ***
C***                   B     -------> BT                                  ***
C***                    BP,BQ            1,BP-1,BQ                        ***
C***                                                                      ***
C***    INPUT VARIABLES:                                                  ***
C***            B:  THE ORIGINAL TENSOR                                   ***
C***            BT:  THE TENSOR OBTAINED FROM THE TRANSPOSITION           ***
C***            B1:  THE CONTRAVARIANT POWER OF B                         ***
C***            BP:  THE COVARIANT POWER OF THE CONTROL VARIABLE OF B     ***
C***            BQ:  THE COVARIANT POWER OF THE INDEPENDENT VARIABLE      ***
C***                 OF B                                                 ***
C***            DIMB:  THE DIMENSION OF B AND BT                          ***
C***            DIMU:  THE DIMENSION OF THE CONTROL VARIABLE              ***
C***            DIMX:  THE DIMENSION OF THE INDEPENDENT VARIABLE          ***
C***            NTYPE:  THE TYPE OF TRANSPOSITION PERFORMED               ***
C***                                                                      ***
C***    TYPES OF TRANSPOSITION:                                           ***
C***            NTYPE=1:  THE TENSOR B HAS A CONTRAVARIANT POWER OF       ***
C***                 THE INDEPENDENT VARIABLE                             ***
C***            NTYPE=2:  THE TENSOR B HAS A CONTRAVARIANT POWER OF       ***
C***                 THE CONTROL VARIABLE.  BP MUST BE ZERO IN THIS       ***
C***                 CASE, AND THE RESULTING TENSOR, BT, WILL HAVE        ***
C***                 ONE COVARIANT POWER OF THE CONTROL VARIABLE          ***
C***                 AND BQ-1 COVARIANT POWER OF THE INDEPENDENT          ***
C***                 VARIABLE.                                            ***
C***                                                                      ***
C***    IMORTANT VARIABLES:                                               ***
C***            N:  THE DIMENSION OF B AS CALCULATED FROM THE OTHER       ***
C***                 INFORMATION (DIMENSIONS OF THE CONTROL AND           ***
C***                 INDEPENDENT VARIABLES, NTYPE, AND B1, BP, BQ)        ***
C***            J1:  THE ELEMENT FROM B WHICH IS BEING MAPPED TO AN       ***
C***                 ELEMENT FROM BT AT EACH STEP                         ***
C***            J:  THE ELEMENT FROM BT TO WHICH THE ELEMENT FROM B IS    ***
C***                 BEING MAPPED AT EACH STEP                            ***
C***            N1:  THE DIMENSION OF THE PART OF THE TENSOR WHICH IS     ***
```

```
C***                    CONTRAVARIANT IN BOTH B AND BT                    ***
C***          N2:  THE DIMENSION OF THE PART OF THE TENSOR WHICH IS       ***
C***                    COVARIANT IN BOTH B AND BT                        ***
C***                                                                      ***
C************************************************************************
C************************************************************************
C************************************************************************
      SUBROUTINE TRANS(B,BT,B1,BP,BQ,DIMB,DIMU,DIMX,NTYPE)
      INTEGER BQ,B1,BP,DIMB , DIMU, DIMX
      DIMENSION B(DIMB), BT(DIMB)
C***
C***  FIRST DIMB IS CHECKED AGAINST THE OTHER DIMENSIONS TO MAKE SURE
C***  THAT THEY AGREE.  IF THEY DISAGREE THE PROGRAM IS TERMINATED.
C***
      N=DIMX**BQ*DIMU**BP
      IF(NTYPE.EQ.2) N=N*DIMU**B1
      IF(NTYPE.EQ.1) N=N*DIMX**B1
      IF(N.EQ.DIMB)GO TO 1
      TYPE 101,NTYPE
      STOP
1     GO TO (2,50), NTYPE
C***
C***  IF NTYPE=1, BP AND B1 MUST NOT BE ZERO.  IF EITHER WERE ZERO, THEN
C***  THERE WOULD BE NO COVARIANT POWER OF THE CONTROL VARIABLE TO RAISE.
C***  IF BP=0 OR B1=0, THE PROGRAM IS TERMINATED.
C***
2     IF(BP.NE.0) GO TO 10
      TYPE 104,NTYPE
      STOP
10    IF(B1.NE.0) GO TO 20
      TYPE 111,NTYPE
      STOP
C***
C***  NEXT J IS INITIALIZED, AND N1 AND N2 ARE CALCULATED.  NOTE THAT
C***  S  CE ONE CONTRAVARIANT POWER IS LOWERED, THE DIMENSION OF THE
C***  P  T OF THE TENSOR WHICH IS CONTRAVRIANT FOR BOTH B AND BT IS
C***  DIMX**(B1-1).  SIMILARLY, THE DIMENSION OF THE PART OF THE TENSOR
C***  WHICH IS COVARIANT FOR BOTH TENSORS IS THE DIMENSION OF THE COVA-
C***  RIANT PART DIVIDED BY DIMU.
C***
20    J=0
      N1=DIMX**(B1-1)
      N2=DIMX**BQ*DIMU**(BP-1)
C***
C***  THE NEXT PART DOES THE ACTUAL ASSIGNMENT OF VALUES TO BT.  THE
C***  ORDERING OF BASIS ELEMENTS IS ALWAYS LEXICOGRAPHIC.  N2 CORRES-
C***  PONDS TO THE NUMBER OF CONSECUTIVE TIMES THE LEXICOGRAPHIC INCRE-
C***  MENTATION IS THE SAME FOR BT AS FOR B.  N1 CORRESPONDS TO THE
C***  NUMBER OF TIMES THAT INCREMENTATION IS THE SAME IN A DIFFERENT
C***  PART OF THE TENSOR.  I1 AND I3 WILL BE INCREMENTED FROM ONE TO
C***  DIMU AND DIMX RESPECTIVELY.  THEY EACH CORRESPOND TO THE INCREMEN-
C***  TATION OF ONE BASIS ELEMENT.  SPECIFICALLY, THEY ARE THE BASIS
C***  ELEMENTS WHICH CHANGE PLACES IN THE MAPPING.  THUS, THERE ARE
C***  IN EFFECT FOUR PARTS WHICH MUST BE ORDERED LEXICOGRAPHICALLY.
C***  IN BT, THE FIRST PART TO BE INCREMENTED IS OF DIMENSION N2, THE
```

```
C***    SECOND OF DIMENSION DIMX, THE THIRD N1, THE FOURTH DIMU.  IN B,
C***    THE ORDER IS N2, DIMU, N1, DIMX.  SINCE THE FOUR LOOPS ARE
C***    INCREMENTED EXACTLY AS THE FOUR PARTS OF BT ARE, INCREMENTING
C***    J BY ONE AT EACH STEP WILL KEEP TRACK OF THE ELEMENT OF BT
C***    THAT IS BEING MAPPED TO.  THE ELEMENT OF B THAT MAPS TO THAT
C***    ELEMENT OF BT IS DETERMINED BY REORDERING THE BASIS ELEMENTS IN BT
C***    BACK TO THE ORDER OF THE ELEMENTS IN B.  EFFECTIVELY THIS IS
C***    DONE WITH J1 BY CALCULATING THE POSITION BEFORE THE LAST N2 ARE
C***    INCREMENTED AND THEN INCREMENTING THE LAST N2 AT THE SAME TIME
C***    AS THE LAST N2 ELEMENTS OF BT ARE INCREMENTED.
C***
        DO 40 I1=1,DIMU
        DO 40 I2=1,N1
        DO 40 I3=1,DIMX
        J1=(I1-1)*N2+(I2-1)*N2*DIMU+(I3-1)*N1*N2*DIMU
        DO 40 I4=1,N2
        J1=J1+1
        J=J+1
40      BT(J)=B(J1)
        RETURN
C***
C***    IF NTYPE=2, BP MUST BE ZERO, AND B1 AND BQ MUST BE NONZERO.
C***    IF ANY OF THESE ARE NOT SATISFIED, THE PROGRAM IS STOPPED.
C***
50      IF(BP.EQ.0)GOTO 60
        TYPE 155
        STOP
60      IF(B1.NE.0.AND.BQ.NE.0)GOTO 70
        TYPE 165
        STOP
C***
C***    THE SAME THINGS SAID ABOUT J, N1, N2, AND J1 FOR NTYPE=1 APPLY
C***    TO THE FOLLOWING AS WELL.
C***
70      N2=DIMX**(BQ-1)
        N1=DIMU**(B1-1)
        J=0
        DO 80 I1=1,DIMX
        DO 80 I2=1,N1
        DO 80 I3=1,DIMU
        J1=(I1-1)*N2+(I2-1)*N2*DIMX+(I3-1)*N2*N1*DIMX
        DO 80 I4=1,N2
        J=J+1
        J1=J1+1
80      BT(J)=B(J1)
        RETURN
C***
C***    THE FOLLOWING ARE THE FORMAT STATEMENTS FOR THE ERRUR CONDITIONS
C***    FOR THE SUBROUTINE TRANS.
C***
101     FORMAT(' SUBROUTINE TRANS. DIM OF TENSOR DOES NOT AGREE WITH
     1 OTHER DIMS. NTYPE=',I4)
104     FORMAT(' TRANS WAS CALLED FOR A TENSOR WITH NO U POWERS.NTYPE=',I4)
111     FORMAT(' TRANS WAS CALLED,NTYPE=',I4,'WITH NO X POWER TO LOWER')
155     FORMAT(' BP SHOULD BE ZERO FOR SUBROUTINE TRANS NTYPE=2')
```

```
165   FORMAT(' TRANS WAS CALLED, NTYPE=2, WITH NO U POWER TO LOWER')
      END
```

```
C***********************************************************************
C***********************************************************************
C***********************************************************************
C***                                                                ***
C***          TITLE:  RAISE                                         ***
C***          AUTHOR:  JOSEPH A. O'SULLIVAN                         ***
C***                                                                ***
C***          THIS SUBROUTINE RAISES AND LOWERS POWERS OF THE CONTROL ***
C***     AND INDEPENDENT VARIABLES.  'RAISING A POWER' MEANS CHANGING  ***
C***     THAT POWER FROM COVARIANT TO CONTRAVARIANT.  SIMILARLY, LOWER- ***
C***     ING POWERS CORRESPONDS TO CHANGING THE POWERS FROM CONTRAVA-   ***
C***     RIANT TO COVARIANT.  THE TENSOR RESULTING FROM THE RAISE       ***
C***     SUBROUTINE, B, WILL BE OF THE SAME DIMENSION AS THE ENTERING   ***
C***     TENSOR, A, AND WILL IN FACT HAVE THE SAME ELEMENTS AS A, ONLY  ***
C***     IN A DIFFERENT ORDER.  THUS, SUBROUTINE RAISE MERELY REORDERS  ***
C***     THE ELEMENTS IN THE ORIGINAL TENSOR.  IT MUST BE NOTED, HOW-   ***
C***     EVER, THAT THE NEW TENSOR IS NOT THE SAME TYPE AS THE ORIGINAL. ***
C***     THE NEW TENSOR WILL HAVE ELEMENTS WHOSE BASIS ELEMENTS ARE     ***
C***     TOTALLY DIFFERENT FROM A'S, AND THUS IT CANNOT BE USED IN THE  ***
C***     SAME CIRCUMSTANCES (E.G. CONTRACTIONS) AS A.                   ***
C***                                                                ***
C***     CALLING VARIABLES:                                         ***
C***          A:  THE ORIGINAL TENSOR                               ***
C***          B:  THE RESULTANT TENSOR                              ***
C***          A1:  THE CONTRAVARIANT POWER OF A                     ***
C***          AP:  THE COVARIANT POWER OF THE CONTROL VARIABLE OF A ***
C***          AQ:  THE COVARIANT POWER OF THE INDEPENDENT VARIABLE  ***
C***                OF A                                            ***
C***          B1:  THE CONTRAVARIANT POWER OF B                     ***
C***          BP:  THE COVARIANT POWER OF THE CONTROL VARIABLE OF B ***
C***          BQ:  THE COVARIANT POWER OF THE INDEPENDENT VARIABLE  ***
C***                OF B                                            ***
C***          DIMU:  THE DIMENSION OF THE CONTROL VARIABLE          ***
C***          DIMX:  THE DIMENSION OF THE INDEPENDENT VARIABLE      ***
C***          DIMA:  THE DIMENSION OF A AND B                       ***
C***          NTYPE:  INDICATES THE TYPE OF 'RAISE' PERFORMED       ***
C***                                                                ***
C***     TYPES OF 'RAISE':                                          ***
C***          NTYPE=1:  (B1-A1) COVARIANT POWERS OF THE CONTROL     ***
C***                VARIABLE OF A ARE RAISED.                       ***
C***                FOR THIS TYPE, IT IS NECESSARY THAT:            ***
C***                     1) AQ=BQ        2) B1-A1=AP-BP             ***
C***                     3) DIMA=DIMU**(A1+AP)*DIMX**AQ             ***
C***                     4) (B1-A1)>0                               ***
C***          NTYPE=2:  (B1-A1) COVARIANT POWERS OF THE INDEPENDENT ***
C***                VARIABLE OF A ARE RAISED.                       ***
C***                FOR THIS TYPE, IT IS NECESSARY THAT:            ***
C***                     1) AP=BP        2) B1-A1=AQ-BQ             ***
C***                     3) DIMA=DIMU**AP*DIMX**(A1+AQ)             ***
C***                     4) (B1-A1)>0                               ***
C***          NTYPE=3:  (A1-B1) CONTRAVARIANT POWERS OF THE CONTROL ***
C***                VARIABLE ARE LOWERED.                           ***
C***                FOR THIS TYPE, IT IS NECESSARY THAT:            ***
```

```
C***                    1) AQ=BQ       2) A1-B1=BP-AP                 ***
C***                    3) DIMA=DIMU**(A1+AP)*DIMX**AQ                ***
C***                    4) (A1-B1)>0                                  ***
C***         NTYPE=4:  (A1-B1) CONTRAVARIANT POWERS OF THE INDE-      ***
C***                PENDENT VARIABLE OF A ARE LOWERED.                ***
C***                FOR THIS TYPE, IT IS NECESSARY THAT:              ***
C***                    1) AP=BP       2) A1-B1=BQ-AQ                 ***
C***                    3) DIMA=DIMU**AP*DIMX**(A1+AQ)                ***
C***                    4) (A1-B1)>0                                  ***
C***                                                                  ***
C***    MAPPING FOR RAISE:                                            ***
C***                A1                B1                              ***
C***               A       --------> B                                ***
C***                AP,AQ             BP,BQ                           ***
C***                                                                  ***
C***    IMPORTANT VARIABLES:                                          ***
C***          N3:  THE DIMENSION OF A CALCULATED FROM THE OTHER       ***
C***                   INFORMATION (DIMU, DIMX, A1, AP, AQ)           ***
C***          IA:  THE ELEMENT OF A WHICH IS BEING MAPPED AT EACH     ***
C***                   STEP                                           ***
C***          IB:  THE ELEMENT IN B TO WHICH THE ELEMENT OF A IS      ***
C***                   BEING MAPPED AT EACH STEP                      ***
C***                                                                  ***
C*********************************************************************
C*********************************************************************
C*********************************************************************
      SUBROUTINE RAISE(A,B,A1,AP,AQ,B1,BP,BQ,DIMU,DIMX,DIMA,NTYPE)
      INTEGER A1,AP,AQ,B1,BP,BQ,DIMU,DIMX
      INTEGER DIMA
      DIMENSION A(DIMA),B(DIMA)
      GO TO (1,2,1,2),NTYPE
1     IF(AQ.EQ.BQ)GOTO 11
      TYPE 10,NTYPE
      STOP
C***
C***  FOR NTYPE=1, AND NTYPE=3 (AND NTYPE=2 OR 4 IF AP=0) THERE IS A
C***  SLIGHTLY DEGENERATE SITUATION.  ALTHOUGH THE NEW TENSOR WILL BE
C***  DIFFERENT THEORETICALLY FROM THE OLD TENSOR, THE ORDERING OF
C***  THE BASIS ELEMENTS CORRESPONDING TO THE NEW TENSOR WILL BE EXACTLY
C***  THE SAME AS THAT FOR THE OLD TENSOR.  THUS THE TWO TENSORS WILL
C***  HAVE EXACTLY THE SAME ELEMENTS IN EXACTLY THE SAME ORDER.
C***
11    DO 19 I1=1,DIMA
19    B(I1)=A(I1)
      RETURN
2     IF(AP.EQ.0)GOTO 11
      N3=DIMX**(A1+AQ)*DIMU**AP
      IF(N3.EQ.DIMA)GOTO 12
      TYPE 20,NTYPE
      STOP
12    IF(NTYPE.EQ.4)GOTO 4
C***
C***  THE FOLLOWING IS FOR NTYPE=2.  THERE ARE FOUR DO LOOPS.  THEY ARE
C***  ORDERED SO THAT THEY INCREMENT IN THE SAME WAY THAT TENSOR A DOES.
C***  THUS, IA IS INCREMENTED ONCE FOR EACH TIME THROUGH THE INNERMOST
```

```
C***    LOOP.  THE DIMENSION OF THE FIRST PART OF B THAT IS INCREMENTED IS
C***    I6; THE SECOND PART HAS DIMENSION I4; THE THIRD PART I5; THE FOURTH
C***    PART I7.  THE POSITION PRIOR TO INCREMENTING THE FIRST I6 IS CALCU-
C***    LATED AS IB AND THEN INCREMENTED FOR EACH OF THE INNERMOST I6 STEPS.
C***
        IA=0
        I7=DIMX**A1
        I4=DIMU**AP
        I5=DIMX**(AQ-BQ)
        I6=DIMX**BQ
        DO 29 I0=1,I7
        DO 29 I1=1,I4
        DO 29 I2=1,I5
        IB=(I1-1)*I6+I6*I4*(I2-1+I5*(I0-1))
        DO 29 I3=1,I6
        IA=IA+1
        IB=IB+1
29      B(IB)=A(IA)
        RETURN
C***
C***    FOR NTYPE=4, CONTRAVARIANT POWERS OF THE INDEPENDENT VARIABLE ARE
C***    LOWERED.  HERE THE DO LOOPS ARE SET UP TO INCREMENT IN THE SAME
C***    MANNER AS THE ELEMENTS OF B ARE ORDERED LEXICOGRAPHICALLY.  IA
C***    IS THEN CALCULATED JUST AS IB WAS CALCULATED FOR NTYPE=2.
C***
1       IB=0
        I7=DIMX**B1
        I4=DIMU**BP
        I5=DIMX**(BQ-AQ)
        I6=DIMX**AQ
        DO 49 I0=1,I7
        DO 49 I1=1,I4
        DO 49 I2=1,I5
        IA=I6*(I1-1+I4*(I2-1+I5*(I0-1)))
        DO 49 I3=1,I6
        IB=IB+1
        IA=IA+1
49      B(IB)=A(IA)
        RETURN
C***
C***    THE FOLLOWING IS THE FORMAT SECTION FOR ERRORS IN 'RAISE'.
C***
10      FORMAT(' THE POWERS OF X DO NOT MATCH. SUB RAISE, NTYPE=',I3)
20      FORMAT(' THE POWERS ARE NOT CORRECT IN SUBROUTINE RAISE,NTYPE=',I3)
        END
```

```
C*****************************************************************************
C*****************************************************************************
C*****************************************************************************
C***                                                                       ***
C***              TITLE:  SYM                                              ***
C***              AUTHOR:  JOSEPH A. O'SULLIVAN                            ***
C***              DATE:  MARCH 1984                                        ***
C***                                                                       ***
C***              SUBROUTINE SYM PERFORMS SYMMETRIZATION OF TENSORS.  IT   ***
C***      OBTAINS TENSORS IN THE SYMMETRIC SPACE FROM TENSORS IN THE NOR-  ***
C***      MAL TENSOR SPACE.  IT ALSO CAN GET LARGER DIMENSION TENSORS      ***
C***      FROM TENSORS IN THE SYMMETRIC TENSOR SPACE.                      ***
C***                                                                       ***
C***      IMPORTANT VARIABLES:                                             ***
C***              TENS1:  LARGE DIMENSION TENSOR WHICH HAS IP1 POWERS OF   ***
C***                      ONE VARIABLE (WHOSE DIMENSION IS IDIMP) AND IQ1  ***
C***                      POWERS OF ANOTHER VARIABLE (WHOSE DIMENSION IS   ***
C***                      IS IDIMQ).                                       ***
C***              TENS2:  SYMMETRIC TENSOR WHICH HAS IPPP POWERS OF ONE    ***
C***                      VARIABLE SYMMETRIZED AND IQQQ POWERS OF THE      ***
C***                      OTHER VARIABLE SYMMETRIZED.                      ***
C***              IJJ:  INDICATES WHETHER OR NOT THERE IS A CONTRAVARIANT  ***
C***                      POWER NOT BEING SYMMETRIZED AND WHAT TYPE IT IS  ***
C***              IT1:  THE DIMENSION OF TENS1                             ***
C***              IT2:  DIMENSION OF TENS2                                 ***
C***              IOPT:  SELECTS THE TYPE OF SYMMETRIZATION PERFORMED      ***
C***                                                                       ***
C*****************************************************************************
C*****************************************************************************
C*****************************************************************************
      SUBROUTINE SYM(TENS1,TENS2,IP1,IQ1,IJJ,IPPP,IQQQ,IT1,IT2,IDIMP,
     1  IDIMQ,IOPT,IERR)
      DIMENSION II(10),IJ(10),IOQ(10),IOP(10),JTEN1(256)
      DIMENSION JTEN2(180),IOQ1(10),IOP1(10),JCONT(256)
      DIMENSION TENS1(IT1),TENS2(IT2),ID(2)
C***
C***  THE FIRST PART OF THE PROGRAM CHECKS SEVERAL OF THE INPUT
C***  ARGUMENTS FOR CONSISTENCY.  THE ARGUMENTS ARE CHECKED AGAINST
C***  THEIR BOUNDS AND AGAINST THE OTHER ARGUMENTS.  IF ANY OF THE
C***  ARGUMENTS DISAGREE WITH EACH OTHER, THEN THE PROGRAM IS TERMINATED.
C***
      INQ=IQQQ
      INP=IPPP
      IF(IT1.LE.256) GOTO 2
      TYPE *,' IN SUB SYM THE DIMENSION OF THE FIRST TENSOR IS TOO LARGE'/
     & ' CALL #',IERR
      STOP
2     IF(IJJ.GT.2.OR.IJJ.LT.0)GOTO 1100
      IF(IJJ.EQ.0)GOTO 10
      ID(1)=IDIMP
      ID(2)=IDIMQ
      IDJ=ID(IJJ)
      GOTO 11
10    IDJ=1
```

```
11      IF(IPPP.EQ.1) INP=0
        IF(IQQQ.EQ.1) INQ=0
        IF(IDIMP.EQ.1) INP=0
        IF(IDIMQ.EQ.1) INQ=0
        IF(IT1.EQ.IT2) GOTO 1201
        IF(IP1.LT.0.OR.IQ1.LT.0.OR.IPPP.LT.0.OR.IQQQ.LT.0) GO TO 1100
        IF(IT1.LE.0.OR.IT2.LE.0.OR.IDIMP.LE.0.OR.IDIMQ.LE.0) GOTO 1100
        IF(IPPP.GT.IP1)GOTO 1000
        IF(IQQQ.GT.IQ1)GOTO 1000
        I1=(IDIMP**IP1)*(IDIMQ**IQ1)*IDJ
        IF(I1.NE.IT1)GOTO 1000
        J=0
        ITOTP=1
        ITOTQ=1
        DO 9 I=1,IT2
9       JTEN2(I)=0
        ICOMQ=IDIMQ+INQ-1
        IC=ICOMQ
        IF(ICOMQ.EQ.0)IC=1
        ICOMQ=IC
        IF(ICOMQ.GT.0)GO TO 304
        TYPE *,' ERROR IN SUB SYM. LINE 304. ICOMQ SHOULD NOT BE LE 0, CALL#',
     & IERR
        STOP
304     DO 14 I=1,ICOMQ
14      ITOTQ=ITOTQ*I
        IF(INQ.EQ.0)GOTO 306
        DO 15 I=1,INQ
15      ITOTQ=ITOTQ/I
306     I15=IDIMQ-1
        IF(I15.GT.0)GOTO 316
        ITOTQ=1
        IF(I15.EQ.0)GOTO 307
        TYPE *,' ERROR IN SUB SYM. LINE 306. I15 SHOULD NOT BE LT 0. CALL #
     & ',IERR
        STOP
316     DO 16 I=1,I15
16      ITOTQ=ITOTQ/I
307     ICOMP=IDIMP+INP-1
        IC=ICOMP
        IF(ICOMP.EQ.0)IC=1
        ICOMP=IC
        IF(ICOMP.GT.0)GOTO 317
        TYPE *, ' ERROR IN SUB SYM. LINE 317. ICOMP SHOULD NOT BE LT 0. CALL #
     & ', IERR
        STOP
317     DO 17 I=1,ICOMP
17      ITOTP=ITOTP*I
        IF(INP.EQ.0)GOTO 309
        DO 18 I=1,INP
18      ITOTP=ITOTP/I
309     I15=IDIMP-1
        IF(I15.GT.0)GOTO 319
        ITOTP=1
        IF(I15.EQ.0)GOTO 320
```

```
      TYPE *,' ERROR IN SUB SYM. LINE 309. I15 IS TOO SMALL. CALL #',IERR
      STOP
319   DO 19 I=1,I15
19    ITOTP=ITOTP/I
320   ITT2=ITOTP*ITOTQ*IDJ*IDIMQ**(IQ1-INQ)*IDIMP**(IP1-INP)
20    IF(ITT2.EQ.IT2)GOTO 24
      TYPE 21, IERR
21    FORMAT(' THE DIMENSION OF THE SMALL TENSOR DOES NOT MATCH'/
     1 ' THE CALCULATED DIMENSION IN SUBROUTINE SYM, CALL ',I2)
      STOP
C***
C***  THE FOLLOWING SECTION  OF THE PROGRAM IS DEVOTED TO SETTING UP
C***  A MAPPING FROM THE ELEMENTS OF THE LARGER DIMENSION TENSOR
C***  TO THE SMALLER DIMENSION SYMMETRIC TENSOR.  THIS MAPPING IS
C***  CARRIED OUT BY ASSIGNING A NUMBER TO EACH ELEMENT OF THE SYMMET-
C***  RIC BASIS AND THEN IN THE MAIN PROGRAM CORRESPONDING EACH
C***  OF THESE NUMBERS TO AN ENTRY IN THE SYMMETRIC TENSOR.  THE
C***  NUMBER IS COMPUTED BY FIRST PUTTING THE BASIS ELEMENTS IN
C***  LEXICOGRAPHIC ORDER.  THE BASIS ELEMENTS ARE THEN MAPPED TO
C***  SYMMETRIC BASIS ELEMENTS.  THESE BASIS ELEMENTS ARE THEN
C***  ASSUMED TO BE DIGITS IN A BASE IDIMQ NUMBER SYSTEM.  THEY
C***  ARE THEN CONVERTED TO BASE TEN AND THIS NUMBER IS THE ONE
C***  WHICH CORRESPONDS TO THE BASIS ELEMENT.  THESE BASE TEN NUMBERS
C***  ARE THEN PUT IN ASCENDING ORDER IN A TENSOR, JTEN2(I).
C***  IN THE SECTION WHICH FOLLOWS THIS MAPPING SECTION, THE CON-
C***  VERSION FROM TENSOR TO TENSOR TAKES PLACE.  THIS IS ACCOMPLISHED
C***  BY AGAIN CMPUTING THE BASE TEN NUMBER CORRESPONDING TO EACH
C***  ELEMENT AND THEN FINDING OUT WHICH ELEMENT IN JTEN2(I) IT
C***  CORRESPONDS TO.  THE CONVERSION IS THEN EASY.  THE REASON
C***  THIS SEEMINGLY ROUNDABOUT METHOD IS USED IS THAT NOT ALL
C***  OF THE COMPONENTS OF THE BASIS MAY BE DESIRED IN SYMMETRIC
C***  FORM.  THUS IF INQ<<IQ1 THE MAPPING SECTION OF THE PROGRAM
C***  WILL NOT BE RUNNING FOR A LONG TIME COMPARED TO THE FOLLOWING
C***  SECTION OF THE PROGRAM.  IF, HOWEVER, INQ=IQ1 AND INP=IP1,
C***  THEN THE MAPPING SECTION WILL TAKE AS LONG AS THE SECTION
C***  WHICH FOLLOWS THIS ONE.
C***  NOTE THAT IN THIS SECTION IOQ(I) AND IOQ1(I) ARE KEPT SEPA-
C***  RATE. THIS IS SO THAT THE NORMAL BASIS COMPONENTS, IOQ(I),
C***  ARE INCREASED LEXICOGRAPHICALLY APART FROM THE SYMMETRIC
C***  BASIS COMPONENTS, IOQ1(I), WHICH ARE REORDERED FOR EACH I.
C***  ALSO NOTE THAT IOQ(1) IS THE 'LAST' COMPONENT; THE COMPON-
C***  ENT WHICH IS INCREMENTED FOR EACH I.
C***
24    IF(INQ.EQ.0)GOTO 380
      DO 80 I=1,INQ
80    IOQ(I)=1
380   IF(INP.EQ.0)GOTO 381
      DO 81 I=1,INP
81    IOP(I)=1
381   IOQ(1)=0
      IF(INQ.EQ.0)GOTO 200
      N1=IDIMQ**INQ
      DO 92 I=1,N1
      IOQ(1)=IOQ(1)+1
      N3=INQ-1
```

```
      DO 82 N2=1,N3
      IF(IOQ(N2).LE.IDIMQ) G   .O 82
      N4=N2+1
      IOQ(N4)=IOQ(N4)+1
      IOQ(N2)=1
82    CONTINUE
      DO 95 N2=1,INQ
95    IOQ1(N2)=IOQ(N2)
      N2=1
      N3=2
83    CONTINUE
      IF (IOQ1(N2).GE.IOQ1(N3))GO TO 84
      N4=IOQ1(N3)
      IOQ1(N3)=IOQ1(N2)
      IOQ1(N2)=N4
84    N3=N3+1
      IF(N3.LE.INQ)GO TO 83
      N2=N2+1
      N3=N2+1
      IF(N3.LE.INQ)GO TO 83
      JTEN1(I)=IOQ1(INQ)-1
      DO 85 N2=2,INQ
      N3=INQ-N2+1
85    JTEN1(I)=JTEN1(I)*IDIMQ+IOQ1(N3)-1
92    JTEN1(I)=JTEN1(I)+1
      GOTO 220
200   JTEN1(I)=0
201   N1=0
      IF(INP.EQ.0)GO TO 1200
      GO TO 230
220   IF(INP.NE.0) GOTO 230
      N10=N1
      GOTO 221
C***
C***  THE FOLLOWING SECTION PERFORMS THE SAME OPERATIONS ON THE U
C***  POWERS THAT WERE PERFORMED ABOVE ON THE X POWERS.  THE MAPPING
C***  IN THIS SECTION WILL CONTINUE TO BE TO THE TENSOR JTEN1(I),
C***  CONTINUING WHERE THE LAST SECTION LEFT OFF IN THE TENSOR.
C***  THUS TO UTILIZE THIS MAPPING FROM TENSOR TO SYMMETRIC TENSOR,
C***  ONE MUST ADD IDIMQ**INQ TO THE CALCULATED ENTRY POINT (THE
C***  BASE TEN NUMBER COMPUTED FROM THE IOP1(I) ASSUMING THE LATTER
C***  TO BE ELEMENTS OF A BASE IDIMP NUMBER SYSTEM).
C***
230   IOP(1)=0
      N10=IDIMP**INP+N1
      N1=N1+1
      DO 90 I=N1,N10
      IOP(1)=IOP(1)+1
      N3=INP-1
      DO 86 N2=1,N3
      IF (IOP(N2).LE.IDIMP) GO TO 86
      N4=N2+1
      IOP(N4)=IOP(N4)+1
      IOP(N2)=1
86    CONTINUE
```

```
      DO 96 N2=1,INP
96    IOP1(N2)=IOP(N2)
      N2=1
      N3=2
87    CONTINUE
      IF(IOP(N2).GE.IOP1(N3))GO TO 88
      N4=IOP1(N3)
      IOP1(N3)=IOP1(N2)
      IOP1(N2)=N4
88    N3=N3+1
      IF (N3.LE.INP)GO TO 87
      N2=N2+1
      N3=N3+1
      IF(N3.LE.INP)GO TO 87
      JTEN1(I)=IOP1(INP)-1
      DO 89 N2=2,INP
      N3=INP-N2+1
89    JTEN1(I)=JTEN1(I)*IDIMP+IOP1(N3)-1
90    JTEN1(I)=JTEN1(I)+1+IDINQ**INQ
C***
C***  THE NEXT SECTION COMPLETES THE MAPPING BY REMOVING THE REDUN-
C***  DANT NUMBERS WHICH COME FROM PERMUTATIONS OF THE BASIS CON-
C***  PONENTS.  THE ELEMENTS CORRESPONDING TO THE POWERS OF U
C***  ARE THOSE AFTER JTEN1(ITOTQ).
C***
221   J=0
      N3=N10-2
      DO 103 N2=1,IT1
103   JCONT(N2)=0
      DO 97 N2=1,N3
      N4=N2+1
      DO 97 N5=N4,N10
      IF(JTEN1(N2).NE.JTEN1(N5))GO TO 97
      IF(JTEN1(N2).EQ.0)GO TO 97
      JTEN1(N5)=0
      JCONT(JTEN1(N2))=JCONT(JTEN1(N2))+1
97    CONTINUE
      DO 98 N2=1,N10
      IF(JTEN1(N2).NE.0)J=J+1
98    CONTINUE
C***
C***  HERE J IS JUST THE NUMBER OF DISTINCT NUMBERS IN JTEN1(I).
C***
240   N6=N10-1
C***
C***  THE FOLLOWING PUTS THE ZERO ENTRIES OF JTEN1(I) AT THE
C***  BEGINNING, THEN LOADS THE NEW TENSOR JTEN2(I) FOR THE
C***  PURPOSES OF THE MAPPING.
C***
      DO 99 N2=1,N6
      N4=N2+1
      DO 99 N3=N4,N10
      IF(JTEN1(N2).LE.JTEN1(N3))GO TO 99
      N5=JTEN1(N3)
      JTEN1(N3)=JTEN1(N2)
```

```
      JTEN1(N2)=N5
99    CONTINUE
      M=1
100   N5=JTEN1(M)
      IF(N5.NE.0)GO TO 101
      M=M+1
      GOTO 100
101   M=M-1
      DO 102 N2=1,J
      M=M+1
102   JTEN2(N2)=JTEN1(M)
C***
C***  THAT ENDS THE MAPPING SECTION.
C***
C***
C***  IOPT = 1 CORRESPONDS TO A LARGE TENSOR TO SYMMETRIC TENSOR
C***  TRANSFORMATION OF COVARIANT TYPE.  IOPT = 3 CORRESPONDS TO THE
C***  SAME TRANSFORMATION OF CONTRAVARIANT TYPE.
C***
      IF(IOPT.NE.1.AND.IOPT.NE.3.AND.IOPT.NE.5)GOTO 25
      DO 23 I=1,IT2
23    TENS2(I)=0
C***
C***  NOTE THAT IN THIS SECTION IJ(IQ1) IS THE "LAST" COMPONENT.
C***  IT IS THE ONE WHICH IS INCREMENTED FOR EACH I.
C***
25    IF(IP1.EQ.0)GOTO 313
      DO 12 I=1,IP1
12    II(I)=1
313   IF(IQ1.EQ.0)GOTO 314
      DO 13 I=1,IQ1
13    IJ(I)=1
314   N0=1
      IQ2=IQ1-1
      IF(IQ1.NE.0)IJ(IQ1)=0
      IP2=IP1-1
      DO 120 I=1,IT1
      IF(IQ1.EQ.0)GO TO 32
      IJ(IQ1)=IJ(IQ1)+1
      IF(IQ1.EQ.1)GO TO 31
      DO 30 N2=1,IQ2
      N3=IQ1+1-N2
      IF(IJ(N3).LE.IDIMQ)GO TO 30
              N4=N3-1
              IJ(N4)=IJ(N4)+1
              IJ(N3)=1
30    CONTINUE
31    IF(IJ(1).LE.IDIMQ) GOTO 40
              IJ(1)=1
              IF(IP1.EQ.0)GOTO 260
32            II(IP1)=II(IP1)+1
              IF(IP1.EQ.1)GOTO 41
              DO 35 N2=1,IP2
                      N3=IP1+1-N2
                      IF(II(N3).LE.IDIMP)GOTO35
```

```
                        N4=N3-1
                        II(N4)=II(N4)+1
                        II(N3)=1
35              CONTINUE
C***
C***    THE NEXT FEW LINES CHECK FOR ERRORS.
C***
40      IF(IP1.EQ.0)GOTO 45
41      IF(II(1).LE.IOINP)GOTO 45
        IF(IDJ.EQ.1)GO TO 1100
        II(1)=1
        IF(NO.GT.IDJ)GO TO 1100
260     NO=NO+1
        IF(NO.GT.IDJ)GO TO 1100
45      CONTINUE
        IF(INQ.EQ.0)GOTO 263
        DO 50 N3=1,INQ
        N2=IQ1-N3+1
50      IOQ(N3)=IJ(N2)
C***
C***    NEXT THE BASIS ELEMENTS ARE REORDERED.  IOQ(1) IS NOW
C***    THE 'LAST' COMPONENT.  SIMILARLY, IOP1(1) WILL BE 'LAST'.
C***
263     IF(INP.EQ.0) GOTO 265
        DO 55 N2=1,INP
        N3=IP1-N2+1
55      IOP(N2)=II(N3)
265     N2=1
        N3=2
60      CONTINUE
        IF(INQ.EQ.0)GOTO 262
        IF(IOQ(N2).GE.IOQ(N3))GO TO 61
        N4=IOQ(N3)
        IOQ(N3)=IOQ(N2)
        IOQ(N2)=N4
61      N3=N3+1
        IF(N3.LE.INQ)GO TO 60
        N2=N2+1
        N3=N2+1
        IF(N3.LE.INQ)GO TO 60
        N2=1
        N3=2
262     IF(INP.NE.0)GOTO 62
        NTEN2=0
        GOTO 268
62      CONTINUE
        IF(IOP(N2).GE.IOP(N3)) GO TO 63
        N4=IOP(N3)
        IOP(N3)=IOP(N2)
        IOP(N2)=N4
63      N3=N3+1
        IF(N3.LE.INP) GO TO 62
        N2=N2+1
        N3=N2+1
        IF(N3.LE.INP) GO TO 62
```

```
      IF(INQ.NE.0)GOTO 267
C***
C***  THE NEXT SECTION FINDS THE BASE TEN NUMBERS ASSOCIATED WITH
C***  THE TWO PARTS OF THE TENSORS.
C***
      NTEN1=1
      NTEN3=1
      NTEN2=IOP(INP)-1
      GOTO 269
267   NTEN2=IOP(INP)-1
268   NTEN1=IOQ(INQ)-1
      DO 64 N2=2,INQ
      N3=INQ-N2+1
64    NTEN1=NTEN1*IDIMQ+IOQ(N3)-1
      NTEN1=NTEN1+1
269   IF(INP.NE.0)GOTO 270
      NTEN4=0
      GOTO 271
270   DO 65 N2=2,INP
      N3=INP-N2+1
65    NTEN2=NTEN2*IDIMP+IOP(N3)-1
      NTEN2=NTEN2+1+IDIMQ**INQ
271   DO 66 N2=1,J
      IF(NTEN1.NE.JTEN2(N2))GO TO 67
      NTEN3=N2
      GO TO 66
67    CONTINUE
      IF(NTEN2.NE.JTEN2(N2))GOTO 66
      NTEN4=N2-ITOTQ
      GO TO 68
66    CONTINUE
      IF(INP.EQ.0)GOTO 68
      TYPE 69,IERR
69    FORMAT(' ERROR, SYM, CALL=',I2,' LINE 66.')
      STOP
68    IF(INP.EQ.0)NTEN2=1
      IF(INP.EQ.0)NTEN4=0
      N3=0
      N4=INQ+1
C***
C***  THE NEXT FEW LINES CALCULATE THE COMPONENT OF TENS2
C***  WHICH WILL BE USED WITH THE ITH COMPONENT OF TENS1 IN THE
C***  MAPPING.
C***
      IF(INQ.EQ.IQ1)GOTO 370
      DO 70 N2=N4,IQ1
70    N3=N3+(IJ(IQ1-N2+1)-1)*IDIMQ**(N2-INQ-1)
370   N5=0
      N4=INP+1
      IF(INP.EQ.IP1)GOTO 371
      DO 71 N2=N4,IP1
71    N5=N5+(II(IP1-N2+1)-1)*IDIMP**(N2-INP-1)
371   N2=NTEN3+ITOTQ*(N3+IDIMQ**(IQ1-INQ)*(NTEN4+ITOTP*(N5+IDIMP**(IP1
     & -INP)*(NO-1))))
      IF(IOPT.NE.1.AND.IOPT.NE.5)GO TO 72
```

```
       TENS2(N2)=TENS2(N2)+TENS1(I)
       GO TO 120
72     IF(IOPT.NE.2)GOTO 73
       TENS1(I)=TENS2(N2)/(FLOAT((JCONT(NTER1)+1)*(JCONT(NTER2)+1)))
       GO TO 120
73     IF(IOPT.NE.3)GO TO 74
       TENS2(N2)=TENS2(N2)+TENS1(I)/(FLOAT((JCONT(NTER1)+1)*(JCONT(NTER2)+
     & 1)))
       GO TO 120
74     IF(IOPT.NE.4)GO TO 1230
       TENS1(I)=TENS2(N2)
120    CONTINUE
       IOPT5=IOPT
       IF(IOPT.EQ.5)IOPT5=2
       IF(IOPT.EQ.5)I=-5
       IOPT=IOPT5
       IF(I.EQ.-5)GOTO 25
       RETURN
1000   TYPE 1001,IERR
1001   FORMAT(' IN SUB SYN THE DIMENSIONS LISTED DO NOT AGREE.', I2)
       STOP
1100   TYPE 1101,IERR
1101   FORMAT (' THERE IS AN ERROR IN SYN CALL NUMBER,',I2)
       STOP
1200   IF(IT1.NE.IT2)GO TO 1220
1201   IF(IOPT.EQ.2)GO TO 1203
       DO 1202 I=1,IT1
1202   TENS2(I)=TENS1(I)
       RETURN
1203   DO 1204 I=1,IT1
1204   TENS1(I)=TENS2(I)
       RETURN
1220   TYPE *,' THE OTHER DIMS INDICATE THAT IT1 SHOULD EQUAL IT2'/
     & ' , BUT THEY ARE NOT EQUAL.  SUB SYN,  LINE 1200,  CALL #',IERR
1221   STOP
1230   TYPE *,' IOPT IS NOT WITHIN THE ACCEPTABLE LIMITS.  CALL #',IERR
       STOP
       END
```

```
C*************************************************************************
C*************************************************************************
C*************************************************************************
C***                                                                   ***
C***           TITLE:  DPROD                                           ***
C***           AUTHOR:  JOSEPH A. O'SULLIVAN                           ***
C***                                                                   ***
C***           SUBROUTINE DPROD CALCULATES THE DIRECT PRODUCT OF TWO   ***
C***     TENSORS.  IN THE PRESENT APPLICATION, THE DIRECT PRODUCT IS   ***
C***     ONLY NECESSARY FOR 'SQUARE' TENSORS.  SQUARE TENSORS HAVE THE ***
C***     COVARIANT AND CONTRAVARIANT DIMENSIONS AND THUS THEIR DIMEN-  ***
C***     SIONS ARE THE SQUARE OF AN INTEGER.  THE RESULT OF A DIRECT   ***
C***     PRODUCT OF TWO SQUARE TENSORS IS AGAIN A SQUARE TENSOR.       ***
C***           THE DIRECT PRODUCT IS COMPUTED BY ADDING THE SUM OF     ***
C***     THE TENSOR PRODUCT OF THE FIRST TENSOR AND AN IDENTITY TENSOR ***
C***     THE SIZE OF THE SECOND TENSOR TO THE TENSOR PRODUCT OF AN     ***
C***     IDENTITY TENSOR THE SIZE OF THE FIRST TENSOR AND THE SECOND   ***
C***     TENSOR.  THE DIRECT PRODUCT OF SQUARE TENSORS IS IDENTICAL TO ***
C***     THE DIRECT PRODUCT OF SQUARE MATRICES, WHERE AN IDENTITY      ***
C***     MATRIX IS USED INSTEAD OF AN IDENTITY TENSOR.                 ***
C***                                                                   ***
C***     CALLING VARIABLES:                                            ***
C***           A:  THE FIRST INPUT TENSOR                              ***
C***           B:  THE SECOND INPUT TENSOR                             ***
C***           C:  THE OUTPUT TENSOR                                   ***
C***           IDIMA:  THE DIMENSION OF A                              ***
C***           IDIMB:  THE DIMENSION OF B                              ***
C***           IDIMC:  THE DIMENSION OF C                              ***
C***           NCALL:  A DEBUGGUNG PARAMETER:  IF THIS SUBROUTINE IS   ***
C***                   CALLED SEVERAL TIMES AND THERE IS AN ERROR IN   ***
C***                   ONE OF THE CALLS, THEN THE CALL IN WHICH THE    ***
C***                   ERROR OCCURS CAN BE UNIQUELY IDENTIFIED BY      ***
C***                   NCALL.                                          ***
C***                                                                   ***
C***     IMPORTANT VARIABLES:                                          ***
C***           IA:  SQUARE ROOT OF IDIMA                               ***
C***           IB:  SQUARE ROOT OF IDIMB                               ***
C***           J:  THE ELEMENT FROM C WHICH IS BEING CALCULATED AT     ***
C***                  EACH STEP                                        ***
C***                                                                   ***
C*************************************************************************
C*************************************************************************
C*************************************************************************
      SUBROUTINE DPROD(A,B,C,IDIMA,IDIMB,IDIMC,NCALL)
      DIMENSION A(IDIMA),B(IDIMB),C(IDIMC)
C***
C***  FIRST THE PROGRAM CHECKS TO MAKE SURE THAT IDIMC=IDIMA*IDIMB.  THEN
C***  IT CHECKS IDIMA AND IDIMB TO SEE IF THEY ARE INDEED SQUARED INTEGERS.
C***  IF ANY OF THESE THREE CONDITIONS ARE VIOLATED, THE PROGRAM IS TERMI-
C***  NATED.
C***
      ID1=IDIMA*IDIMB
      IF(ID1.EQ.IDIMC)GOTO 10
```

```
      TYPE *,' THE DIMENSIONS DO NOT MATCH, SUB DPROD, NCALL=',NCALL
      STOP
10    IA=IFIX(SQRT(FLOAT(IDIMA))+.0001)
      IB=IFIX(SQRT(FLOAT(IDIMB))+.0001)
      IF(IA**2.EQ.IDIMA)GOTO 9
      TYPE *,' THE DIM OF A IN DPROD WAS NOT SQUARE, NCALL=',NCALL
      STOP
9     IF(IB**2.EQ.IDIMB)GOTO 8
      TYPE *,' THE DIM OF B IN DPROD WAS NOT SQUARE, NCALL=',NCALL
      STOP
C***
C***  J IS NOW SET TO ZERO.  THE FOUR DO LOOPS ARE SET UP SO THAT THEY
C***  INCREMENT IN THE SAME MANNER AS THE LEXICOGRAPHIC ORDERING FOR C.
C***  THUS, J IS INCREMENTED BY ONE EVERY TIME THROUGH THE INNERMOST
C***  LOOP TO KEEP TRACK OF THE ELEMENT IN C WHICH IS BEING CALCULATED.
C***  THE DELTA FUNCTION IS USED FOR THE IDENTITY TENSOR.  IN THE
C***  LINE WHICH CALCULATES C(J), THERE ARE TWO TERMS.  THE FIRST REP-
C***  RESENTS THE TENSOR PRODUCT OF A AND AN IDENTITY TENSOR THE SAME
C***  SIZE AS B.  THE IDENTITY TENSOR CAN BE SEEN TO BE THE SAME SIZE AS
C***  B BY NOTICING THAT DELTA'S TWO ARGUMENTS EACH VARY FROM ONE TO IB.
C***  THE TERM CAN BE SEEN TO BE A TENSOR PRODUCT BY NOTICING THAT THE
C***  LEXICOGRAPHIC VARIATION OF THE ARGUMENTS OF A AND THE IDENTITY
C***  TENSOR IS CORRECT IN THE SENSE THAT THE COVARIANT PART OF THE
C***  IDENTITY IS VARIED FIRST, THEN THE COVARIANT PART OF A, THEN THE
C***  CONTRAVARIANT PART OF THE IDENTITY, THEN THE CONTRAVARIANT PART OF
C***  A.  THIS ORDERING IS OBVIOUSLY THE SAME AS THAT IN C, SO THIS TERM
C***  IS THE EXPECTED TYPE OF TENSOR.  SIMILAR COMMENTS CAN BE MADE ABOUT
C***  THE SECOND TERM.  IN PARTICULAR, THE DELTA FUNCTION'S ARGUMENTS RANGE
C***  FROM ONE TO IA EACH, THUS THE IDENTITY TENSOR IT REPRESENTS IS THE
C***  SAME SIZE AS A, AND THE ORDERING IS SUCH THAT THE TERM IS THE SAME
C***  TYPE AS THE FIRST TERM AND C.
C***
8     J=0
      DO 13 I1=1,IA
      DO 13 I2=1,IB
      DO 13 I3=1,IA
      DO 13 I4=1,IB
      J=J+1
13    C(J)=A((I1-1)*IA+I3)*DELTA(I2,I4)+DELTA(I1,I3)*B((I2-1)*IB+I4)
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C***********************************************************************
C***********************************************************************
C***********************************************************************
C***                                                                 ***
C***            TITLE:  DELTA                                        ***
C***                                                                 ***
C***            DELTA IS A FUNCTION WHICH IS USED PRIMARILY IN THE   ***
C***    SUBROUTINE DPROD TO MAKE THE CALCULATIONS MUCH MORE INTUITIVE. ***
C***    IT IS A REAL FUNCTION WITH INTEGER VARIABLES.                ***
C***                                                                 ***
C***    INPUT VARIABLES:  I, J                                       ***
C***                                                                 ***
C***    FUNCTION:                                                    ***
C***                    I===J ---> DELTA=1.0                         ***
C***                    I=/=J ---> DELTA=0.0                         ***
C***                                                                 ***
C***********************************************************************
C***********************************************************************
C***********************************************************************
      FUNCTION DELTA(I,J)
      IF(I.NE.J)GO TO 1
      DELTA=1.
      RETURN
1     DELTA=0.
      RETURN
      END
```

```
C***********************************************************************
C***********************************************************************
C***********************************************************************
C***                                                                ***
C***            TITLE:  PERM                                        ***
C***            AUTHOR:  JOSEPH A. O'SULLIVAN                       ***
C***                                                                ***
C***            SUBROUTINE PERM IS A REORDERING OPERATOR WHICH PERFORMS ***
C***    A (AP,AQ) PERMUTATION ON THE COVARIANT BASIS ELEMENTS OF A.  ***
C***    IT TAKES THE AP BASIS ELEMENTS CORRESPONDING TO THE COVARIANT ***
C***    POWERS OF THE CONTROL VARIABLE AND PLACES THEM AFTER THE AQ  ***
C***    BASIS ELEMENTS CORRESPONDING TO THE COVARIANT POWERS OF THE  ***
C***    INDEPENDENT VARIABLE.  THAT IS TO SAY THAT THE LEXICOGRAPHIC ***
C***    ORDERING, THE THE COVARIANT BASIS ELEMENTS CORRESPONDING TO THE ***
C***    CONTROL VARIABLE WILL BE PERMUTED BEFORE THOSE OF THE INDEPEN- ***
C***    DENT VARIABLE.  THIS IS USUALLY DONE PRIOR TO A CONTRACTION  ***
C***    IN WHICH THE COVARIANT CONTROL VARIABLE IS INVOLVED.         ***
C***                                                                ***
C***    CALLING VARIABLES:                                          ***
C***            A:  THE INPUT TENSOR                                ***
C***            B:  THE OUTPUT TENSOR                               ***
C***            A1:  THE CONTRAVARIANT POWER IN A                   ***
C***            AP:  THE COVARIANT POWER OF THE CONTROL VARIABLE IN A ***
C***            AQ:  THE COVARIANT POWER OF THE INDEPENDENT VARIABLE ***
C***                    IN A                                        ***
C***            IDIMA:  THE DIMENSION OF A                          ***
C***            IDIMU:  THE DIMENSION OF THE CONTROL VARIABLE       ***
C***            IDIMX:  THE DIMENSION OF THE INDEPENDENT VARIABLE   ***
C***            NTYPE:  THE TYPE OF PERMUTATION PERFORMED           ***
C***            NCALL:  A DEBUGGING VARIABLE; IF THIS SUBROUTINE IS ***
C***                    CALLED MANY TIMES, THE CALL IN WHICH THE ERROR ***
C***                    OCCURRED CAN BE UNIQUELY SPECIFIED USING THIS ***
C***                    VARIABLE.                                   ***
C***                                                                ***
C***    TYPES OF PERMUTATIONS:                                      ***
C***            NTYPE=1: THE CONTRAVARIANT POWERS OF A ARE OF THE SAME ***
C***                    TYPE AS THE COVARIANT AQ.                   ***
C***            NTYPE=2: THE CONTRAVARIANT POWERS OF A ARE OF THE SAME ***
C***                    TYPE AS THE COVARIANT AP.                   ***
C***            NTYPE=3: A HAS NO CONTRAVARIANT POWERS.  FOR THIS TYPE ***
C***                    A1 MUST BE ZERO.                            ***
C***            NTYPE=10:  A HAS A1 CONTRAVARIANT POWER OF THE CONTROL ***
C***                    VARIABLE AND BOTH AP AND AQ COVARIANT POWER OF ***
C***                    THE INDEPENDENT VARIABLE                    ***
C***            NTYPE=11:  A HAS DIMENSION A1 CONTRAVARIANT PART AND ***
C***                    COVARIANT PARTS AQ AND AP.  THIS IS USED    ***
C***                    MAINLY WHEN AP INVOLVES BOTH CONTROL AND INDE- ***
C***                    PENDENT VARIABLES.                          ***
C***                                                                ***
C***    THE MAPPING IS:                                             ***
C***                    A1                A1                        ***
C***                   A     --------> B                            ***
C***                    AP,AQ             AQ,AP                     ***
```

```
C***                                                                    ***
C***    HERE IT MUST BE NOTED THAT THE AQ HAVE NOT MAGICALLY BECOME     ***
C***    COVARIANT POWER OF THE CONTROL VARIABLE.  THEY ARE STILL COVA-  ***
C***    RIANT POWERS OF THE INDEPENDENT VARIABLE, BUT THE ORDER OF THE  ***
C***    ELEMENTS IN THE TENSOR HAVE CHANGED.                            ***
C***                                                                    ***
C***    IMPORTANT VARIABLES:                                            ***
C***                 ID1:  THE DIMENSION OF THE CONTRAVARIANT PART OF THE ***
C***                       TENSOR                                       ***
C***                 ID:  THE DIMENSION OF A AS CALCULATED FROM THE OTHER ***
C***                       INFORMATION (IDIMU, IDIMX, ID1, AP, AQ)      ***
C***                 I1:  THE DIMENSION OF THE SECOND PART OF TENSOR A TO ***
C***                       BE INCREMENTED IN ITS LEXICOGRAPHIC ORDERING ***
C***                 I2:  THE DIMENSION OF THE FIRST PART OF A TO BE INCRE- ***
C***                       MENTED                                       ***
C***                 N6:  THE ELEMENT IN A WHICH IS BEING MAPPED AT EACH ***
C***                       STEP                                         ***
C***                 J:  THE ELEMENT IN B TO WHICH THE DESIGNATED ELEMENT ***
C***                       IN A IS BEING MAPPED AT EACH STEP            ***
C***                                                                    ***
C***********************************************************************
C***********************************************************************
C***********************************************************************
        SUBROUTINE PERM(A,B,A1,AP,AQ,IDIMA,IDIMU,IDIMX,NTYPE,NCALL)
        INTEGER A1,AP,AQ
        DIMENSION A(IDIMA), B(IDIMA)
C***
C***    FIRST THE PROGRAM CHECKS THE DIMENSIONS IN THE CALLING STATEMENT TO
C***    MAKE SURE THEY AGREE WITH EACH OTHER.  IF THE DIMENSION OF A AS CAL-
C***    CULATED FROM THE OTHER INFORMATION DOES NOT EQUAL IDIMA, THE PROGRAM
C***    IS TERMINATED.
C***
        ID1=1
        IF(NTYPE.EQ.1)ID1=IDIMX**A1
        IF(NTYPE.EQ.2)ID1=IDIMU**A1
        IF(NTYPE.EQ.10)GO TO 30
        IF(NTYPE.EQ.11)ID1=A1
        ID=IDIMX**AQ*IDIMU**AP*ID1
        IF(ID.EQ.IDIMA)GOTO 10
?       TYPE *,' THE DIMENSIONS DO NOT AGREE IN SUB PERM CALL=',NCALL
        STOP
10      I1=IDIMX**AQ
        I2=IDIMU**AP
C***
C***    FOR NTYPE=1, 2, OR 3, THE VARIABLES I1, I2, AND ID1 ARE CALCU-
C***    LATED ABOVE.  FOR NTYPE=10, THEY ARE CALCULATED AFTER LINE 30.
C***    THE DO LOOPS BELOW ARE SET UP SO THAT THEY INCREMENT IN THE
C***    SAME MANNER AS B.  THUS J IS SET TO ZERO AND THEN INCREMENTED
C***    EACH TIME THROUGH THE INNERMOST LOOP TO KEEP TRACK OF THE ELE-
C***    MENT IN B TO WHICH THE ELEMENT IN A IS BEING MAPPED.  N6, THE
C***    ELEMENT IN A BEING MAPPED, IS CALCULATED USING N3, N4 AND N5
C***    AND THE KNOWLEDGE OF THEIR RELATIONSHIP TO THE ELEMENT OF THE
C***    TENSOR BEING MAPPED TO.  FOR INSTANCE, IF N5 IS GREATER THAN
C***    ONE, IT IS KNOWN THAT THE DESIRED ELEMENT OF A, N6, IS AT LEAST
C***    (N5-1)*I1.  ALSO, IF N3 IS GREATER THAN ONE, N6 IS AT LEAST
```

```
C***      (N5-1)*I1*I2,
C***
11        J=0
          DO 20 N3=1,ID1
          DO 20 N4=1,I1
          DO 20 N5=1,I2
          J=J+1
          N6=((N3-1)*I2+N5-1)*I1+N4
20        B(J)=A(N6)
          RETURN
C***
C***      FOR MTYPE=10, THE CONTRAVARIANT POWERS ARE OF THE CONTROL VARIABLE
C***      AND BOTH AP AND AQ ARE COVARIANT POWERS OF THE INDEPENDENT VARIABLE.
C***      ID, ID1, I1, AND I2 ARE CALCULATED FROM THIS KNOWLEDGE, AND THEN
C***      CONTROL IS PASSED BACK TO THE DO LOOPS ABOVE.
C***
30        ID=IDIMU**A1*IDIMX**(AP+AQ)
          IF(ID.NE.IDIMA)GO TO 9
          ID1=IDIMU**A1
          I1=IDIMX**AQ
          I2=IDIMX**AP
          GO TO 11
          END
```

C-3

APPENDIX B

This appendix contains several plots to support the validity of the regions of usefulness. Specifically, plots are included which show the significance of boundary points, thereby showing the dramatic decrease in performance and even instability of the points outside of the region. Plots are also included to compare the performance of the different controllers in the regions where all are valid. The regions of usefulness will be examined one quadrant at a time. For each set of initial conditions there are six possible plots, each comparing linear, quadratic and third order feedback. The six plots are the first state, the second state, the first control, the second control, the norm (euclidean) of the state and the norm of the control. For example, each of these plots for the initial condition $x(0) = (1.0,0.0)$ is shown in Figures B.1 and B.2. Figure B.1 shows the plots for the states while B.2 shows the plots for the controls. Throughout this appendix the solid line will be for the third order, the dotted line for the second order, and the dashed line for the linear controller.

As noted, Figures B.1 and B.2 correspond to $x(0) = (1.,0.)$. This initial condition is well within all three regions (Figures 5.5 through 5.7) as can be noted by examining the top two plots in Figure B.1. Here it is seen that all three controls are driving the state to zero, and (from the bottom plot in Figure B.1) the final norm is less than .1. For this particular point it is noted that the third order outperforms the other two in all six plots, and the linear outperforms the second order. Here "outperforms" will be a qualitative measure of which controller is best. The criteria used to determine this can be described by examining the bottom plot in Figure B.1. Here the norm of the state using the third order controller is less than or equal to that due to

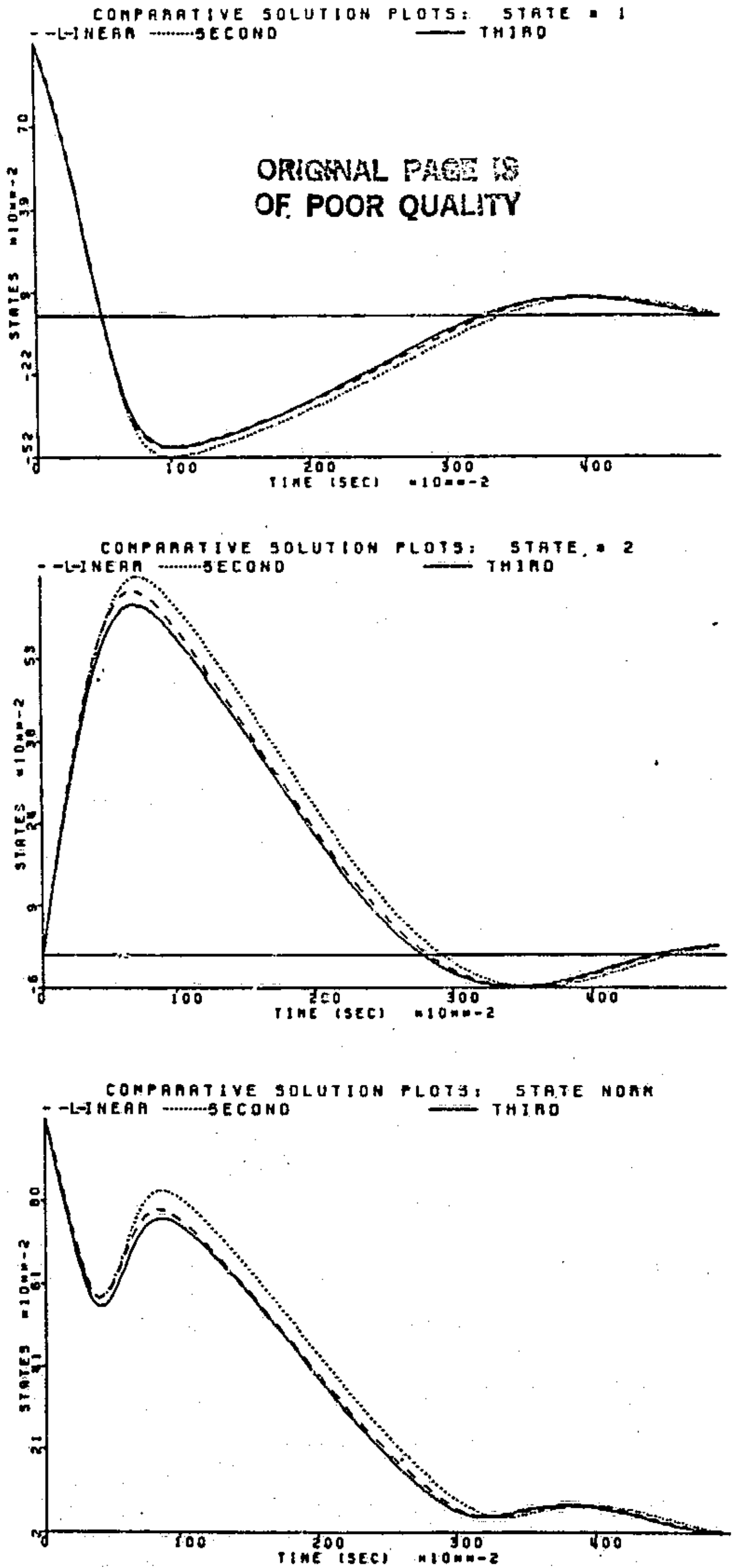


Figure B.1 State plots for x(0) = (1.,0.)

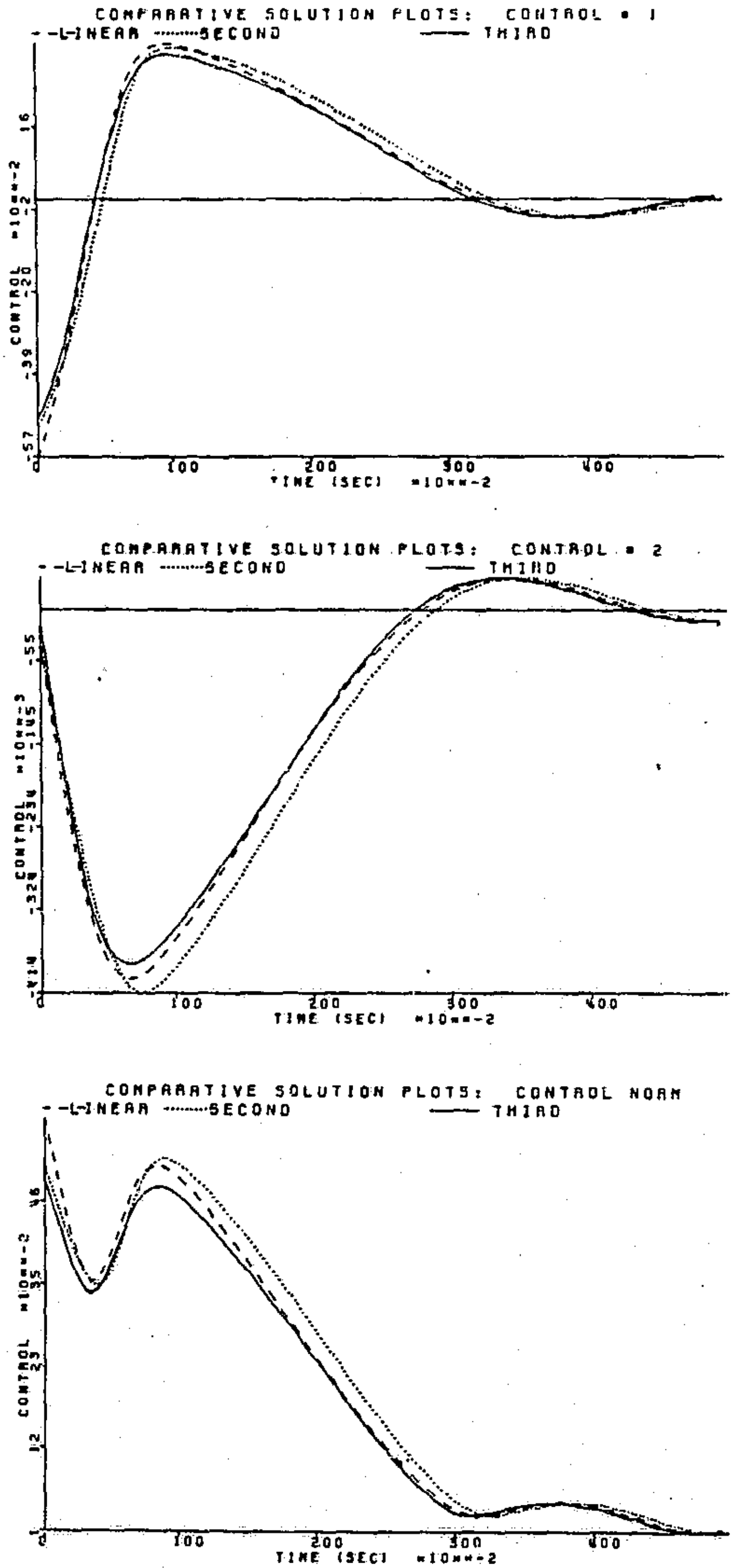


Figure B.2 Control plots for x(0) = (1.,0.)

the linear controller for the entire 5 seconds. While the second order plot is less than the third order for a short time at about 3.3 seconds, the large difference in favor of the third order plot for the first 3 seconds more than makes up for that short interval and the third order is judged better. Similarly the linear is judged better than the second order controller. This particular set of plots does not show the dramatic increase in performance of the third order controller which will be shown with other plots, but it does demonstrate how the controllers will be compared.

Figures B.3 and B.4 show the plots for $x(0) = (.6,.6)$. These plots are similar to those for $x(0) = (1.,0.)$ in that the third order is best, linear second, and second order is third. It should be noted that this is typical of points inside all three regions inside the first quadrant. The second order region, however, is much larger in this quadrant as the next set of plots show.

Figures B.5 and B.6 are the plots for $x(0) = (.8,.8)$. Looking at Figure 5.5, one notices that this is the last point along the $x_1(0) = x_2(0)$ line which is in the linear region. Figures B.7 and B.8 show a point a little further out, $x(0) = (.85,.85)$, and they show the criterion used to determine whether a point is in the region or not. Notice that in Figure B.5 the linear still outperforms the quadratic feedback and all three controllers are driving the states toward zero as desired. Figures B.7 and B.8 show that with slightly larger initial conditions the system with only a linear controller is unstable. The quadratic feedback still drives the norm of the state toward zero as seen in the bottom plot in Figure B.7. Thus, for this point, the border in Figure 5.5 carries a large significance -- it is the furthest it is safe to use linear feedback.

ORIGINAL PAGE IS OF POOR QUALITY

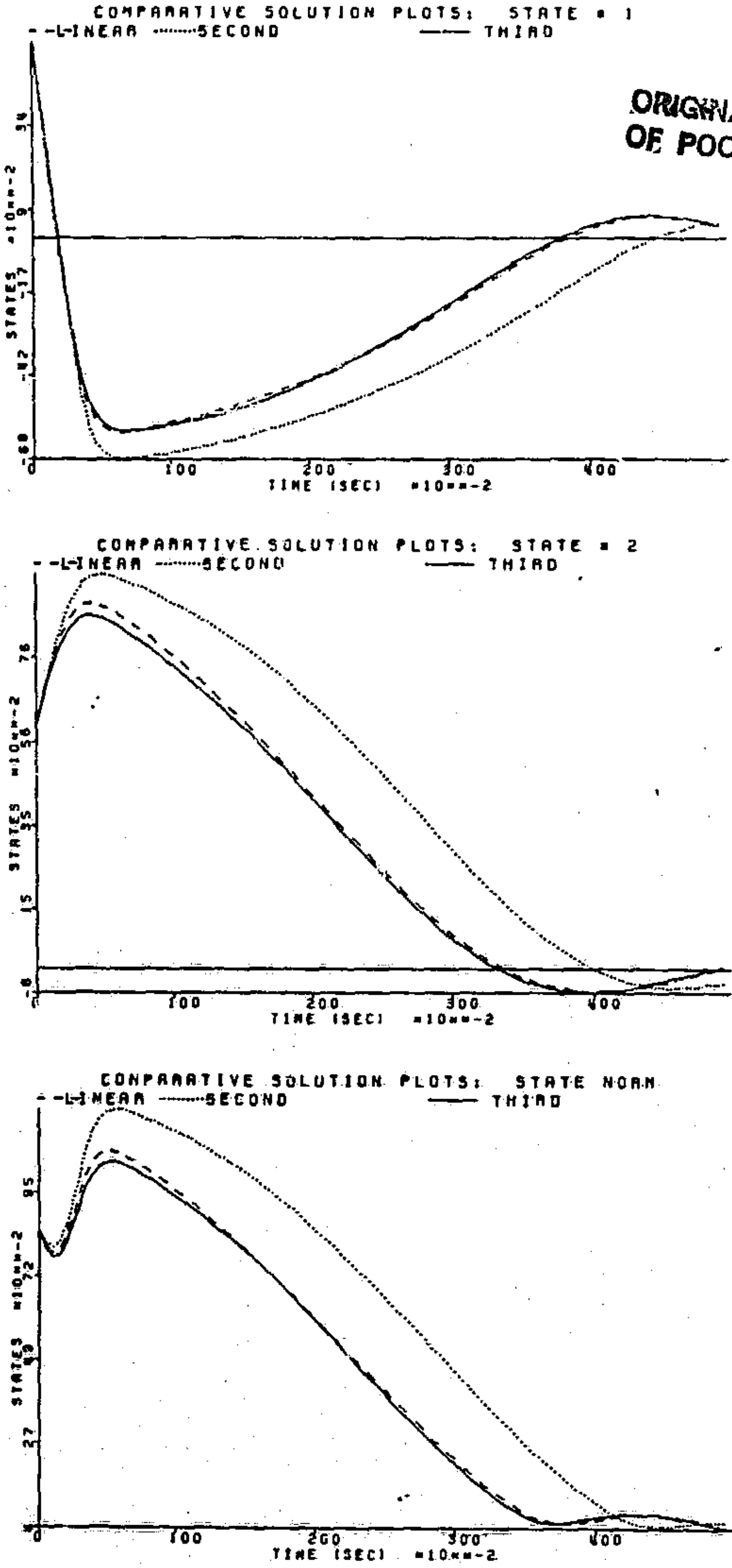


Figure B.3 State plots for x(0) = (.6,.6)

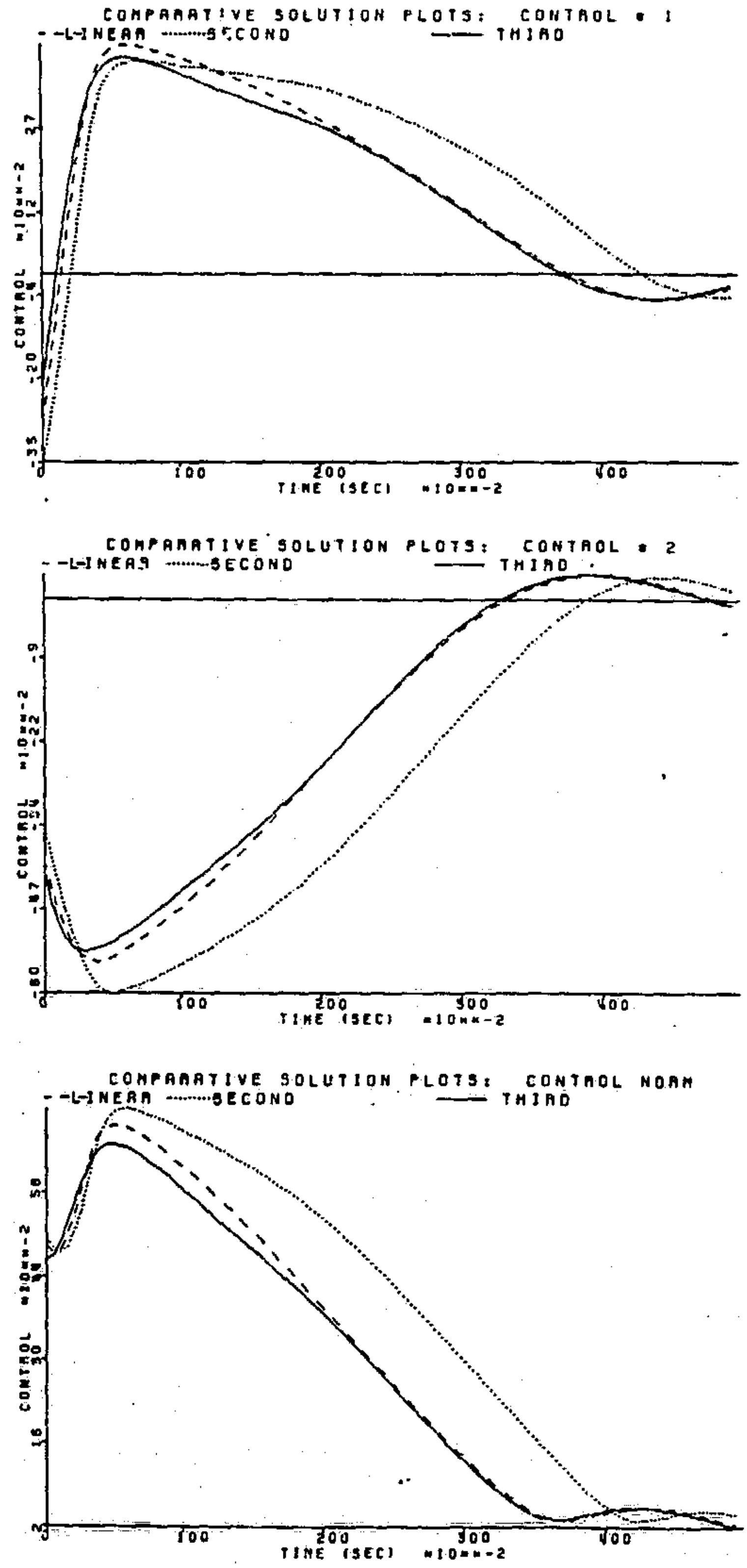


Figure B.4 Control plots for x(0) = (.6,.6)

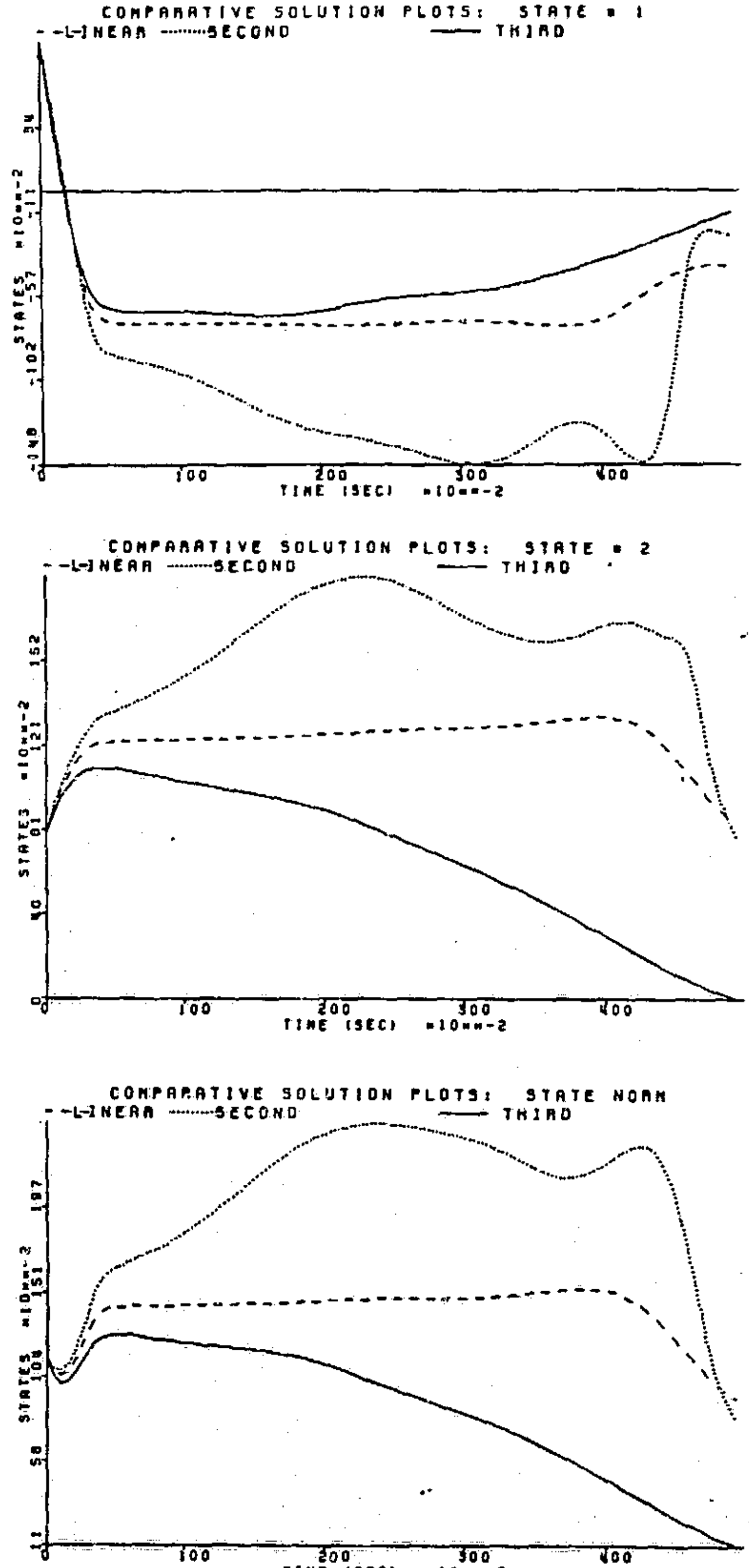


Figure B.5 State plots for x(0) = (.8,.8)

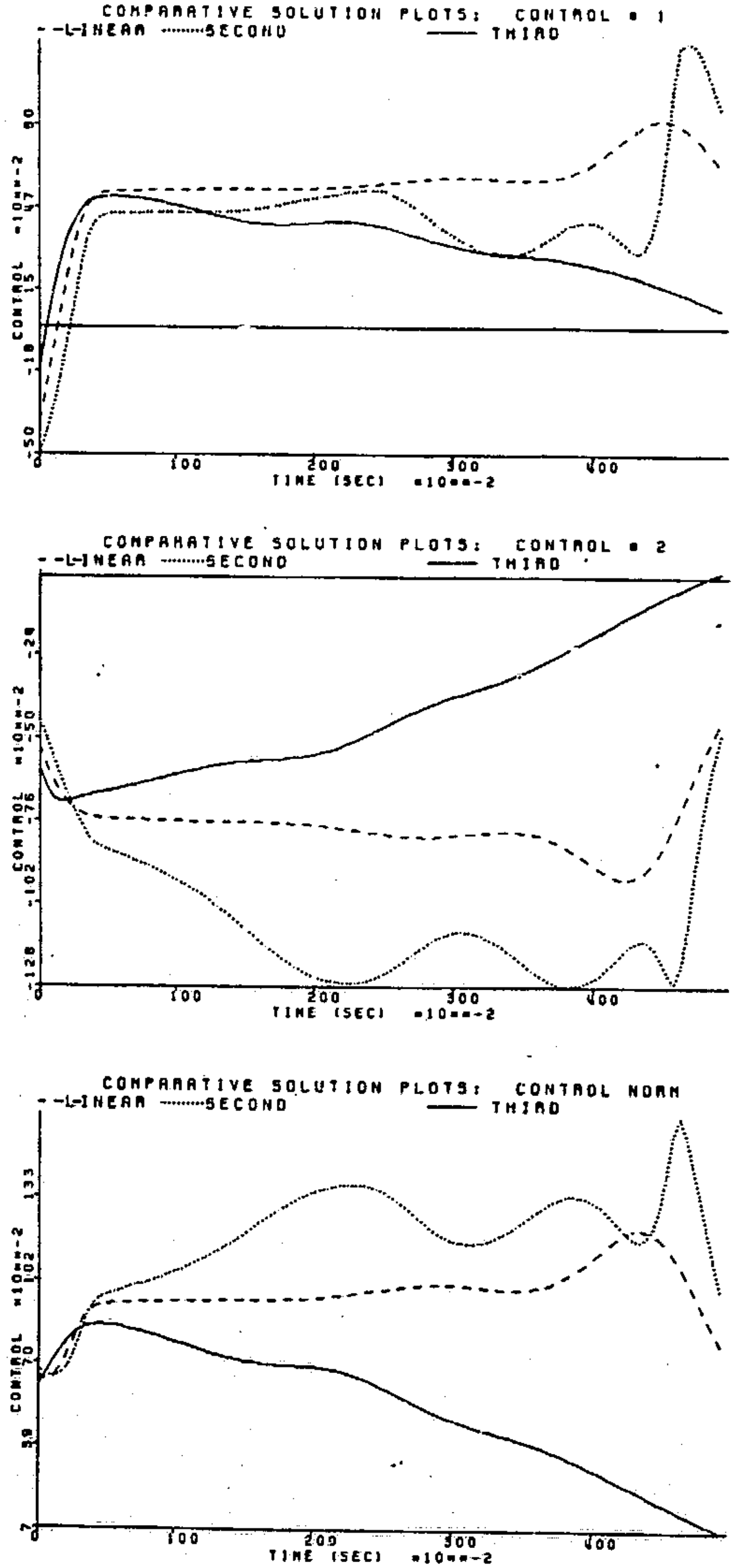


Figure B.6 Control plots for x(0) = (.8,.8)

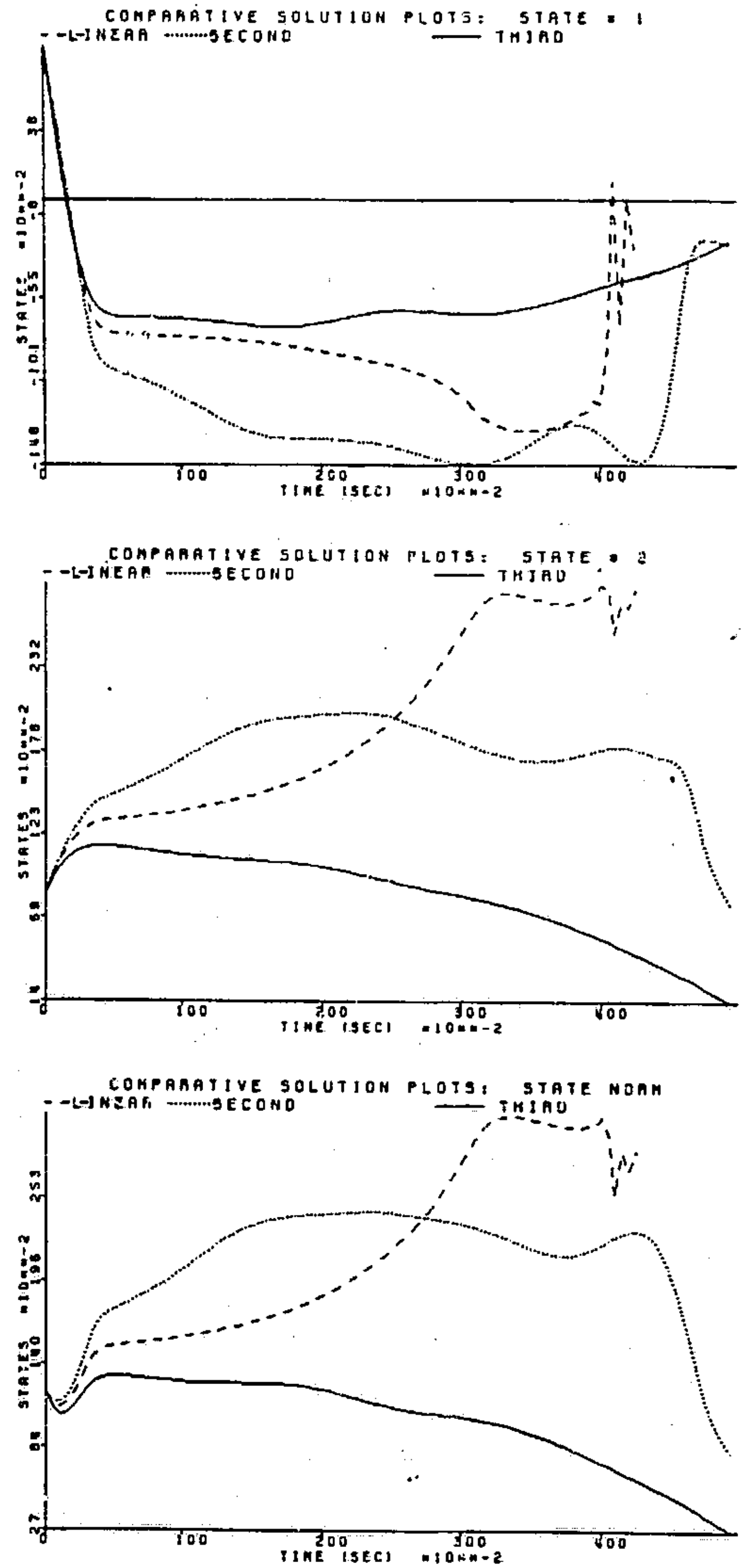


Figure B.7 State plots for x(0) = (.85,.85)
Notice that the linear feedback
goes unstable at 4 seconds.

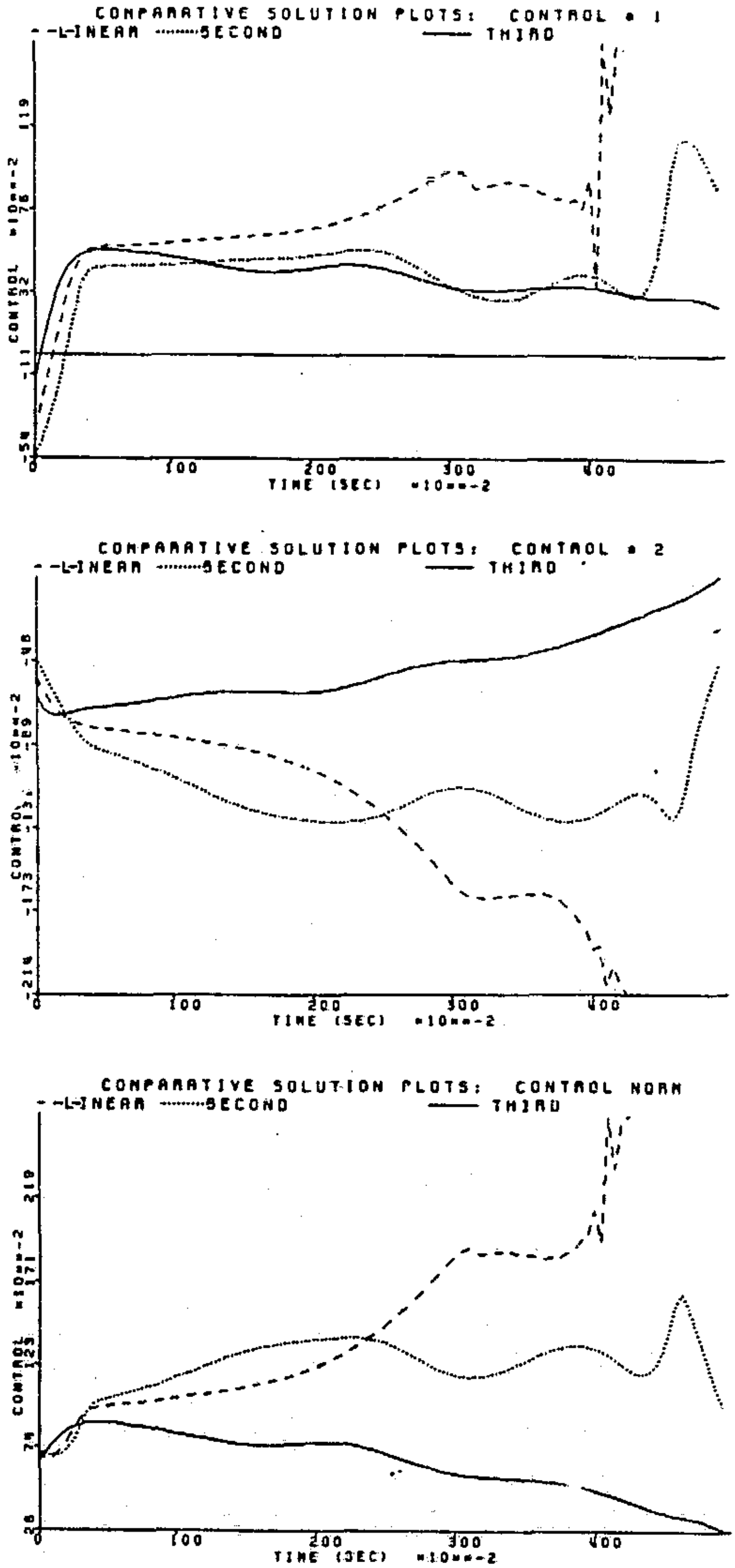


Figure B.8 Control plots for x(0) = (.85,.85)
(Linear goes unstable)

The final plots from the first quadrant are Figures B.9 and B.10 for x(0) = (2.0,0.7). This is well out of the linear region of usefulness and just on the edge of the second order region. The third order is still much better than the second order at this point.

Figures B.11 and B.12, for x(0) = (0.,1.) show trajectories similar to those seen in the middle of the first quadrant. The third order plots outperform the others while the linear plots top the second order plots. The second order feedback at this point is far inferior to the other two.

Points in the second quadrant which are in all three regions have plots of which Figures B.13 and B.14 (x(0) = (-1.1,1.0)) are typical. All three types of feedback are close with the second order being slightly inferior. Another example of this is shown in Figures B.15 and B.16 (x(0) = (-1.5,0.3)) where the linear feedback is slightly inferior. It is seen that as the initial conditions get closer to the $x_2(0) = 0.$ axis the plots show that as each order of feedback is added the trajectories improve. Figures B.17 and B.18 show the plots for x(0) = (-1.5,0.03). The linear is seen to be inferior to each of the others with the third order being the best. The third order is not the best everywhere, however. A close look at the regions in the second quadrant of Figures 5.6 and 5.7 reveals a region above the third order region where the second order feedback is still useful. Plots from a point (x(0) = (-1.1,2.2)) in that area are in Figures B.19 and B.20. Here, the third order plots go unstable immediately and the linear plots go unstable at 4 seconds. Quadratic feedback, however, remains stable. The final plots from the second quadrant are shown in Figures B.21 and B.22 (x(0) = (-1.2,1.7)). The third order plots are better than the second by a wide margin in the states and by slightly less in the controls.

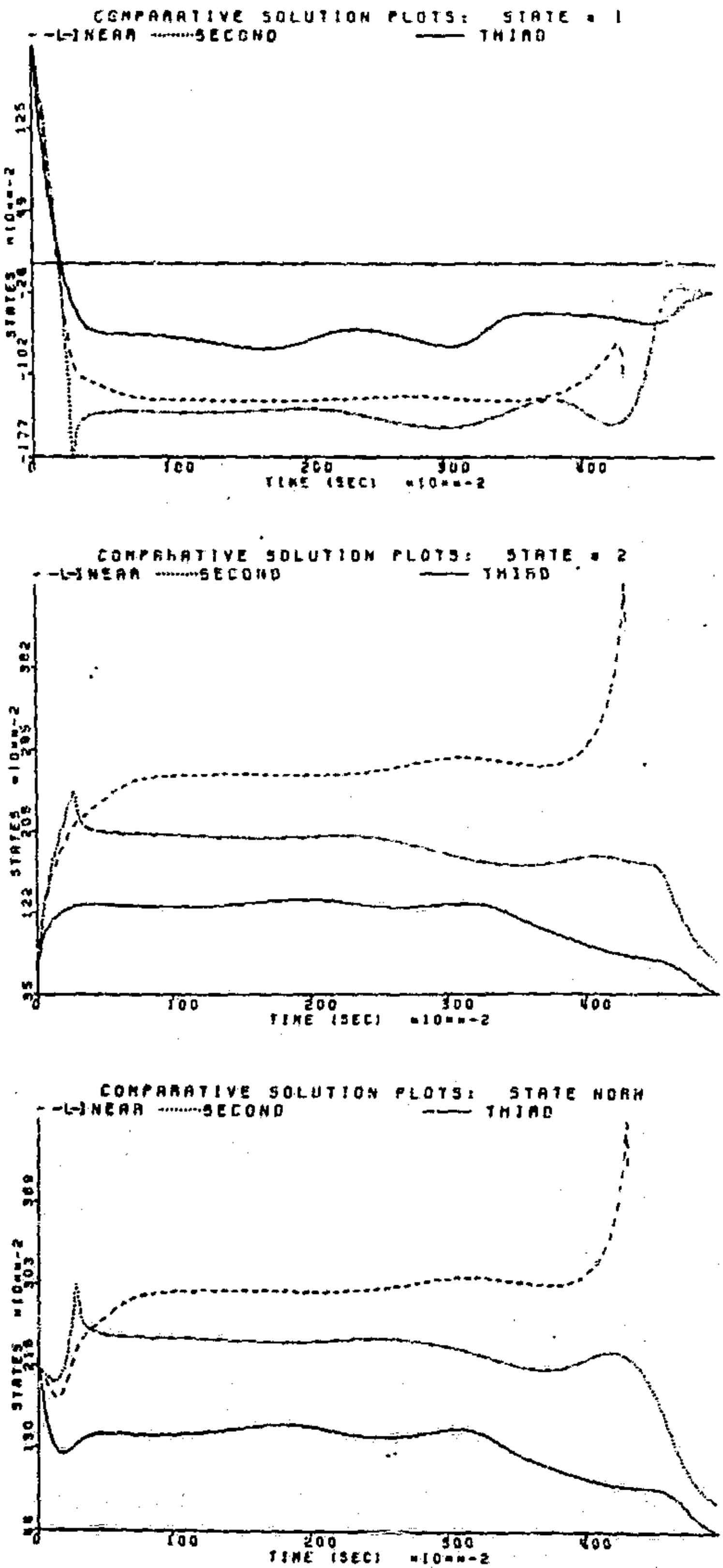
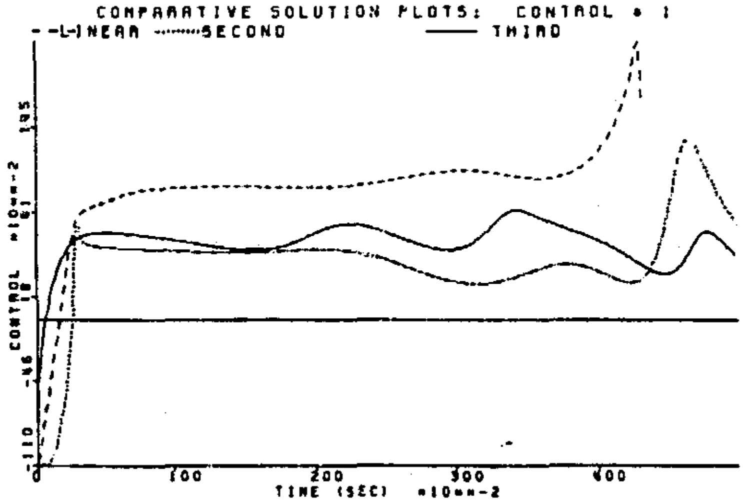
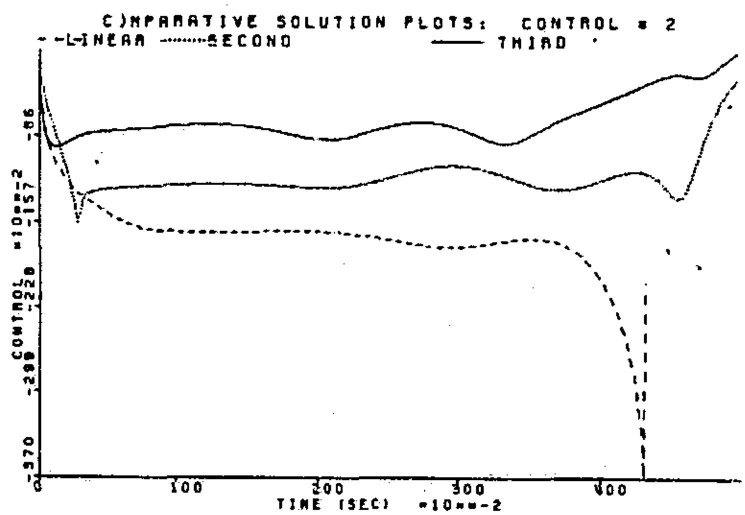


Figure B.9 State plots for x(0) = (2.0,0.7)
(Linear goes unstable)





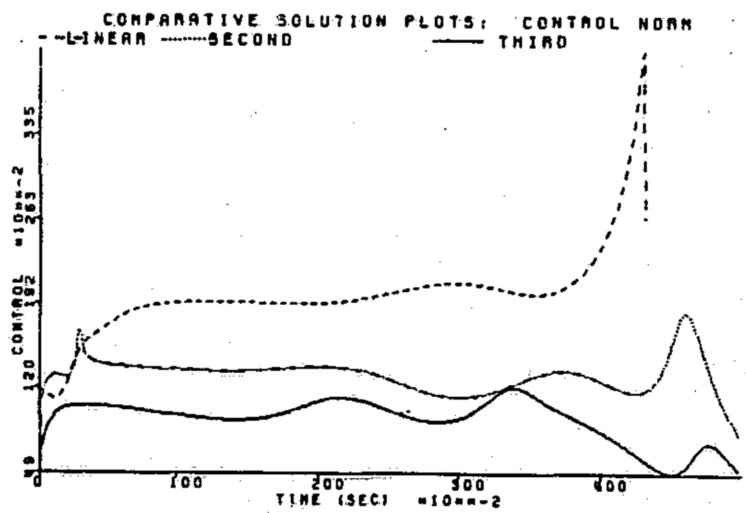


Figure B.10 Control plots for x(0) = (2.0,0.7)
(Linear goes unstable)

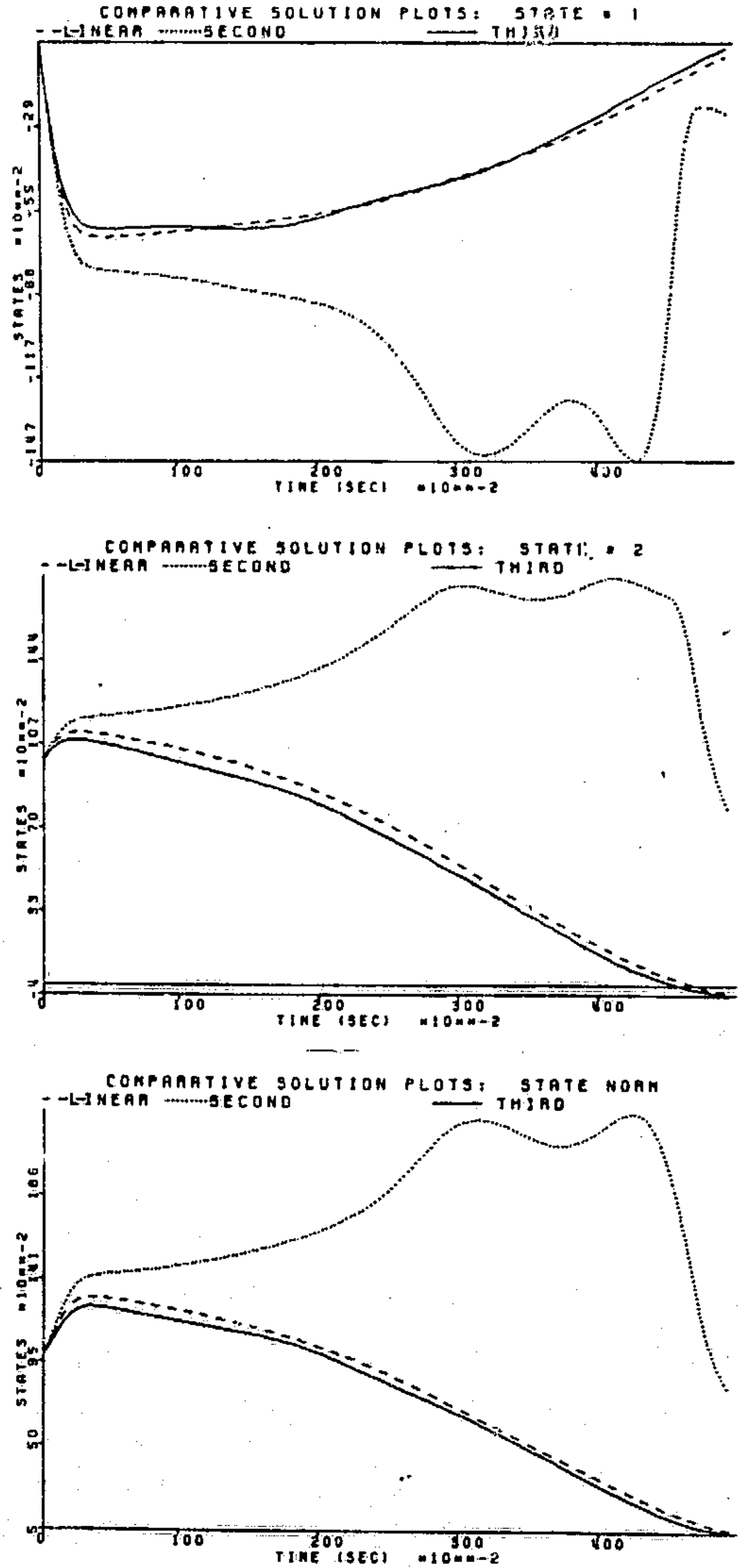


Figure B.11 State plots for x(0) = (0.,1.)

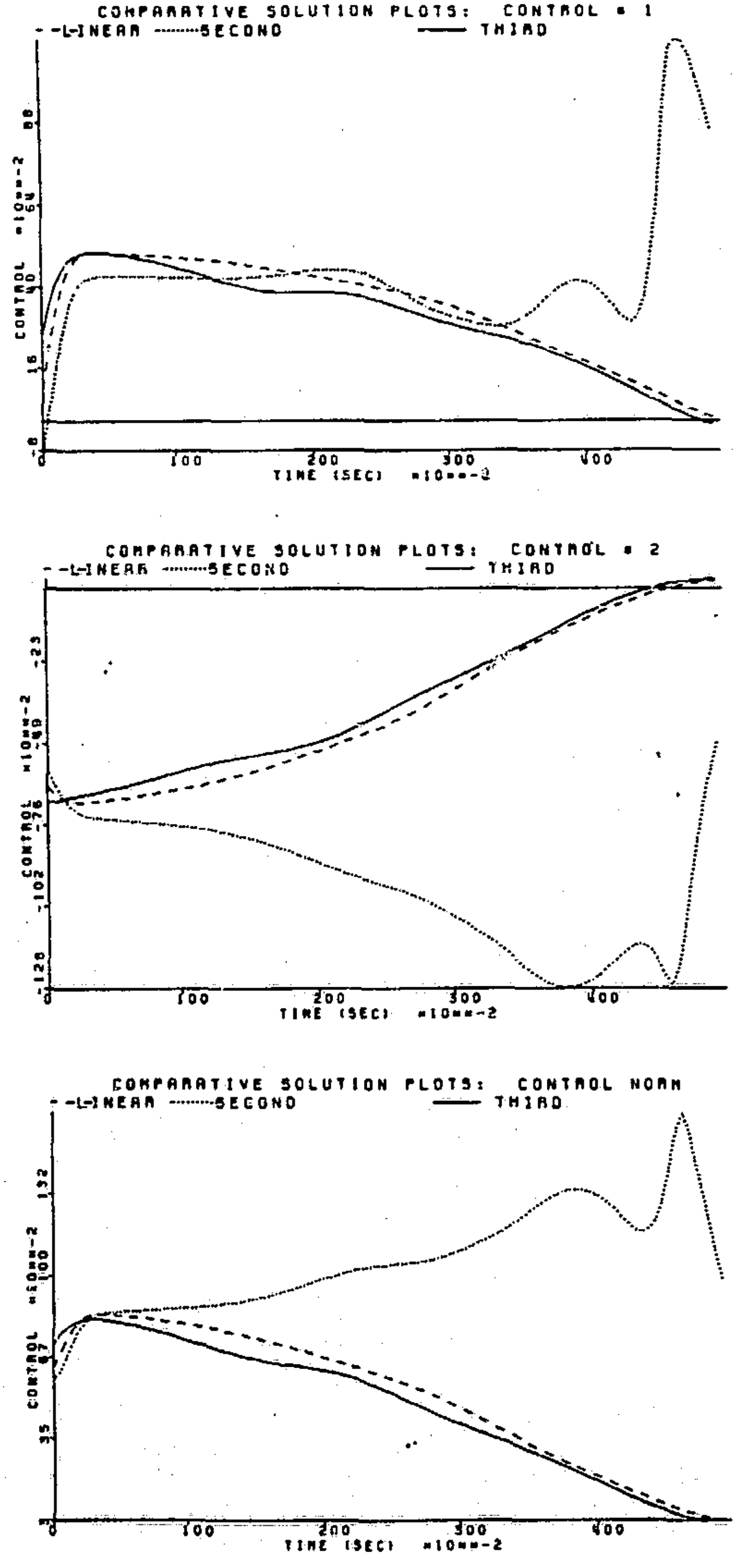
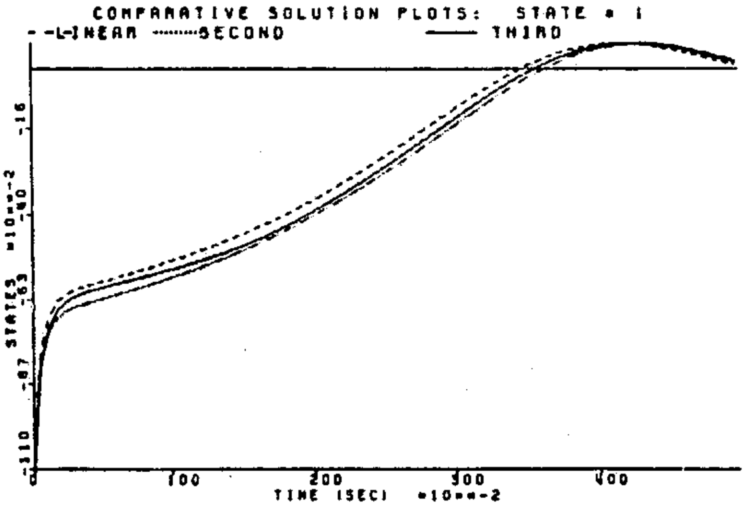
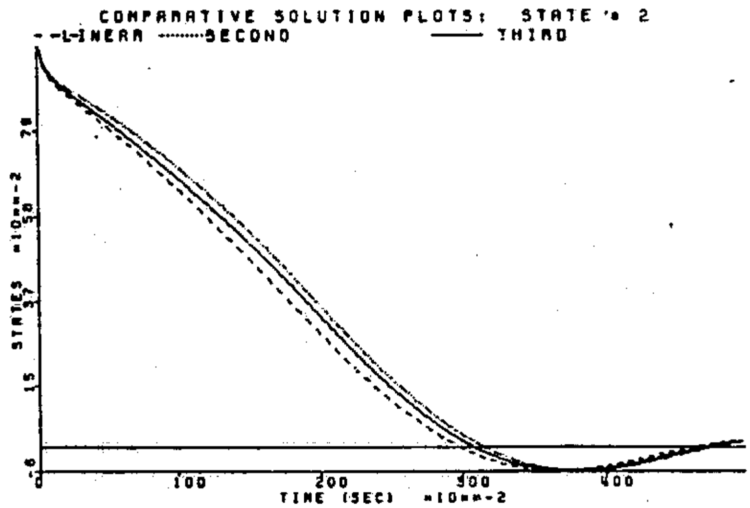


Figure B.12 Control plots for x(0) = (0.,1.)





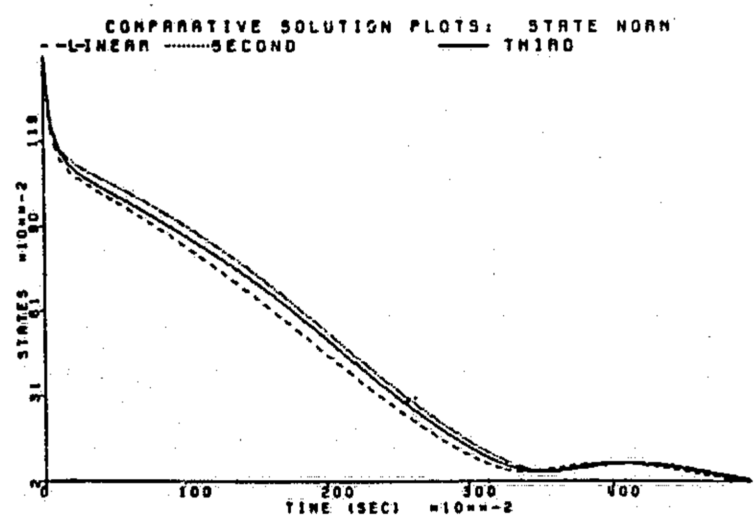


Figure B.13 State plots for $x(0) = (-1.1,1.0)$

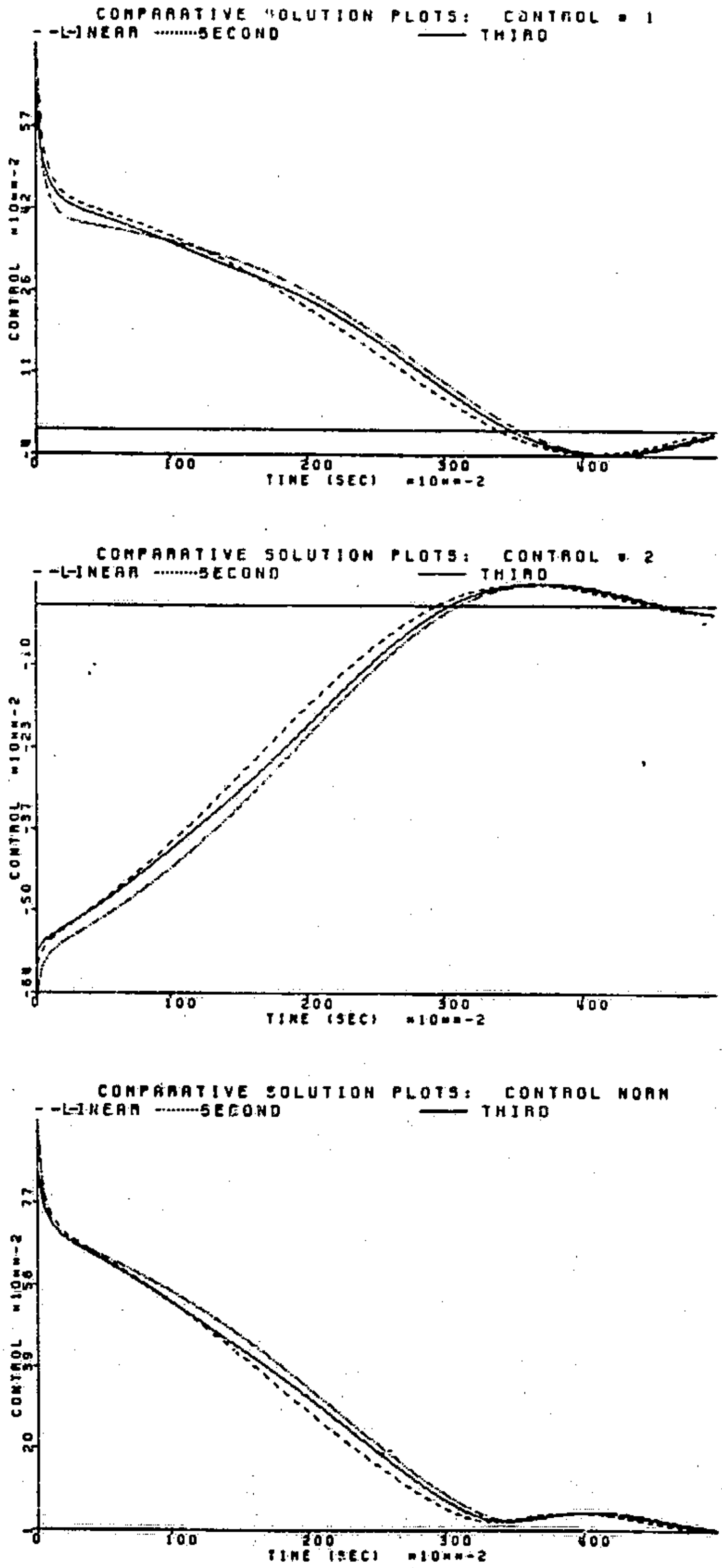


Figure B.14 Control plots for x(0) = (-1.1,1.0)

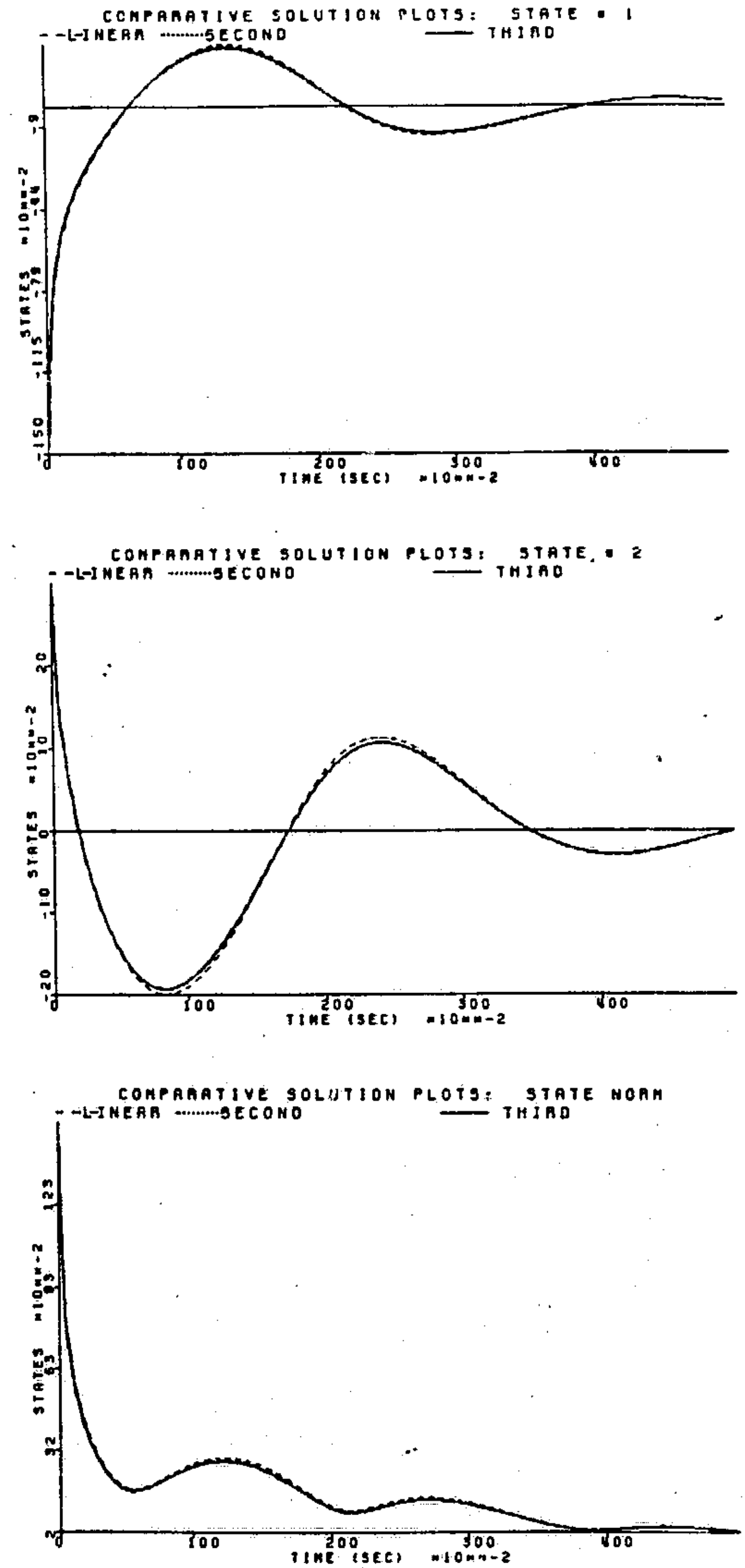


Figure B.15 State plots for x(0) = (-1.5,0.3)

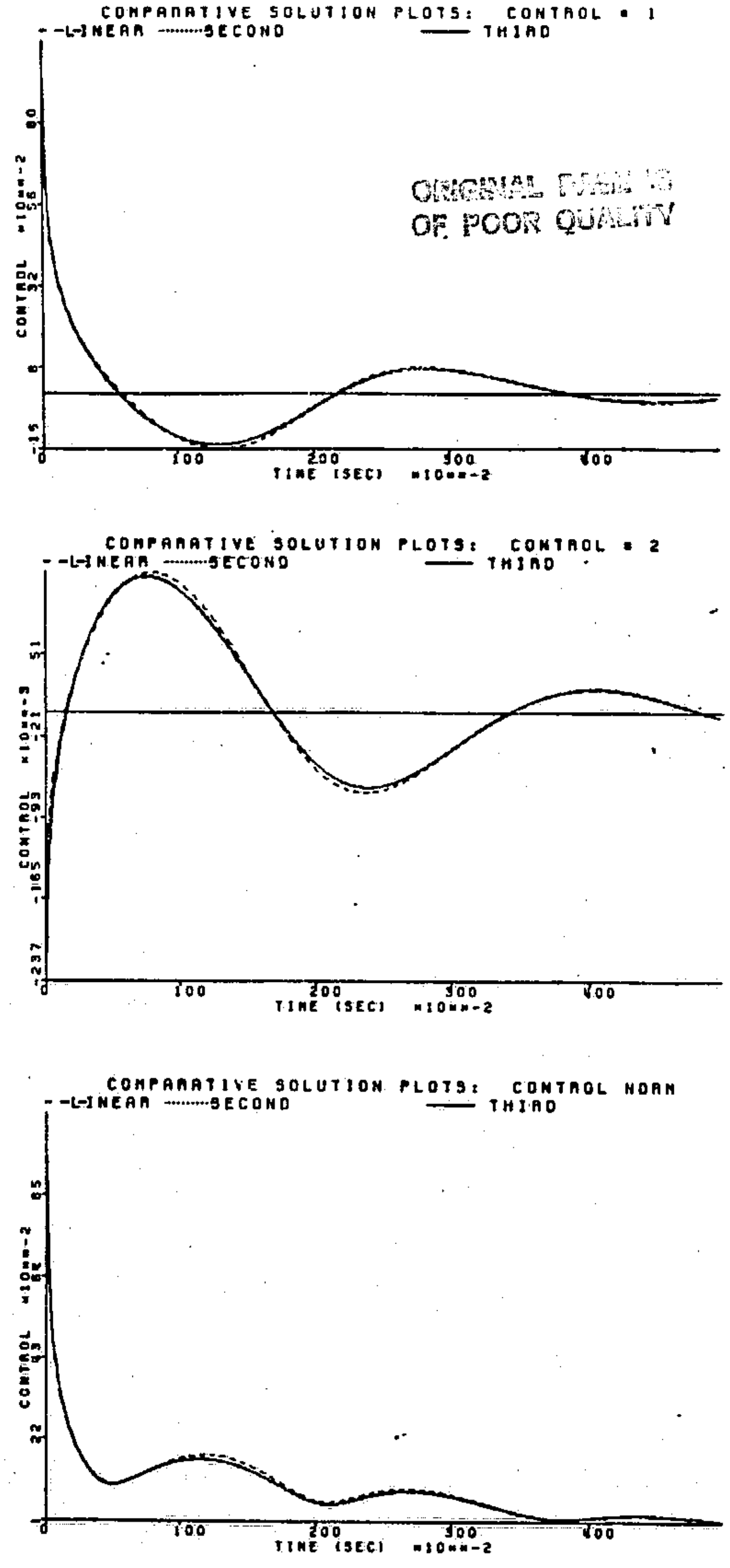


Figure B.16 Control plots for x(0) = (-1.5,0.3)

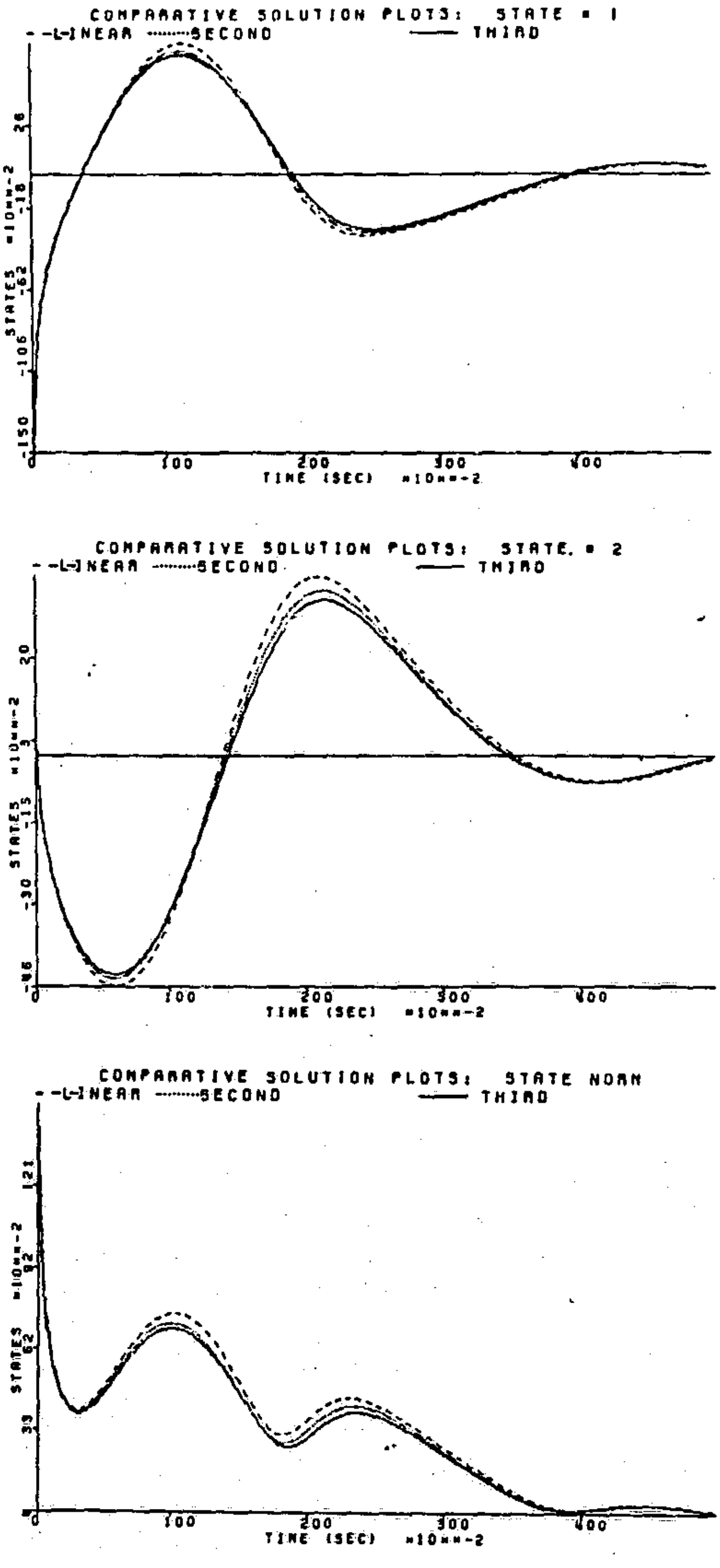


Figure B.17 State plots for x(0) = (-1.5,0.03)

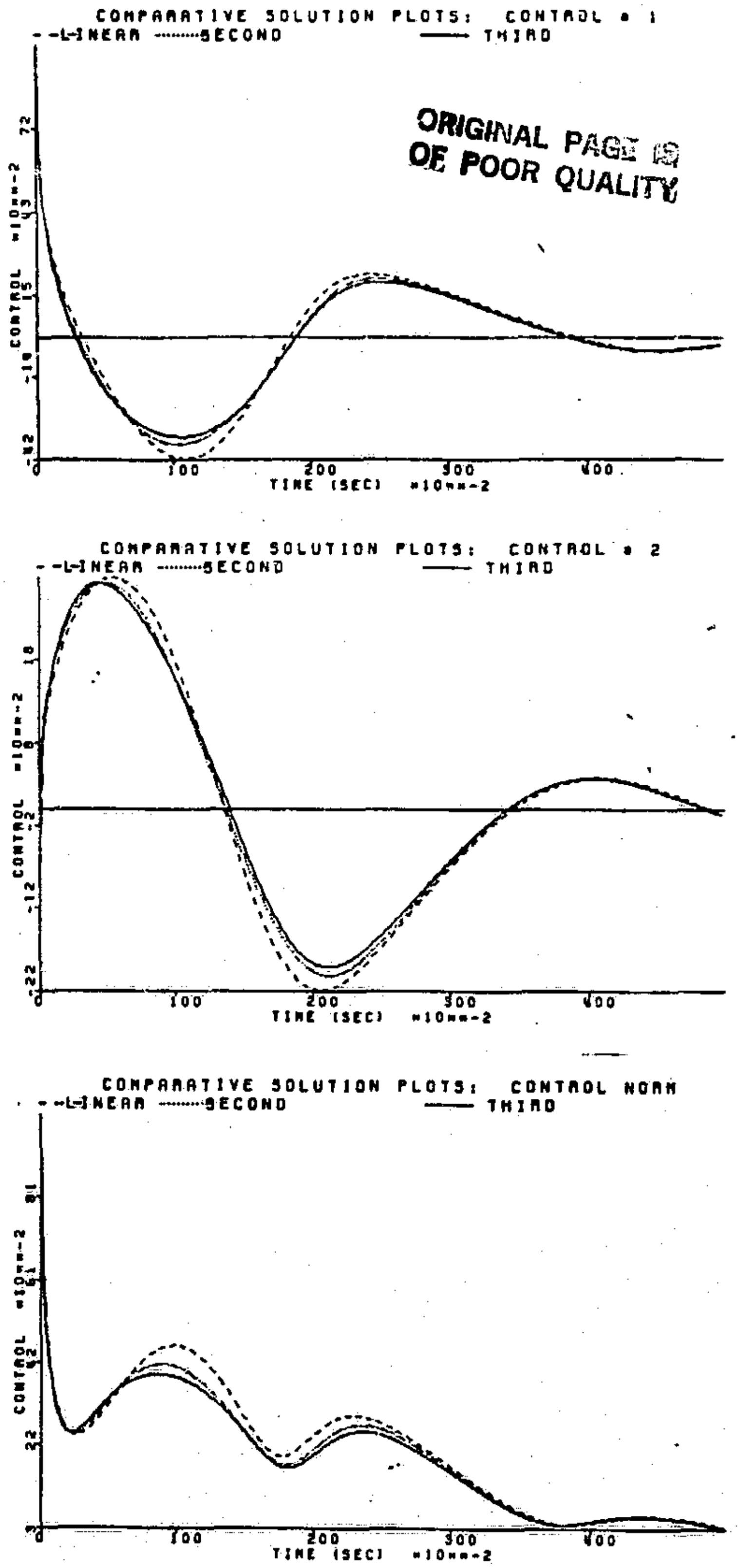


Figure B.18 Control plots for $x(0) = (-1.5,0.03)$

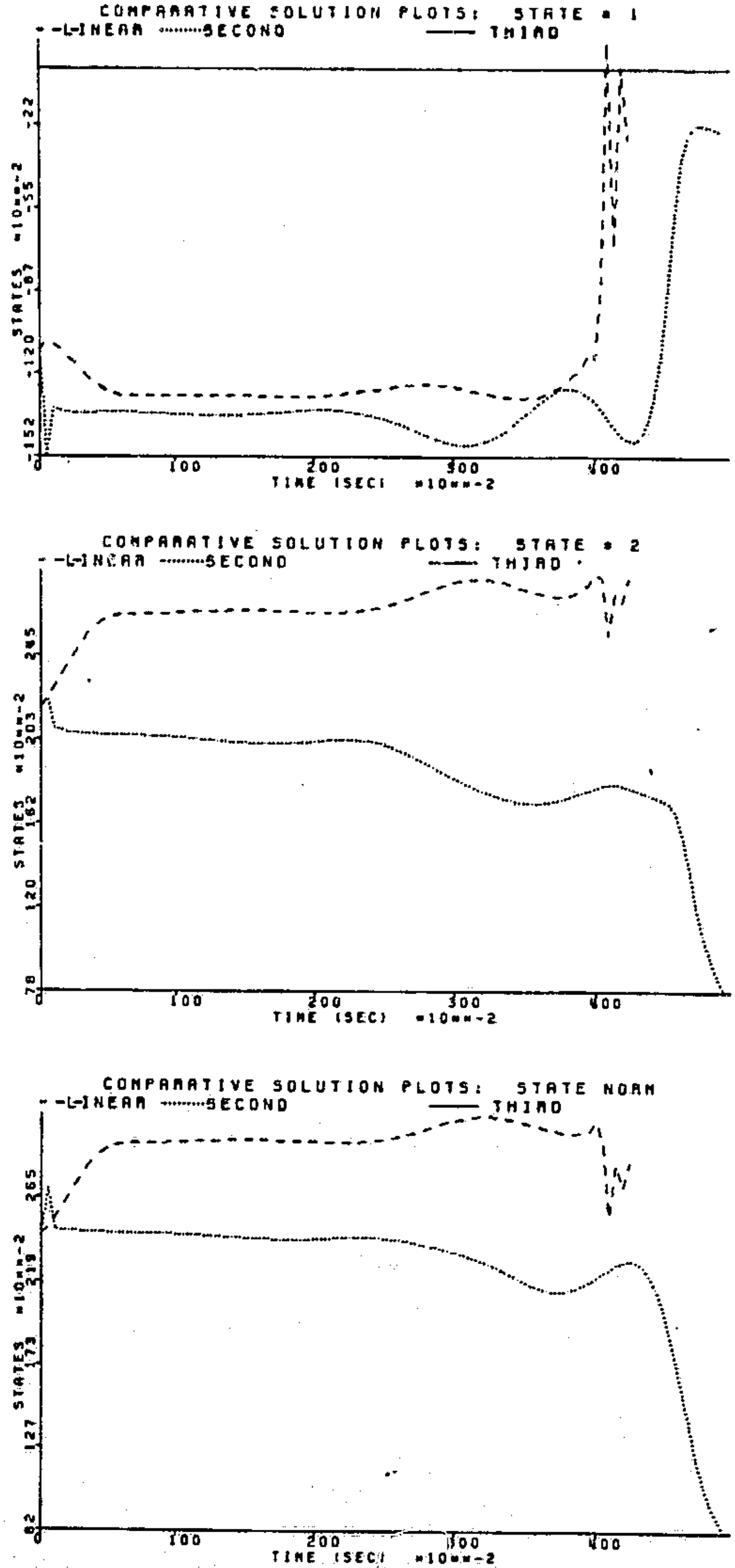


Figure B.19 State plots for $x(0) = (-1.1, 2.2)$
(Third order is unstable immediately;
linear goes unstable)

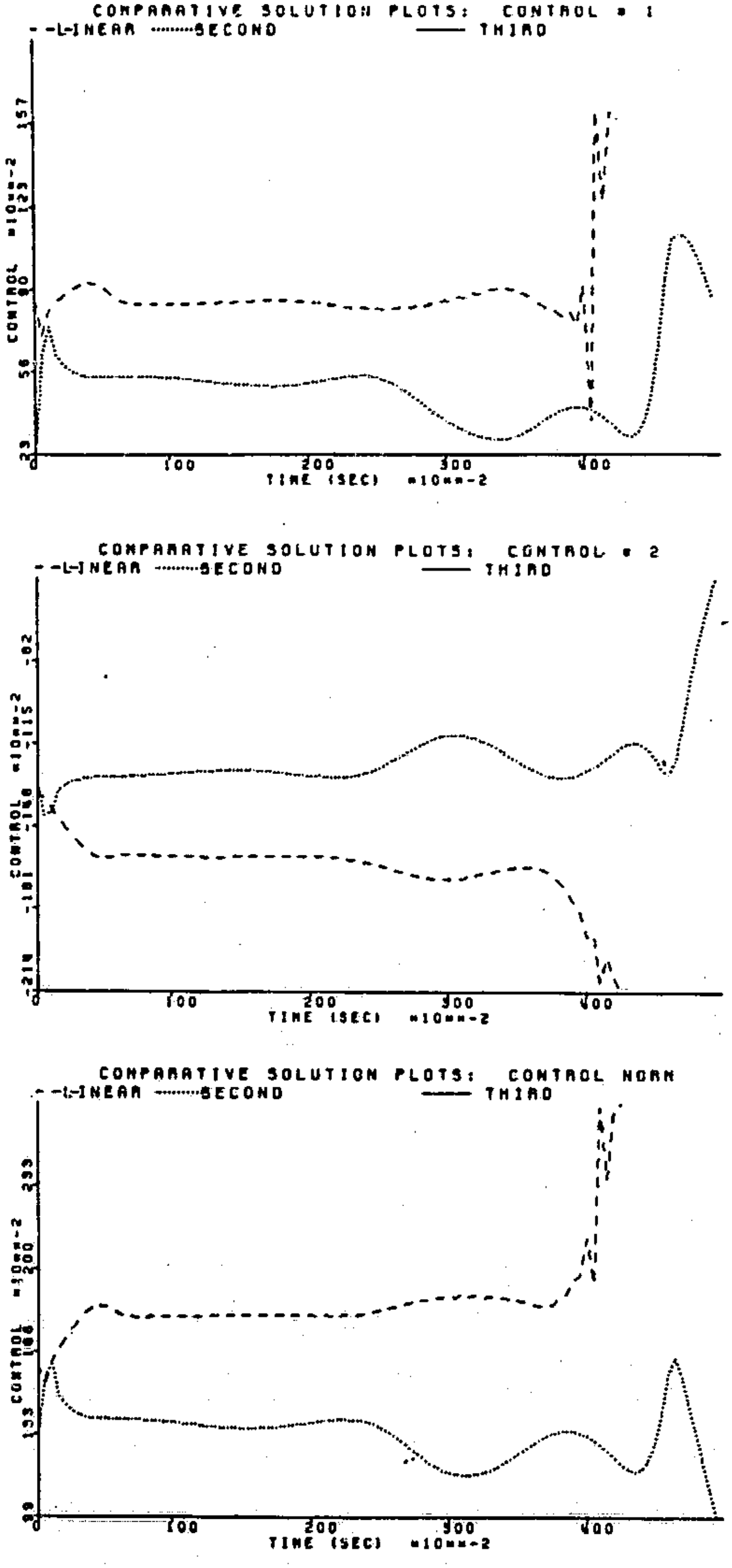


Figure B.20 Control plots for x(0) = (-1.1,2.2)
(Third order is unstable immediately;
linear goes unstable)

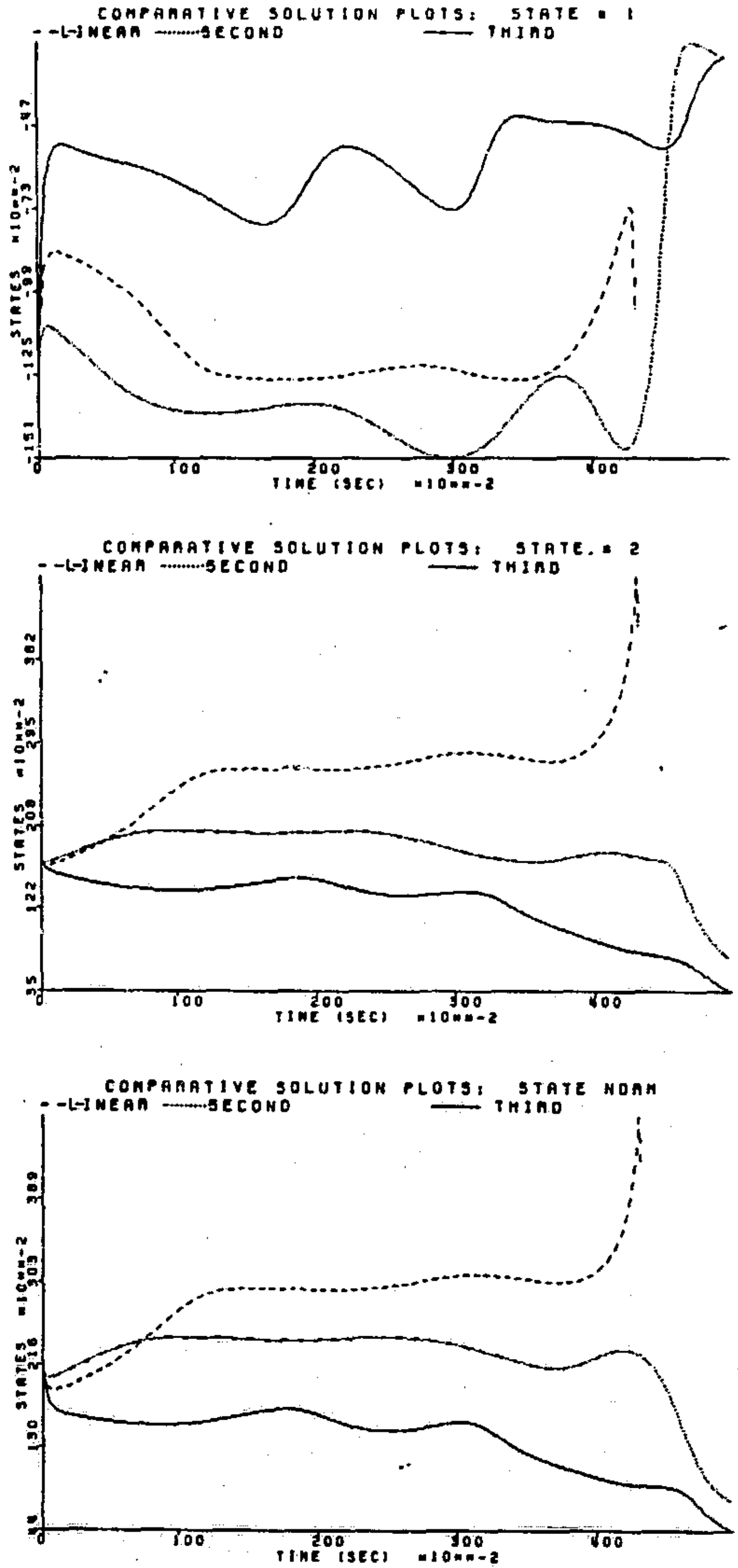


Figure B.21 State plots for $x(0) = (-1.2,1.7)$ (Linear goes unstable)

ORIGINAL PAGE IS
OF POOR QUALITY

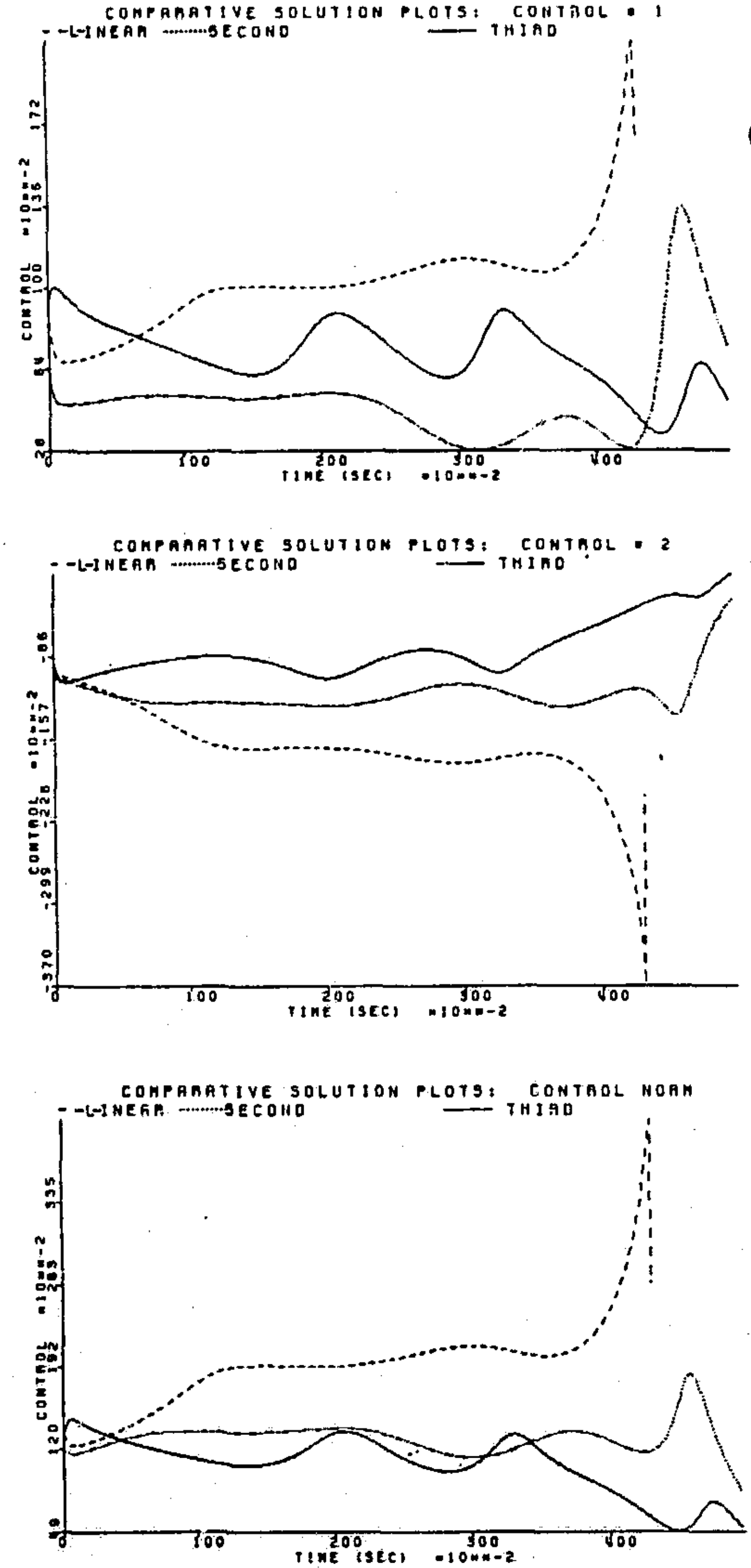


Figure B.22 Control plots for x(0) = (-1.2,1.7)
(Linear goes unstable)

Figures B.23 and B.24 ($x(0) = (-1.,0.)$) are very similar to Figures B.15 and B.16 and demonstrate a trend which continues throughout the third quadrant. The third order is the best, the second order is slightly worse and the linear lags behind. Figures B.25 through B.30 ($x(0) = (-.6,-.6)$, $(-1.5,-.3)$ and $(-1.,-1.)$, respectively) will be analyzed simultaneously. All three show the trend just mentioned. Another trend can be noticed by looking at the top plots in Figures B.26, B.28, and B.30. In Figure B.26 (top) the third order controller is seen to have a flat stretch around .75 seconds, while the second order and linear controllers have peaks. The same phenomenon is seen in Figure B.28. In Figure B.30 one can see that this flat region has become a sort of antinode, peaking in the opposite direction. This seems to be the major difference in the control plots in this quadrant and must account for some of the increased performance of the third order controller. The same type of control plot is seen in Figure B.32 for $x(0) = (-1.6,-0.8)$, Figure B.34 for $x(0) = (-1.3,-1.3)$, and Figure B.36 for $x(0) = (-2.1,-1.2)$. Figures B.33 through B.36 are the most dramatic because for these plots the system with third order feedback is the only stable one. They show that the linear and quadratic feedback systems go unstable immediately while the third order feedback behaves nicely.

Figures B.37 and B.38 ($x(0) = (0.,-1.)$) show the same pattern as the graphs from the third quadrant. The third order controller outperforms the other two, the antinode in the first control having the same effect as before.

The main thrust of the plots from the fourth quadrant is an examination of the boundaries. First of all, note that the point $x(0) = (1.0,-1.0)$ is in the linear region (Figure 5.5) while $x(0) = (1.1,-1.1)$ is not. Figures B.39

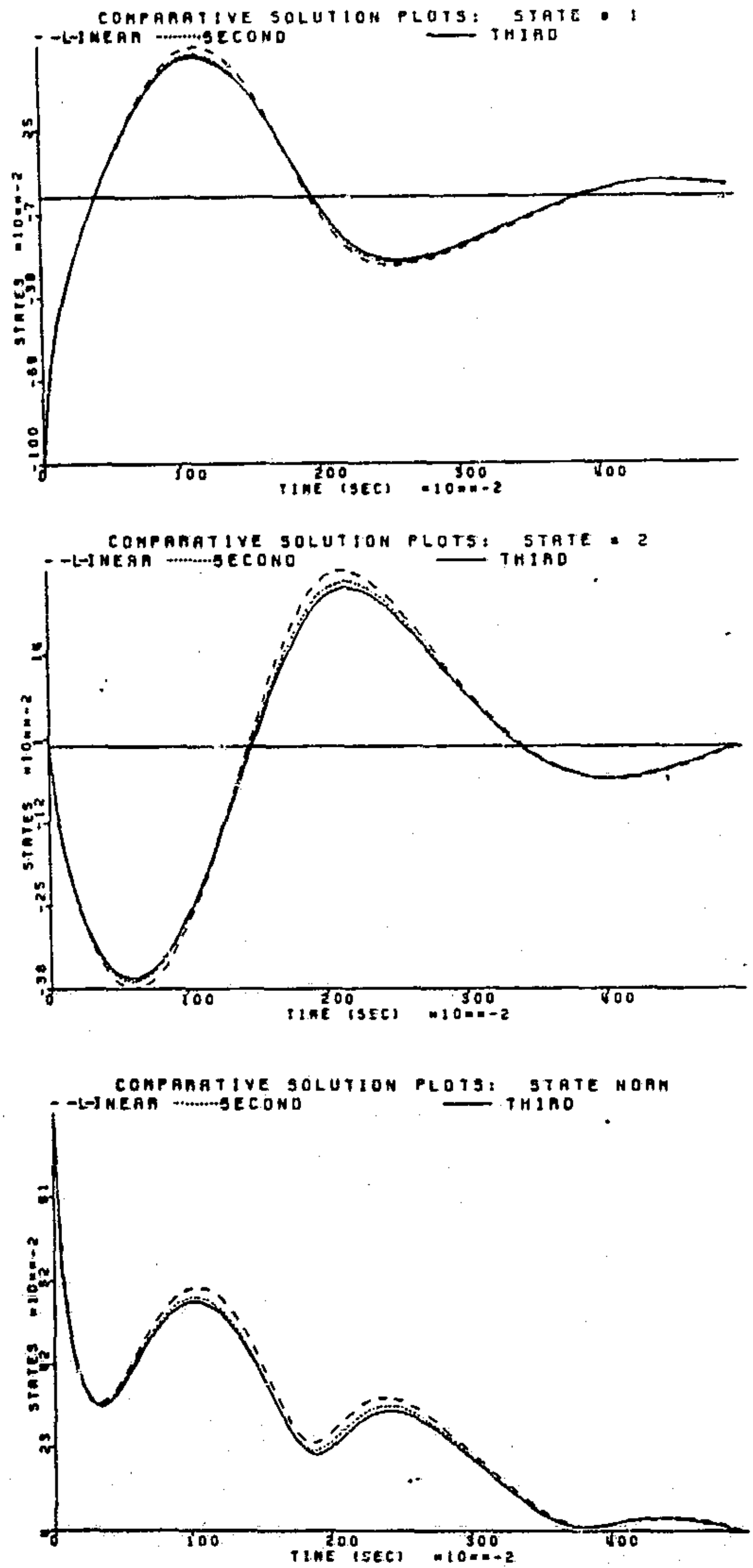


Figure B.23 State plots for x(0) = (-1.,0.)

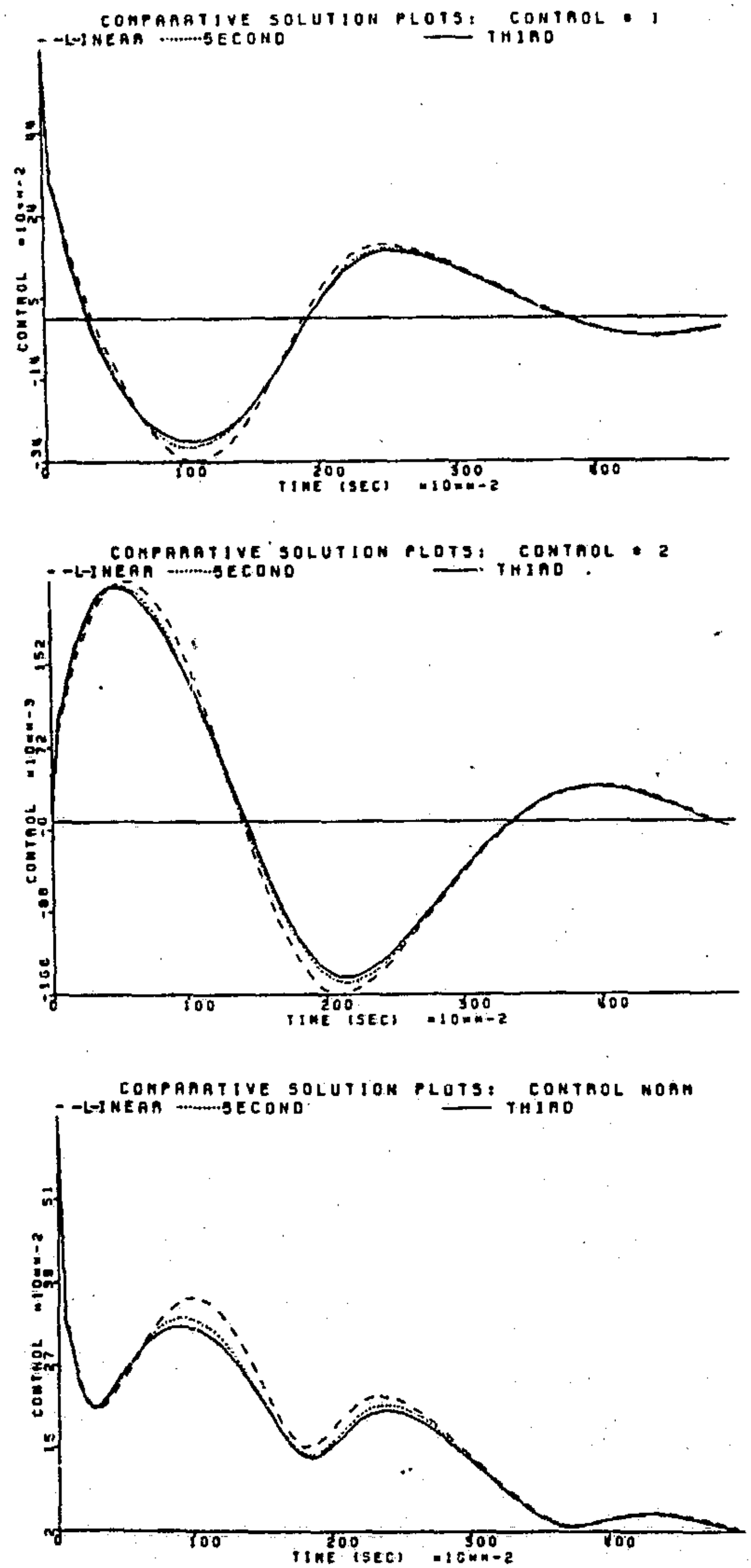


Figure B.24 Control plots for x(0) = (-1.,0.)

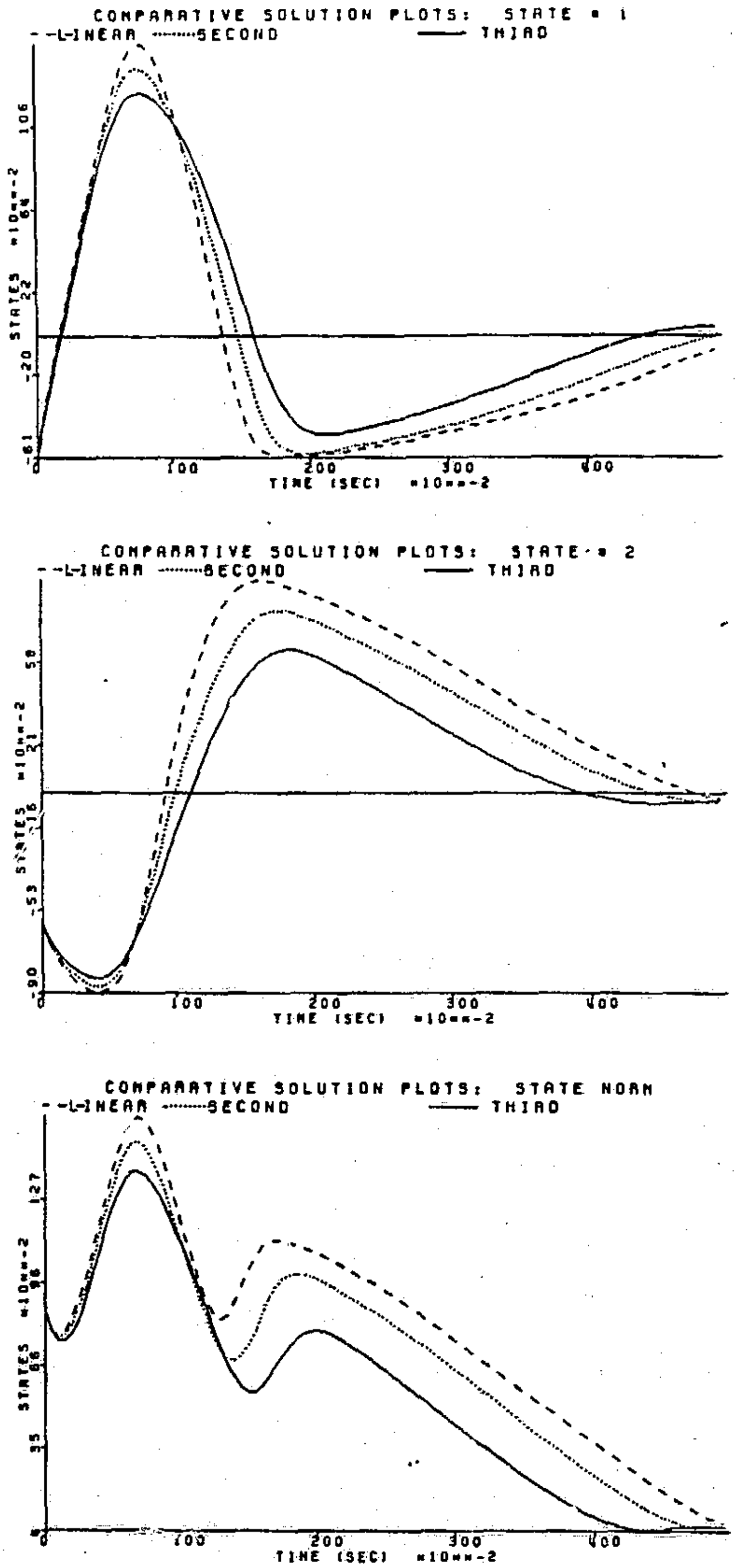


Figure B.25 State plots for x(0) = (-.6,-.6)

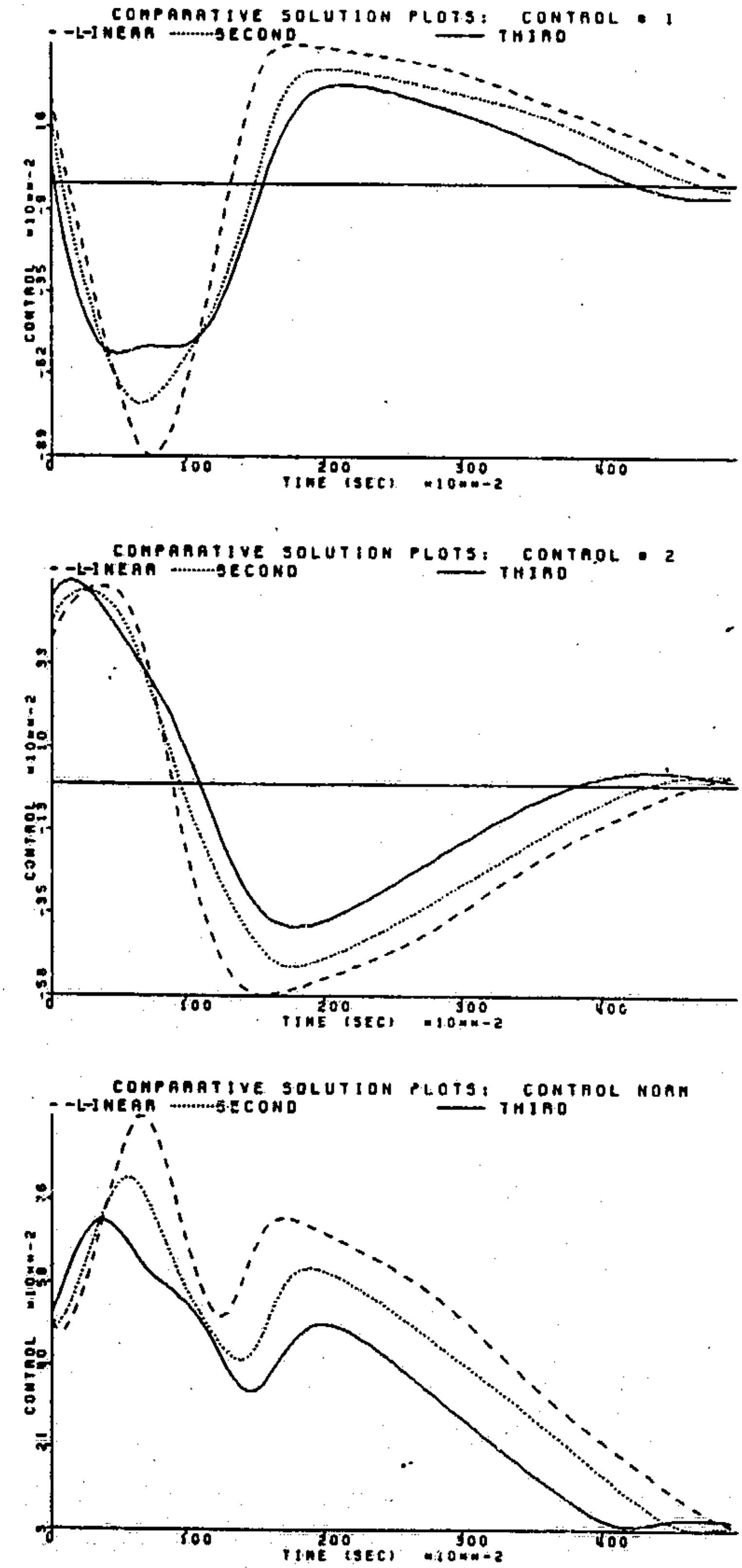


Figure B.26 Control plots for $x(0) = (-.6,-.6)$

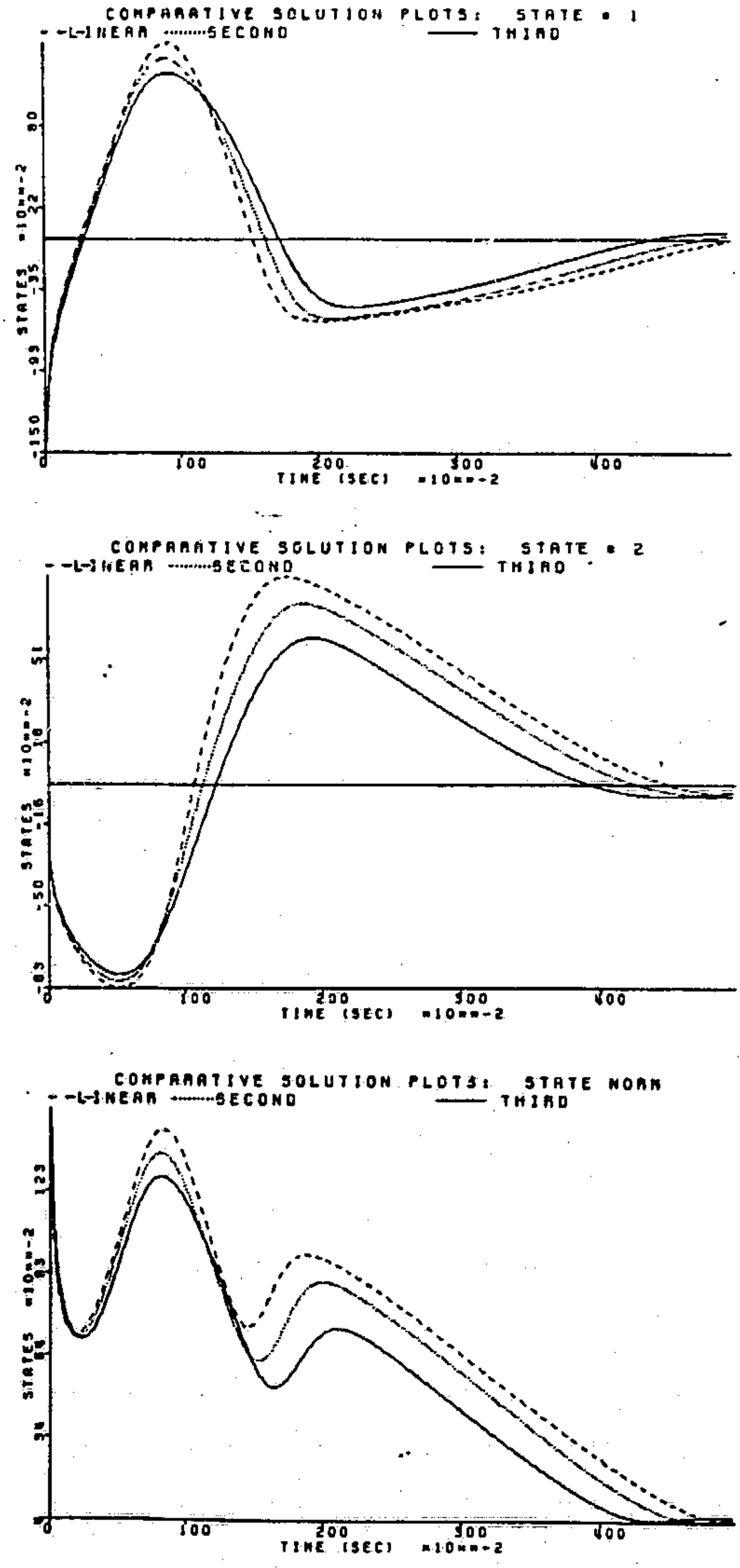


Figure B.27 State plots for x(0) = (-1.5,-0.3)

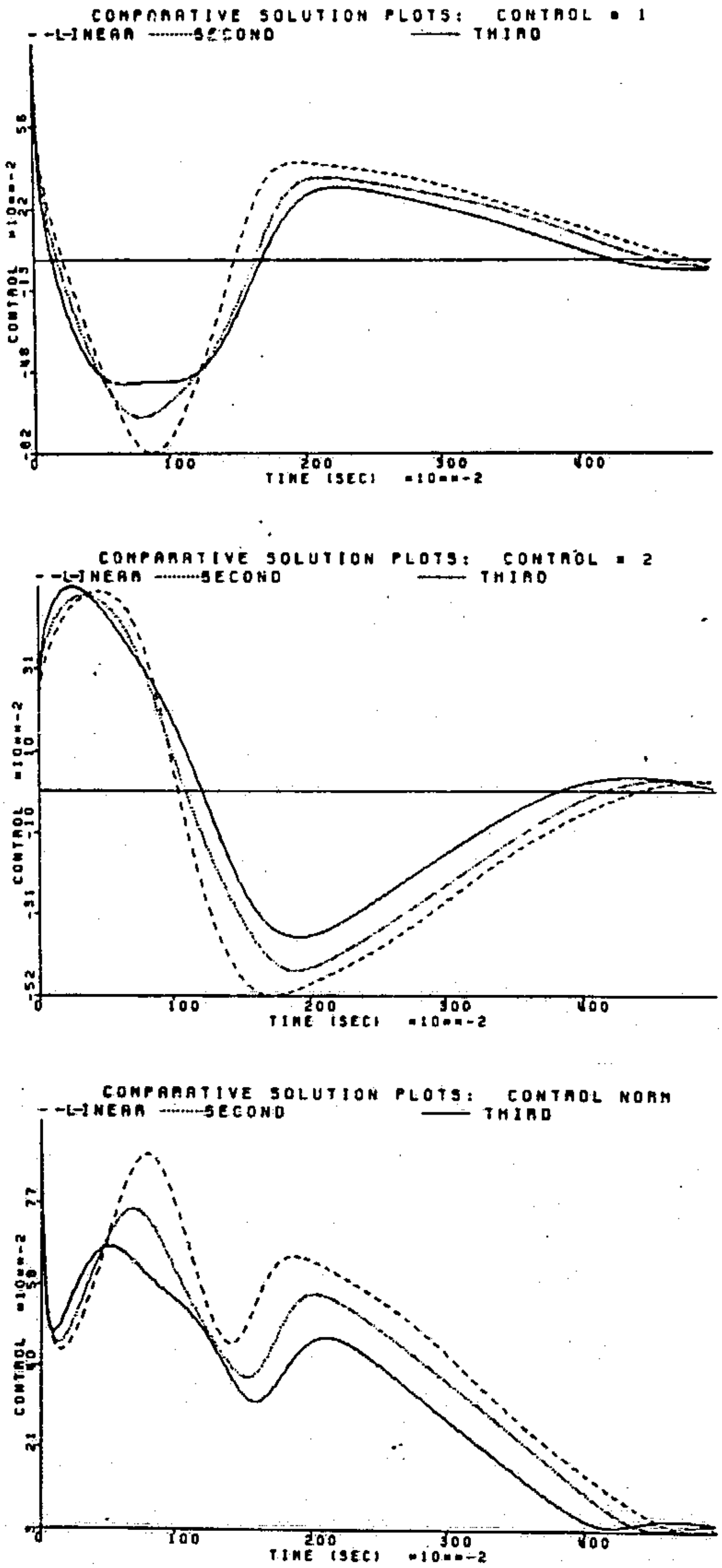


Figure B.28 Control plots for x(0) = (-1.5,-0.3)

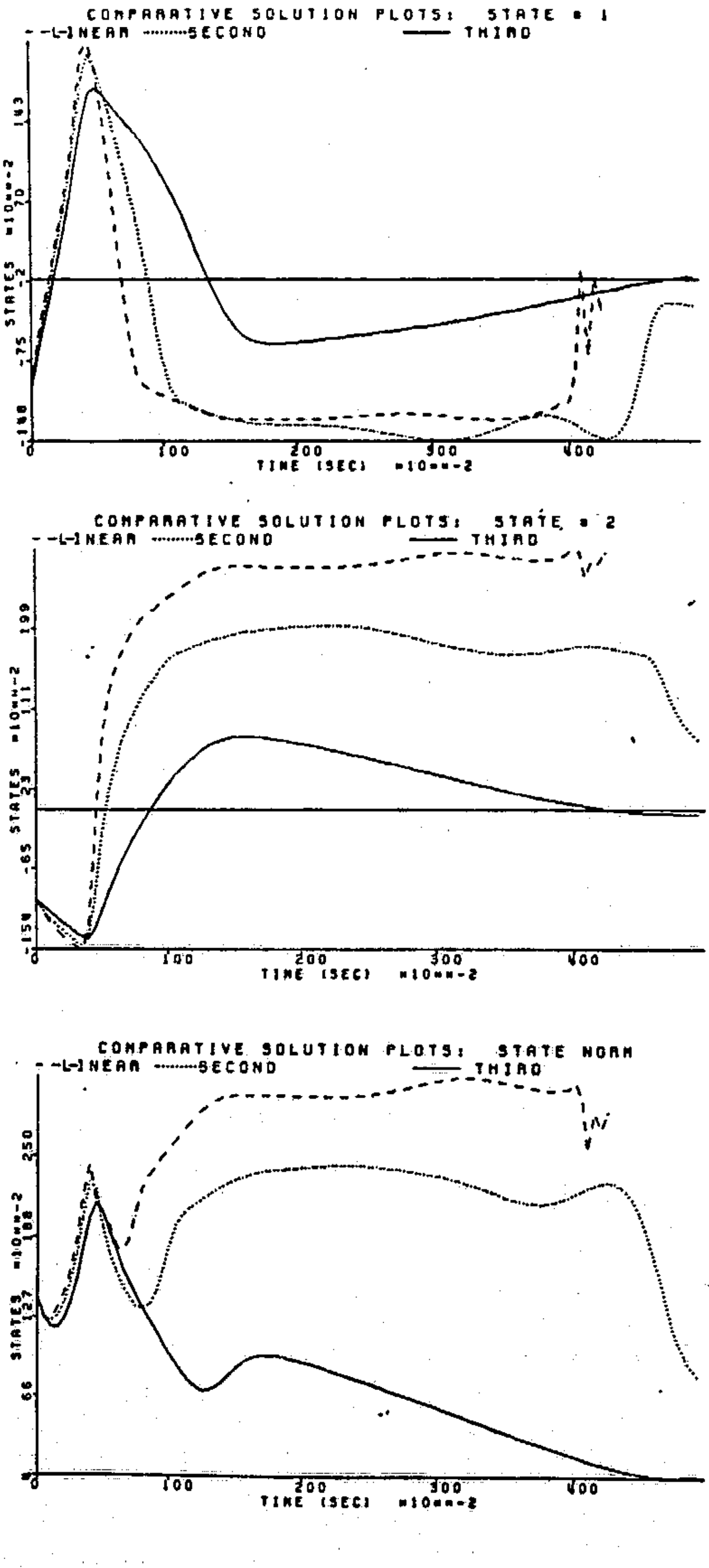


Figure B.29 State plots for x(0) = (-1.,-1.)
(Linear goes unstable)

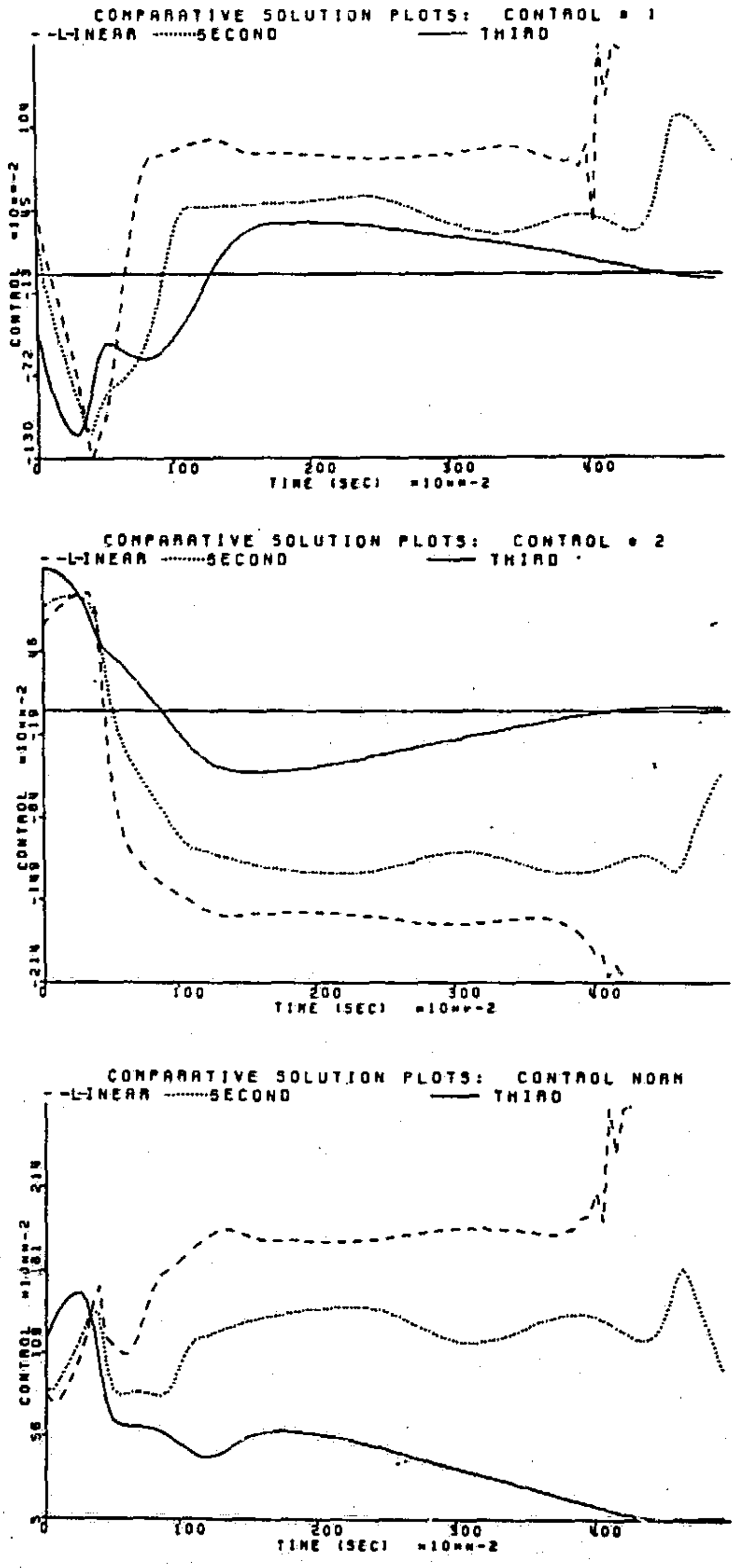


Figure B.30 Control plots for x(0) = (-1.,-1.)
(Linear goes unstable)

ORIGINAL PAGE IS
OF POOR QUALITY

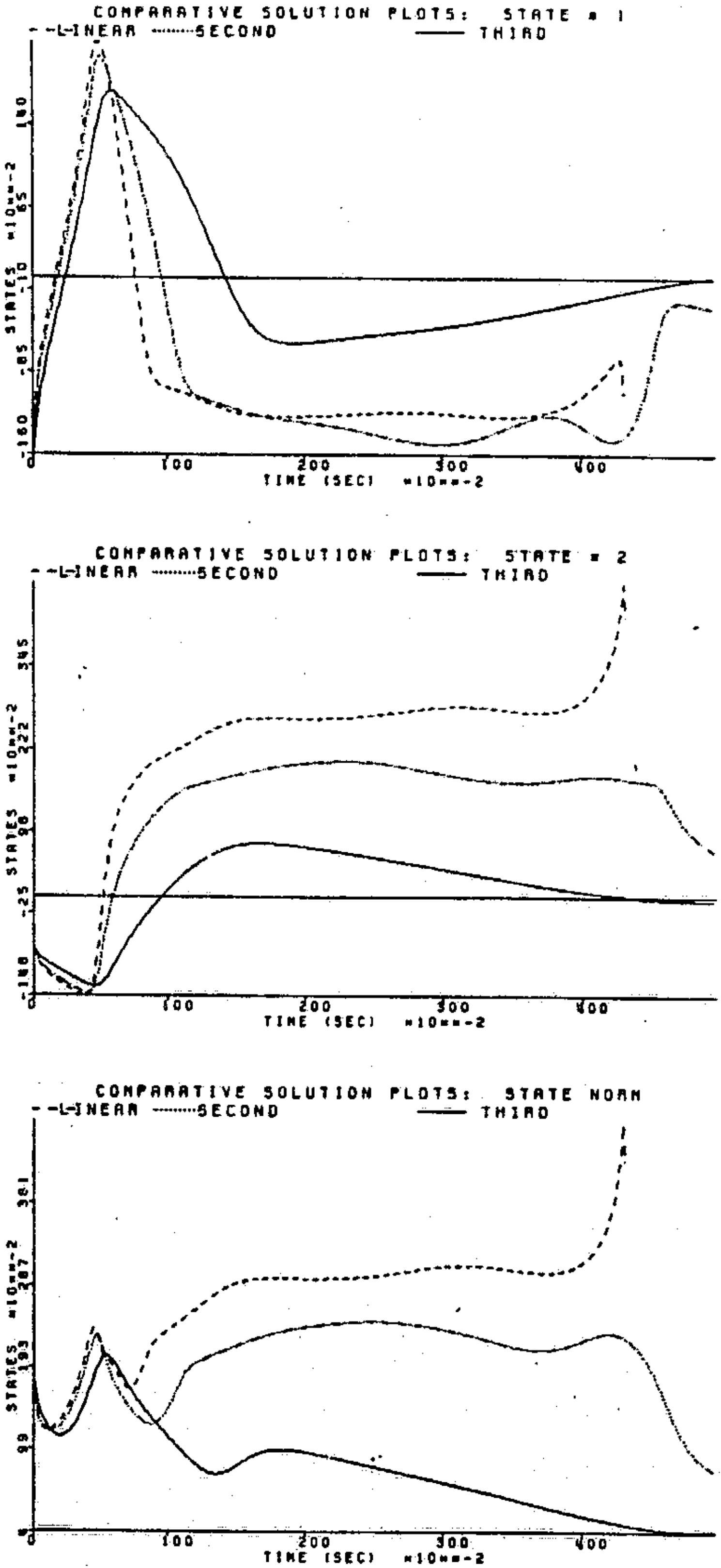


Figure B.31 State plots for x(0) = (-1.6,-0.8)
(Linear goes unstable)

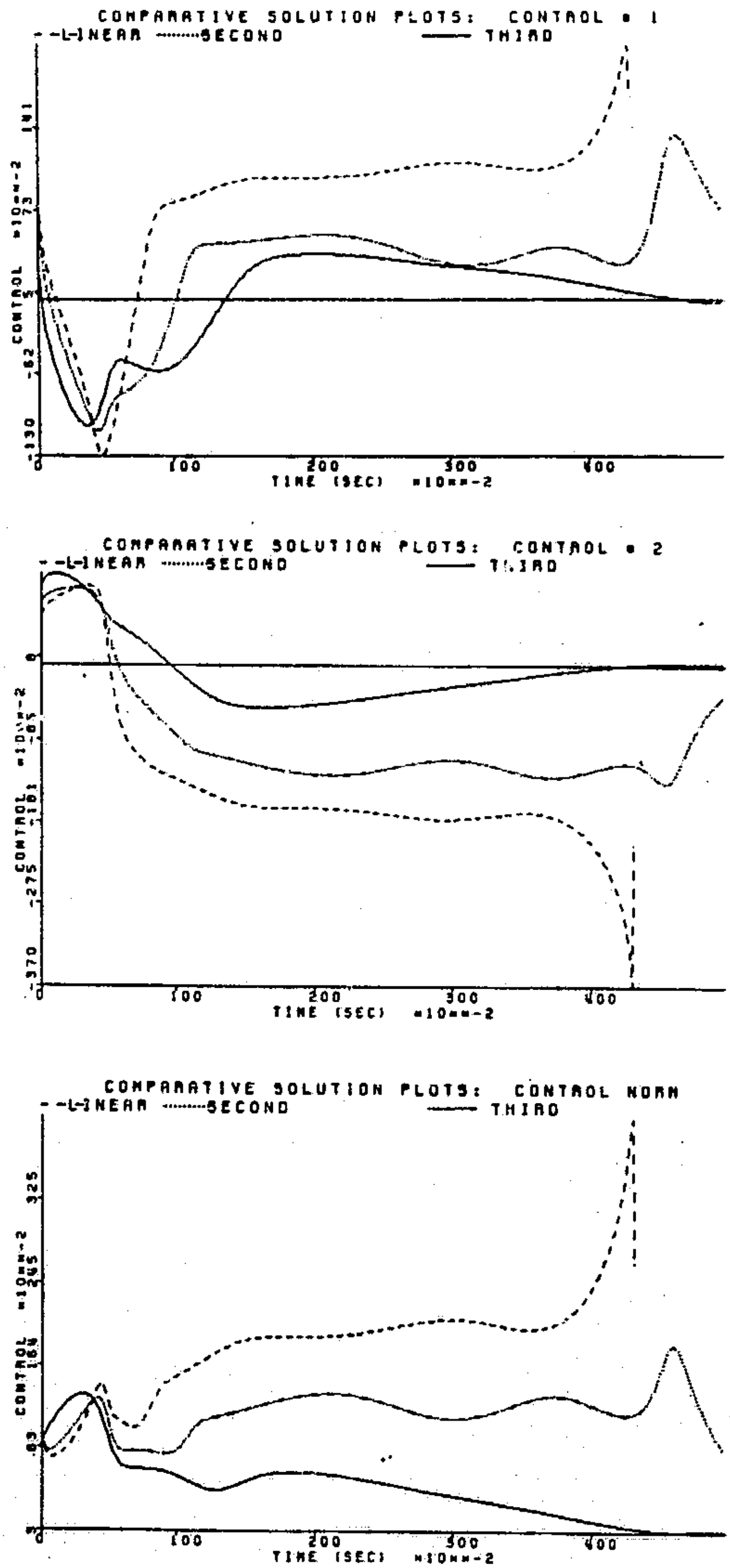


Figure B.32 Control plots for x(0) = (-1.6,-0.8)
(Linear goes unstable)

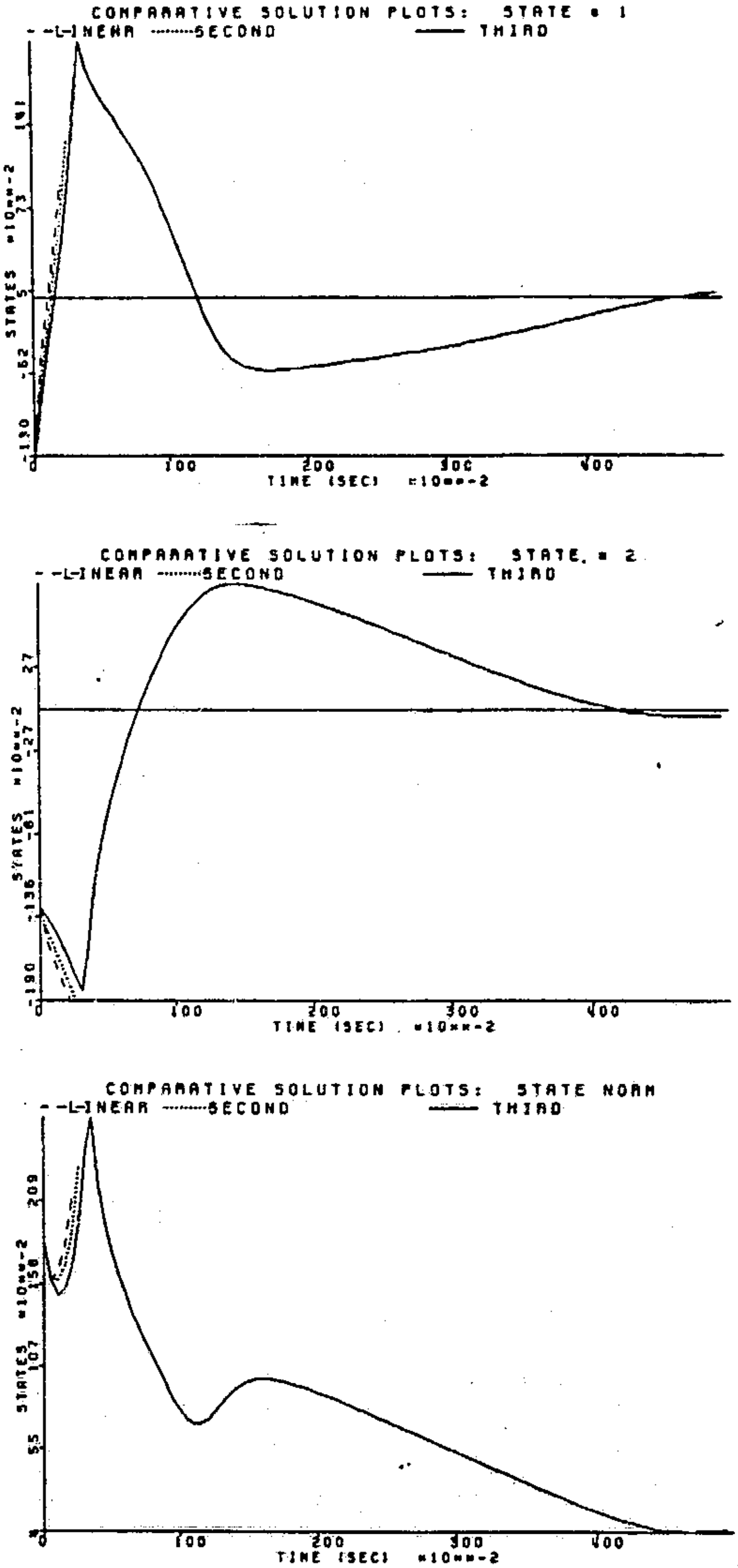


Figure B.33 State plots for $x(0) = (-1.3,-1.3)$
(Linear and second order go unstable)

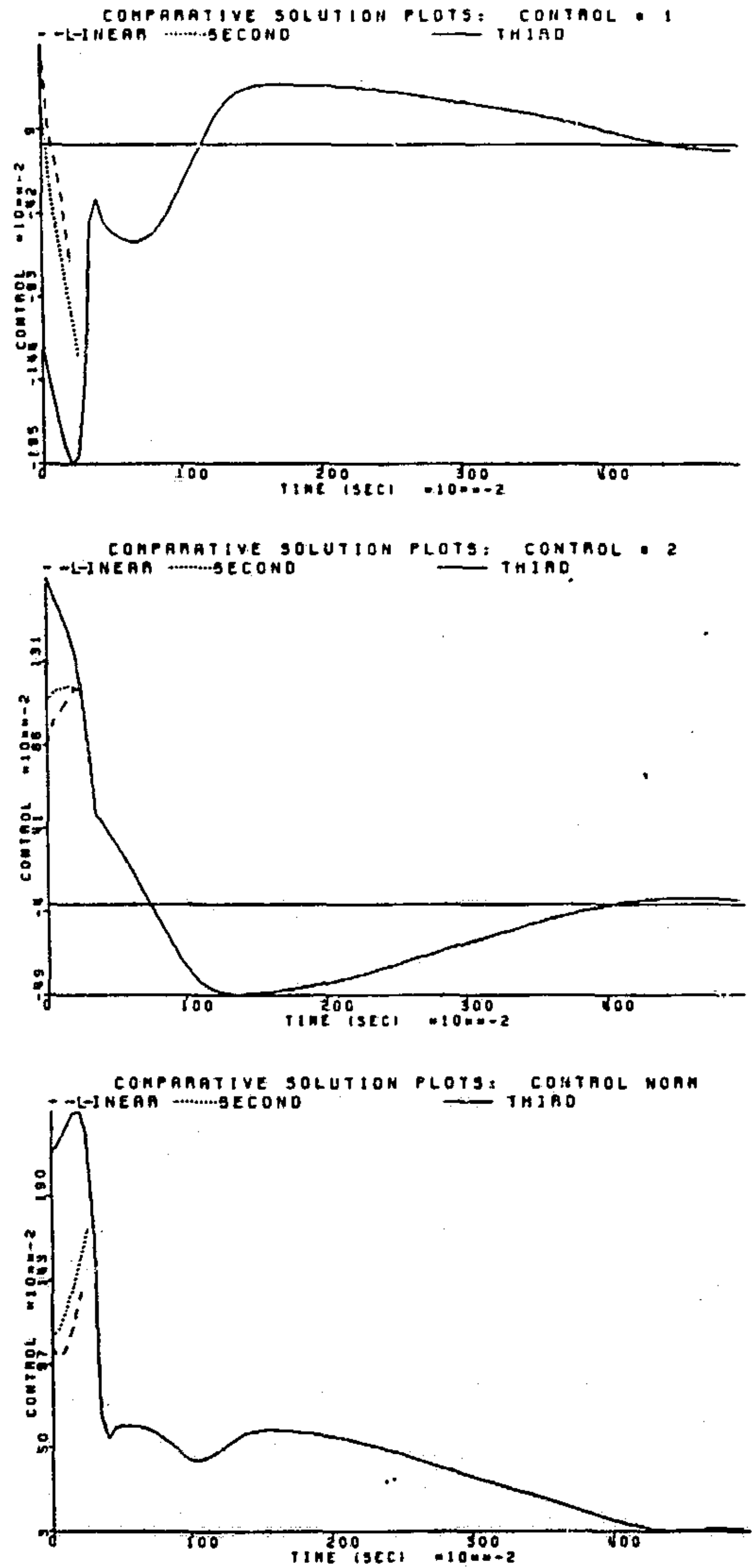


Figure B.34 Control plots for x(0) = (-1.3,-1.3)
(Linear and second order go unstable)

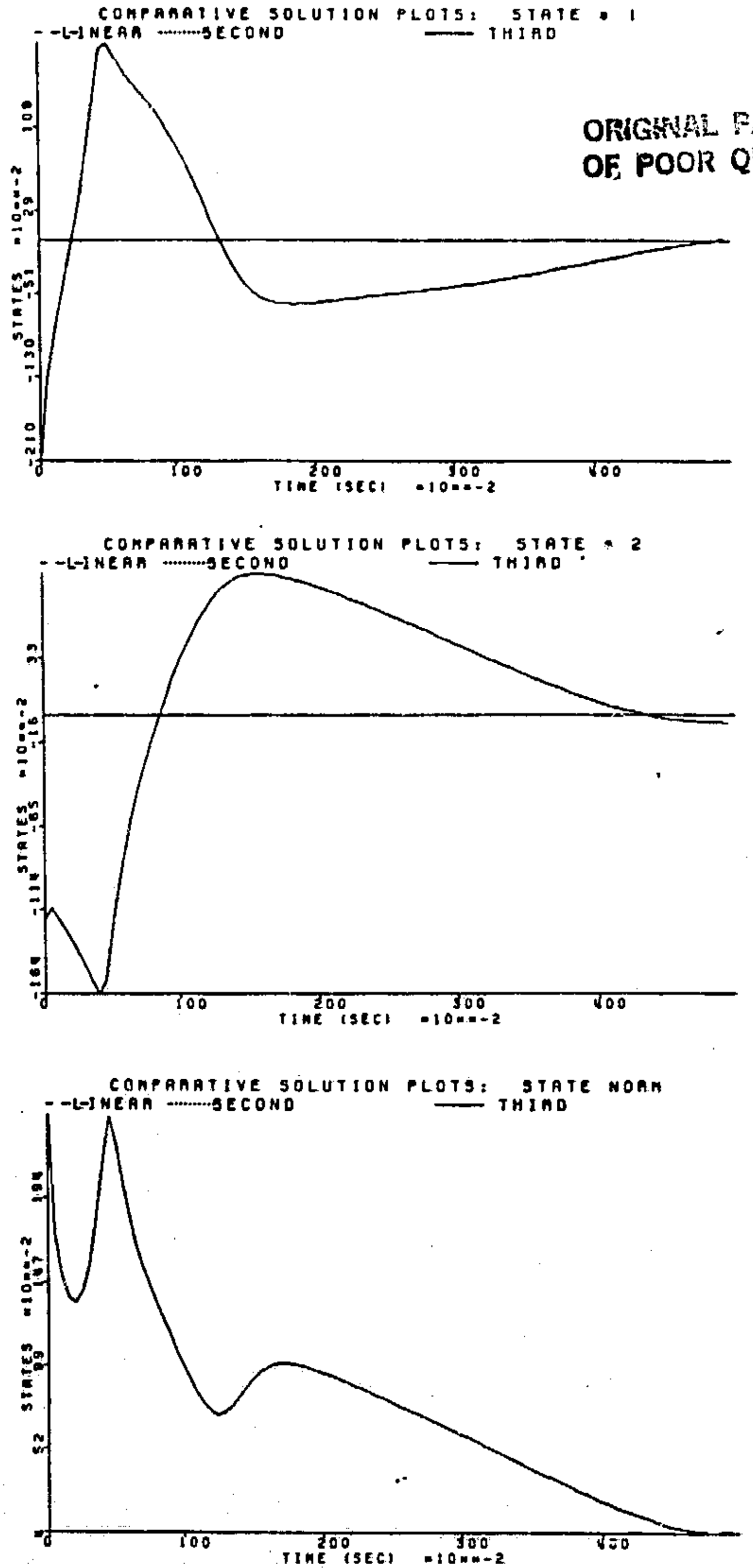


Figure B.35 State plots for x(0) = (-2.1,-1.2)
(Linear and second order are unstable immediately)

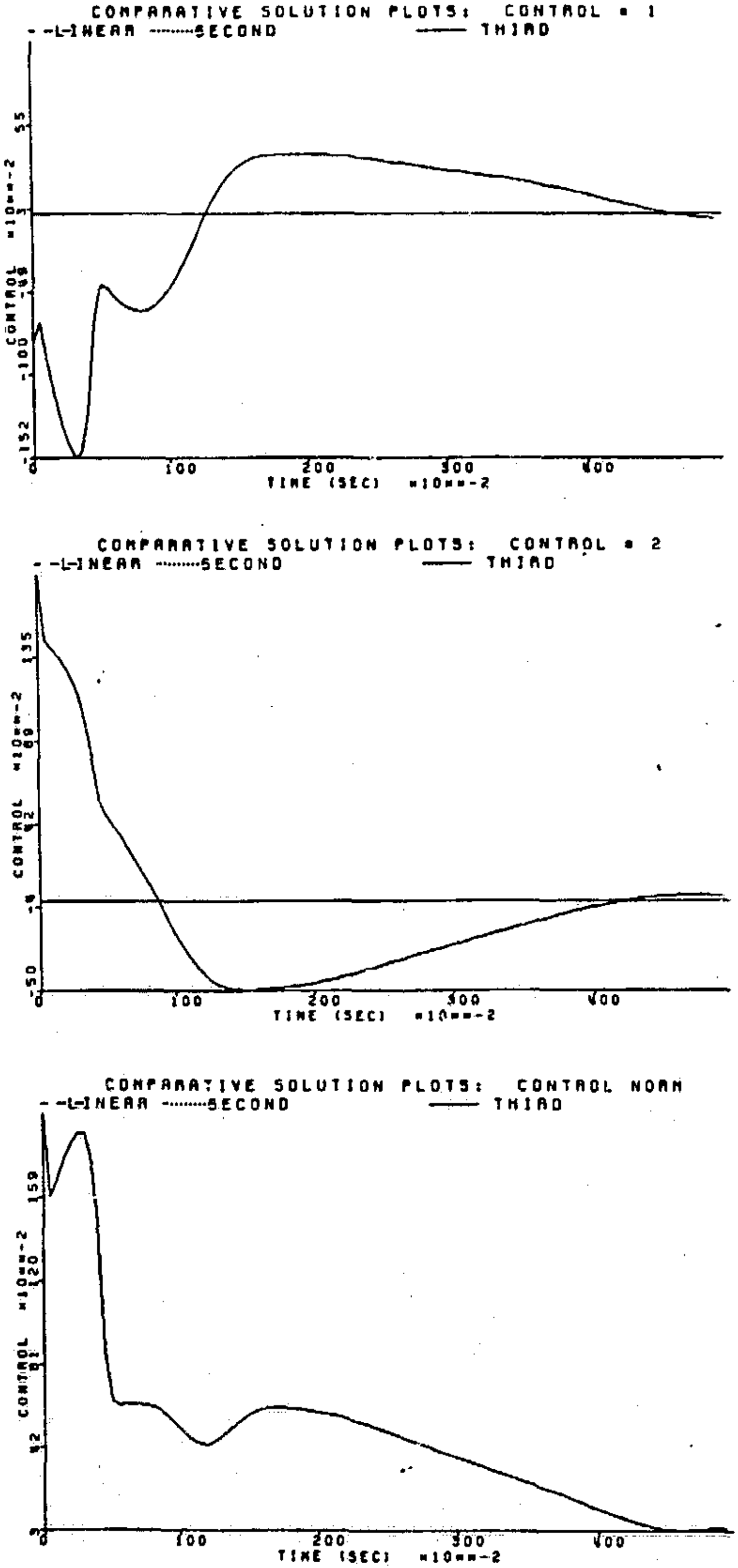


Figure B.36 Control plots for x(0) = (-2.1,-1.2) (Linear and second order are unstable immediately)

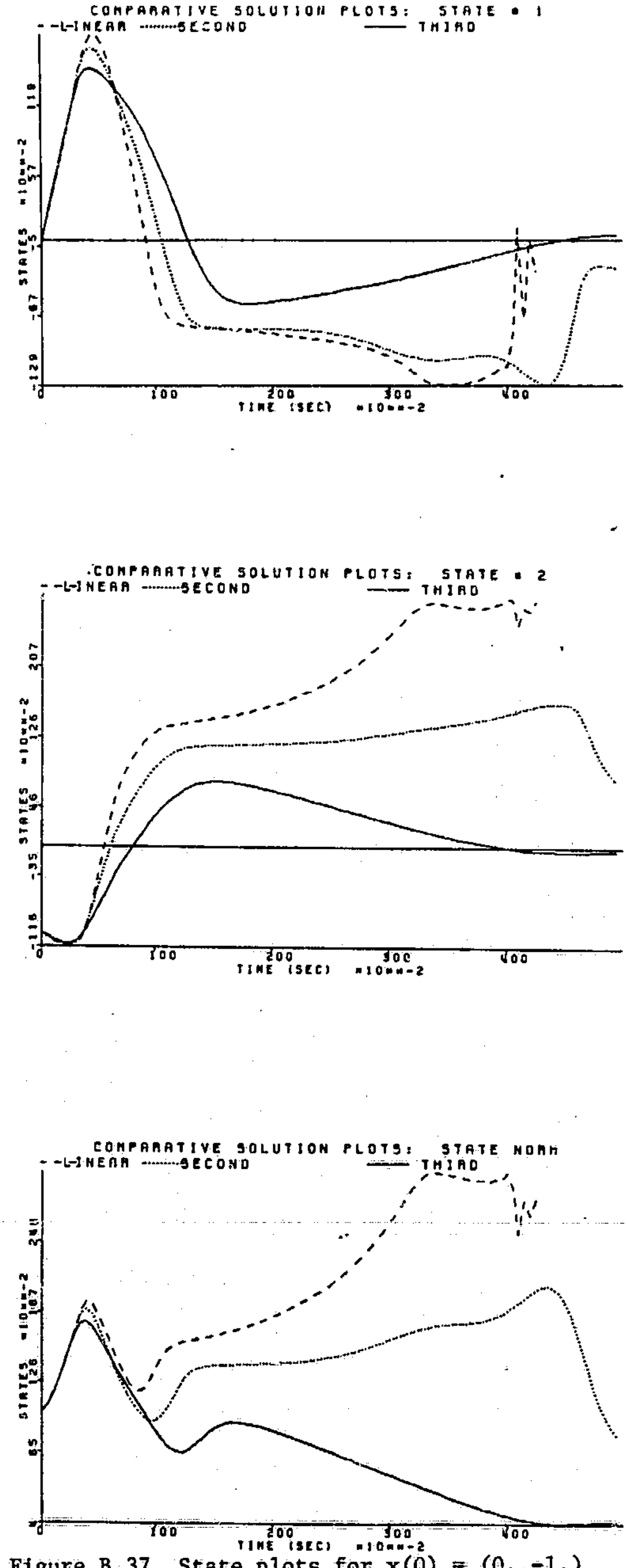


Figure B.37 State plots for x(0) = (0.,-1.)
(Linear goes unstable)

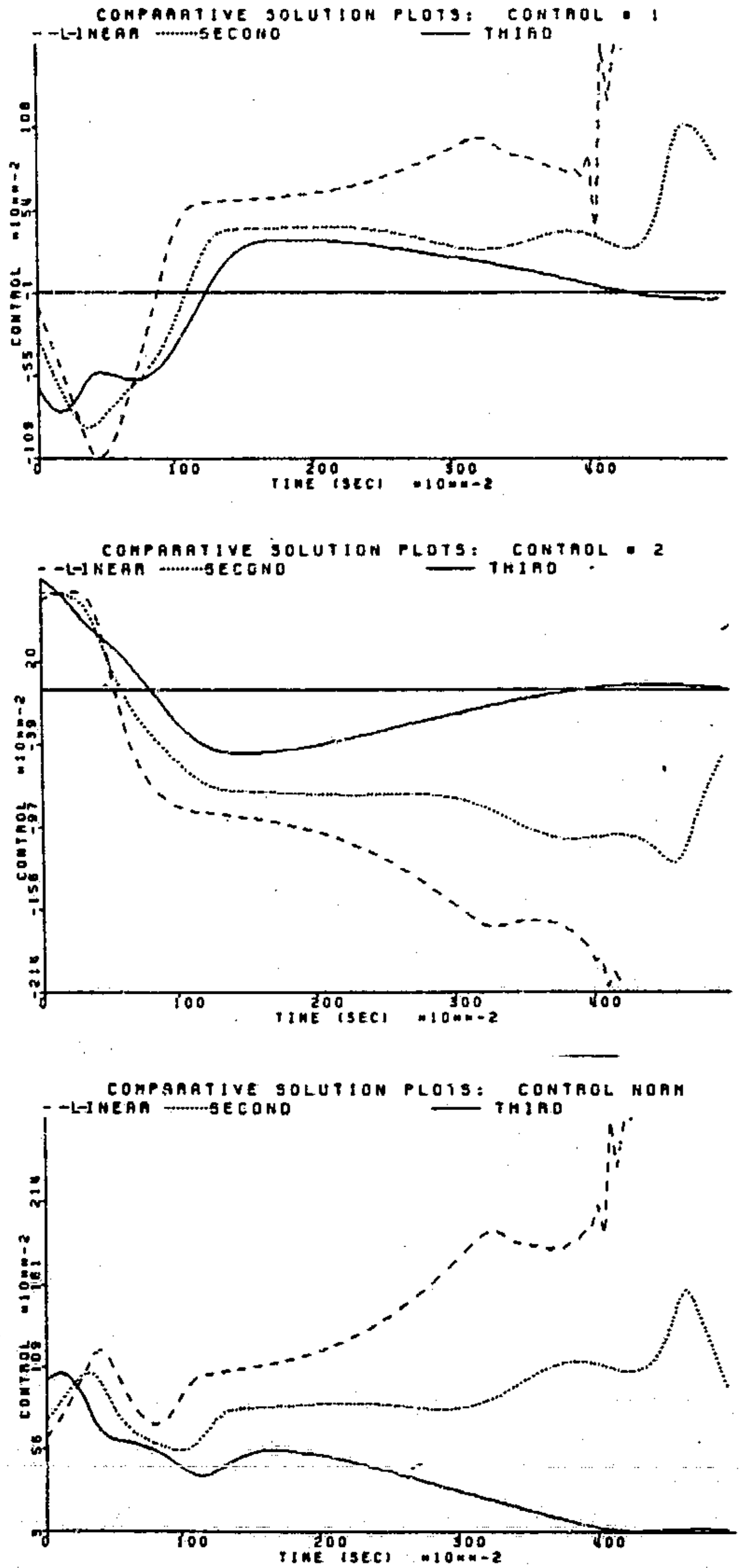


Figure B.38 Control plots for x(0) = (0.,-1.)
(Linear goes unstable)

and B.40 show the plots for $x(0) = (1.05,-1.05)$. Here the linear plot is starting to lose ground, its final state norm being about 0.5. Figure B.41 shows three plots for $x(0) = (1.1,-1.1)$ and it is seen that the linear feedback system is unstable. Thus the boundary is again very significant. Another point to note in Figure B.40 is that the antinode in the first control is evident in this quadrant also.

Figures B.42 through B.45 show the significance of the boundary of the second order region. Note that in B.42 and B.43 ($x(0) = (2.1,-.15)$) the second order feedback system still seems to behave satisfactorily while a slight perturbation to $x(0) = (2.15,-1.55)$ drives it unstable (Figures B.44 and B.45).

Finally, Figures B.46 through B.49 demonstrate the significance of the boundary of the third order region. In Figures B.46 and B.47 ($x(0) = (2.3,-1.5)$, inside the region in Figure 5.7), the third order controller is still accomplishing its goal. The state trajectories and control trajectories all go toward zero. The plots of the state norm and control norm in Figure B.48 for $x(0) = (2.375,-1.5)$ show that the third order is still effective. Figure B.49 shows that when $x(0) = (2.376,-1.5)$ the third order feedback system goes unstable. Thus again the boundary of the region is seen to be very significant.

In summary, this appendix has verified the regions shown in Figures 5.5 through 5.7. The regions get larger as more feedback terms are added. Furthermore, there is an increase in performance as terms are added throughout the third and fourth quadrants. In the first quadrant and part of the second, the linear is slightly better than the second where both are useful although

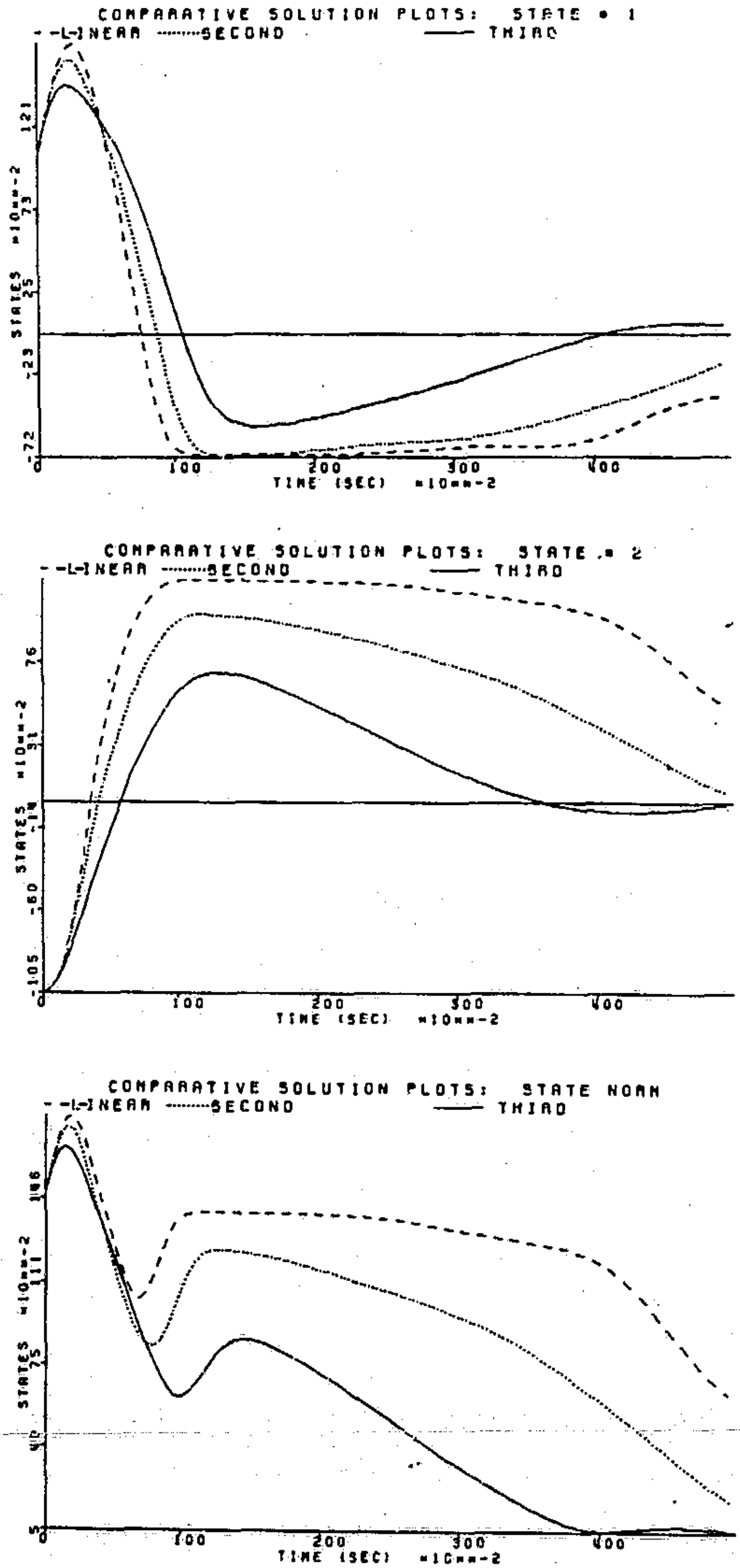


Figure B.39 State plots for x(0) = (1.05,-1.05)

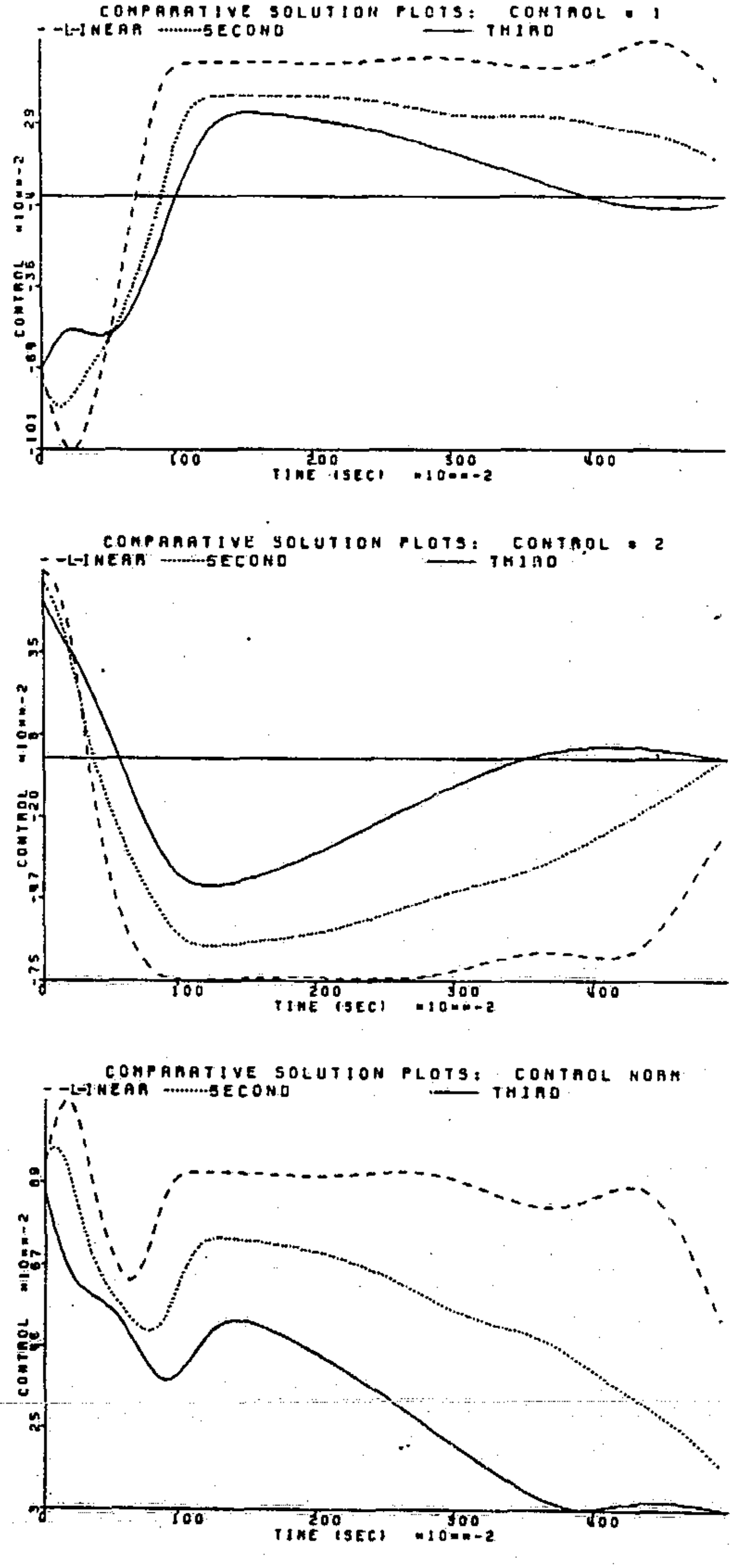


Figure B.40 Control plots for x(0) = (1.05,-1.05)

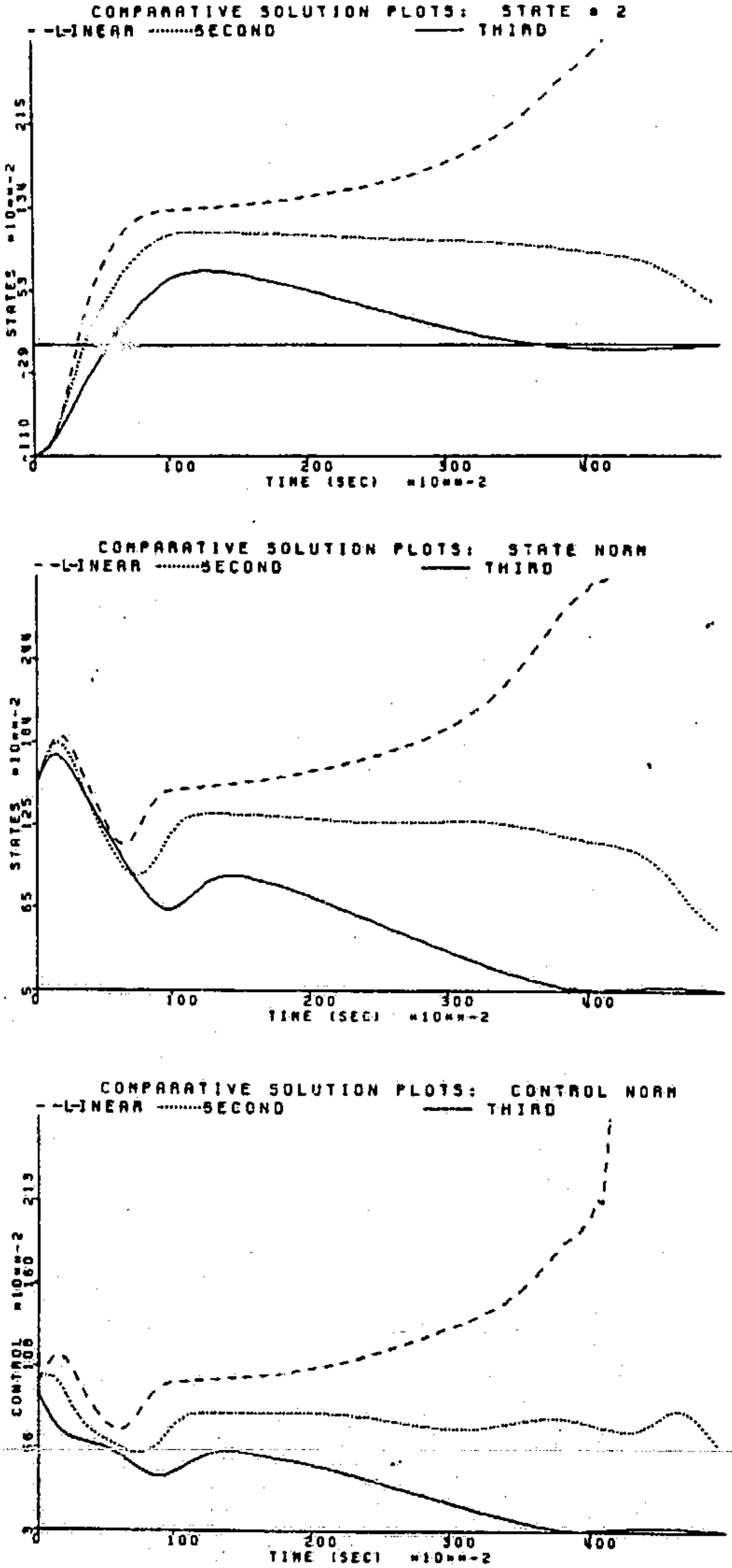


Figure B.41 Selected plots for x(0) = (1.1,-1.1)
(Linear goes unstable)

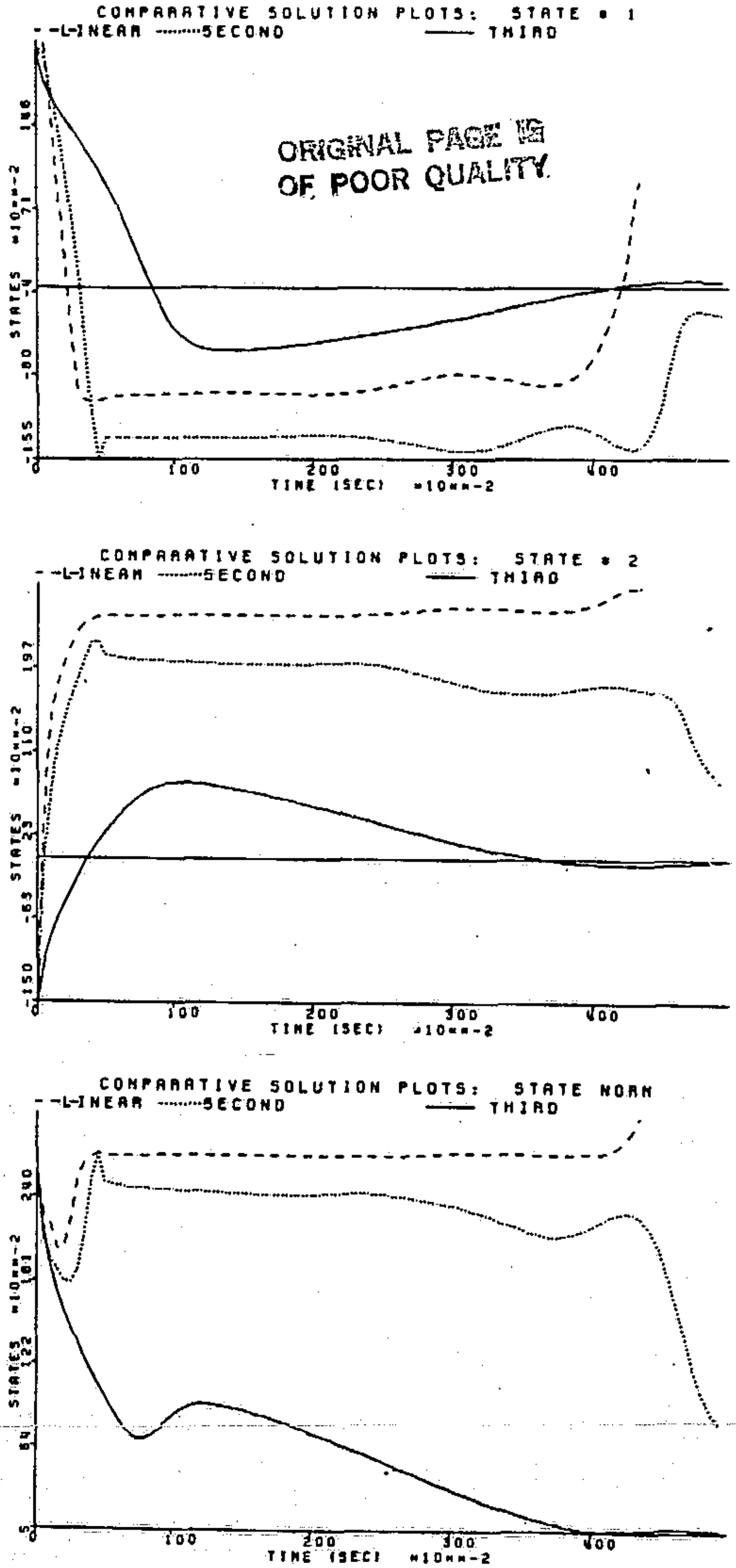


Figure B.42 State plots for x(0) = (2.1,-1.5)
(Linear goes unstable)

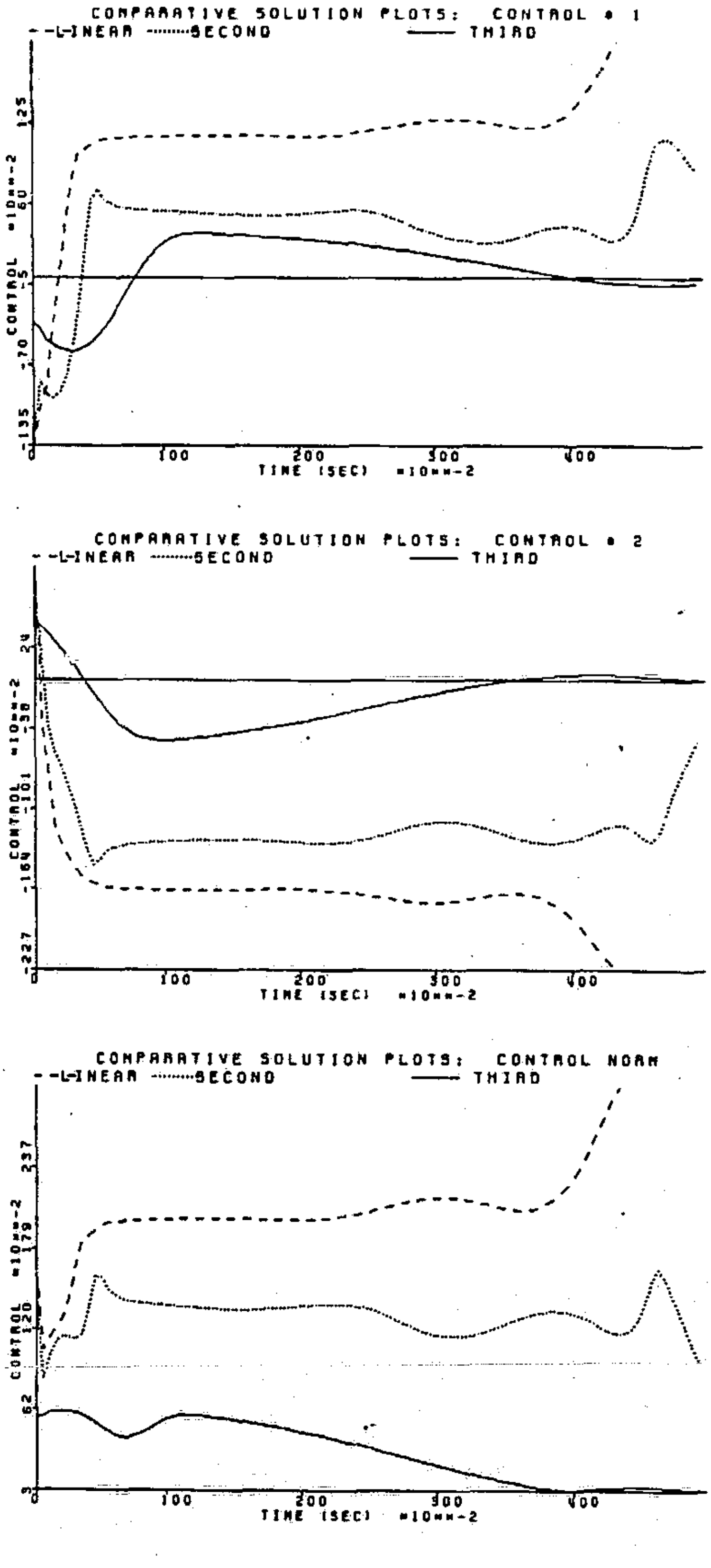


Figure B.43 Control plots for x(0) = (2.1,-1.5)
(Linear goes unstable)

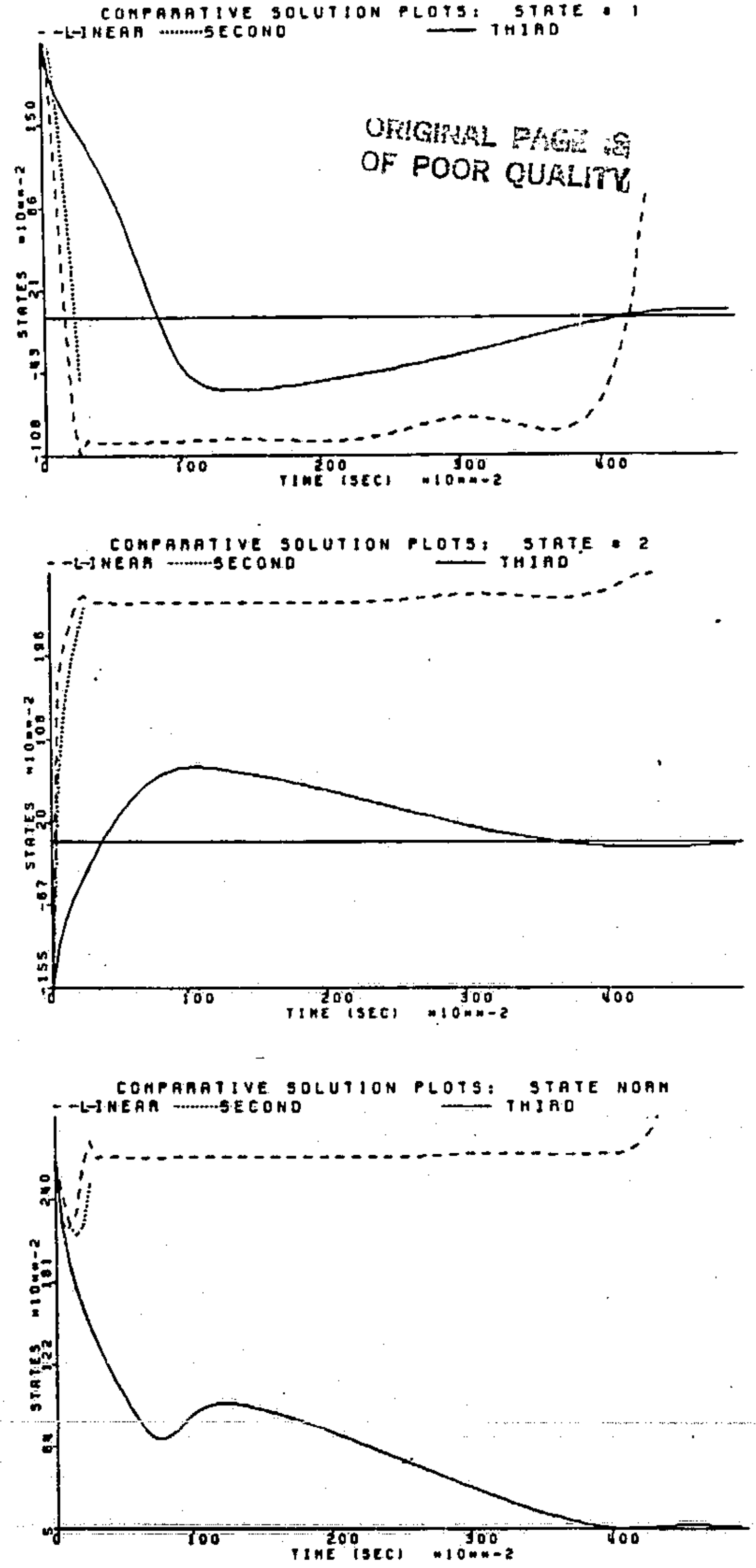


Figure B.44 State plots for $x(0) = (2.15,-1.55)$
(Linear and quadratic go unstable)

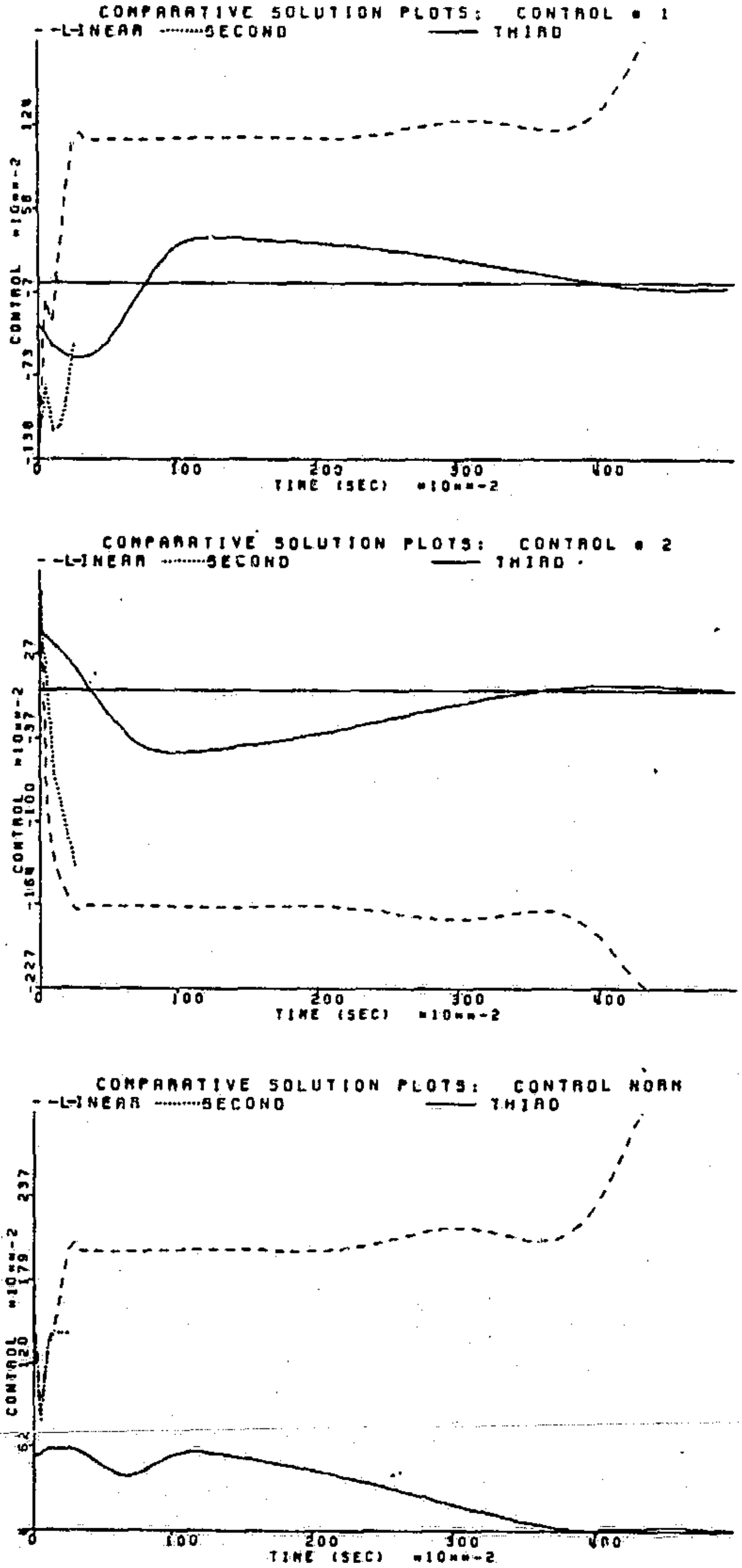


Figure B.45 Control plots for $x(0) = (2.15,-1.55)$
(Linear and quadratic go unstable)

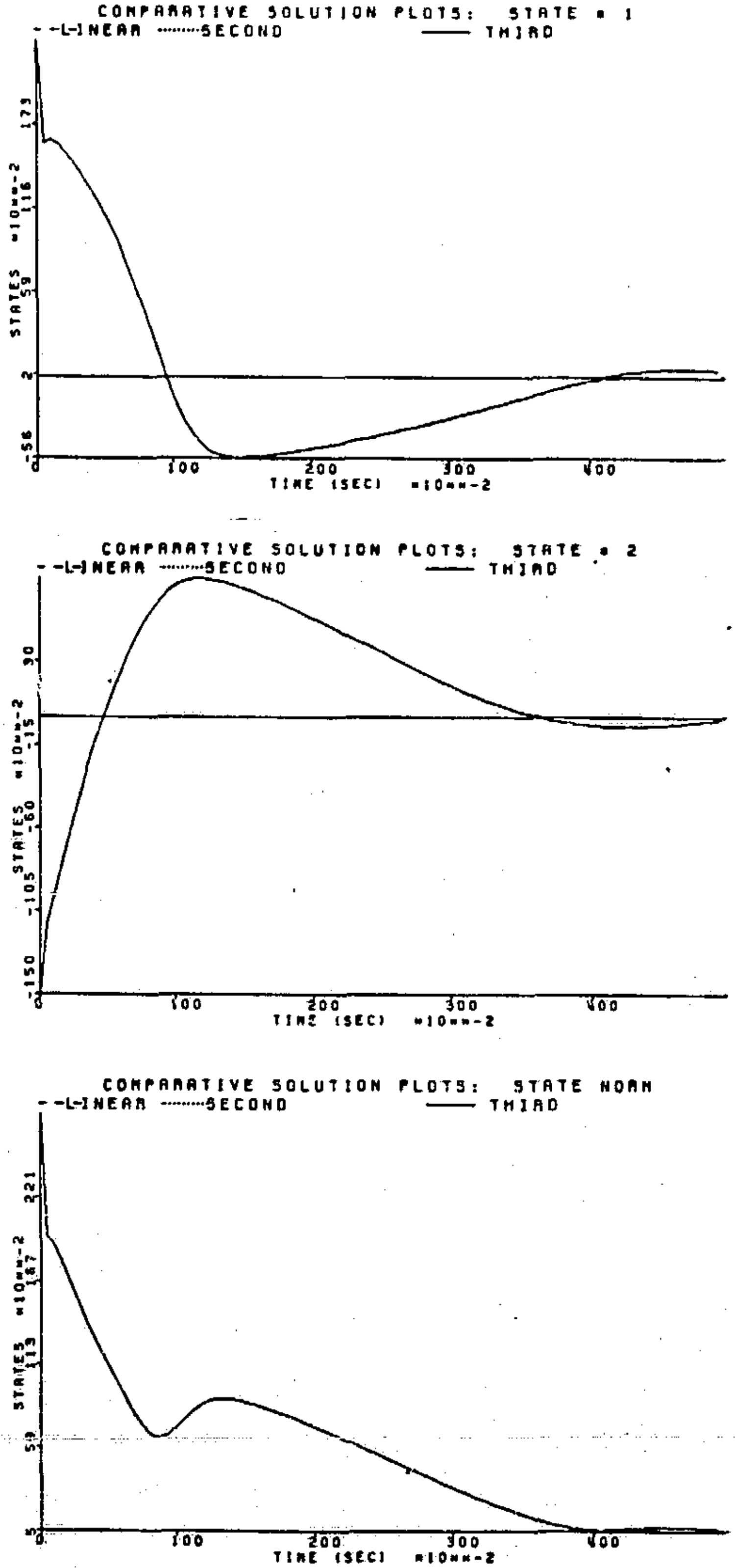


Figure B.46 State plots for x(0) = (2.3,-1.5) (Linear and quadratic are unstable immediately)

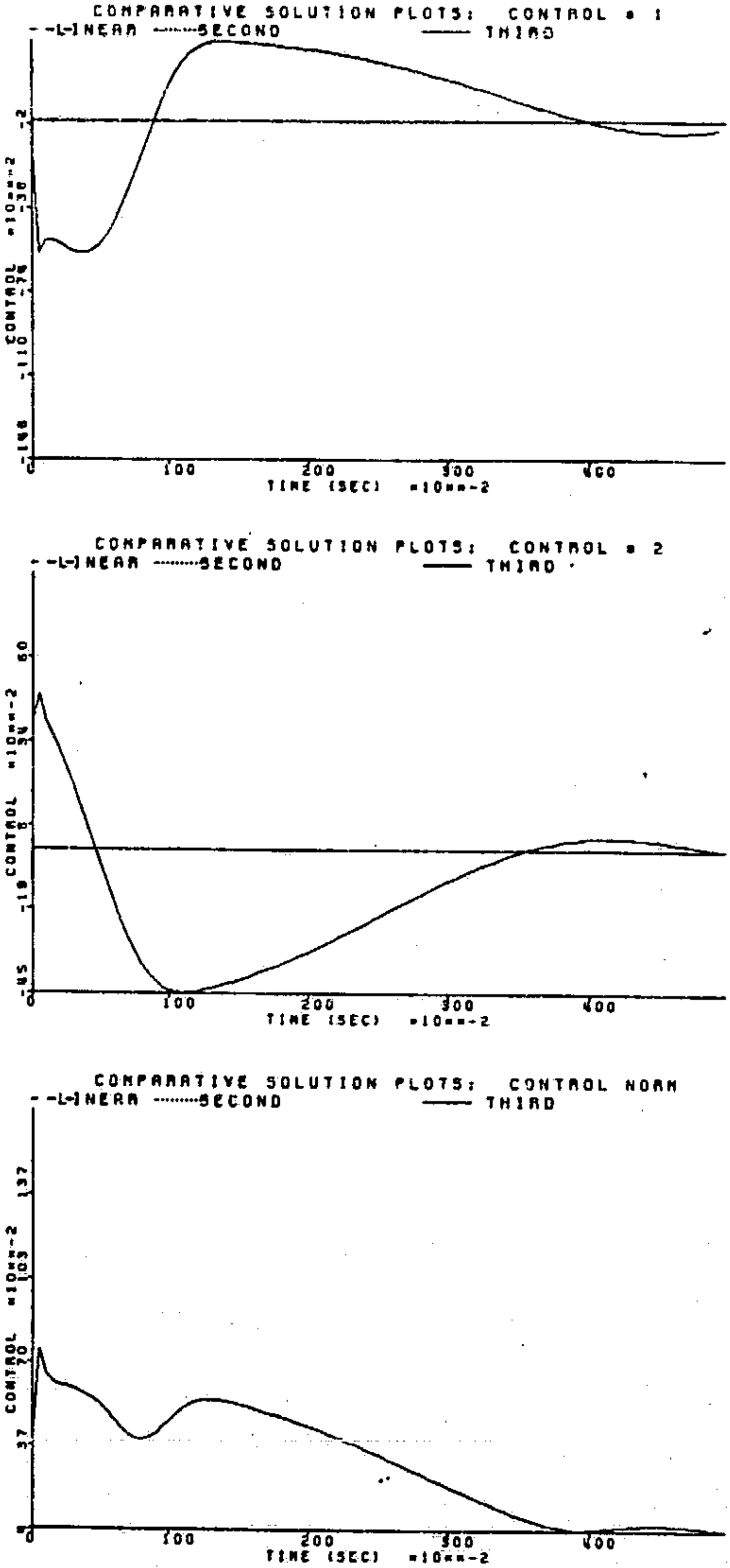
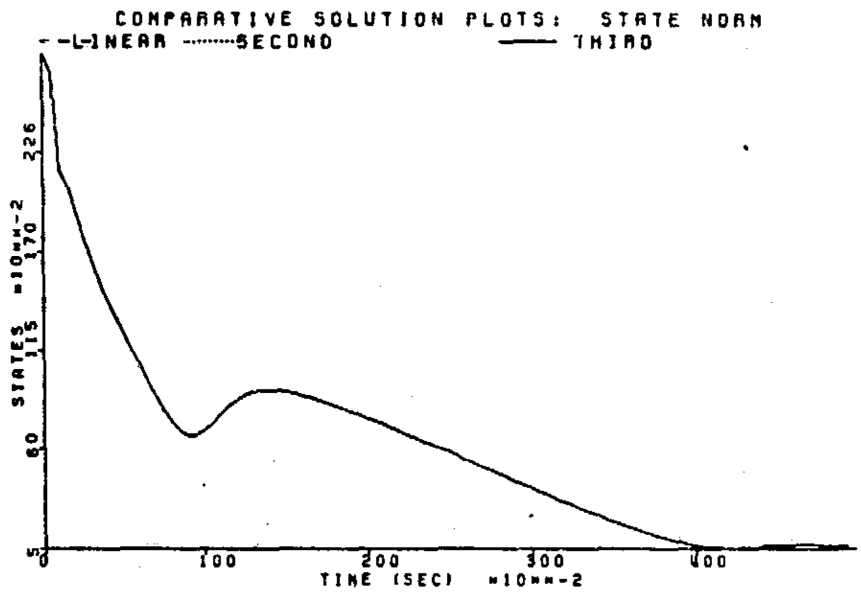


Figure B.47 Control plots for x(0) = (2.3,-1.5) (Linear and quadratic are unstable immediately)



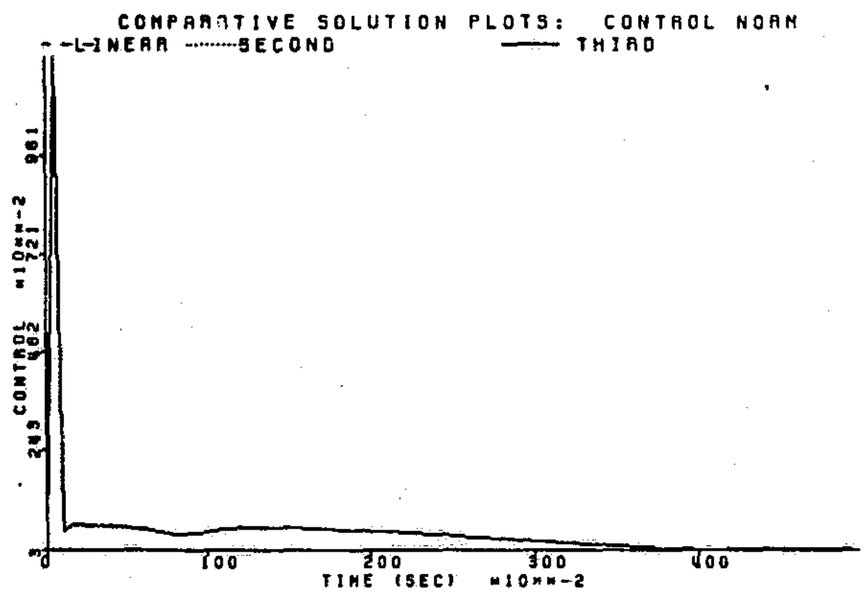
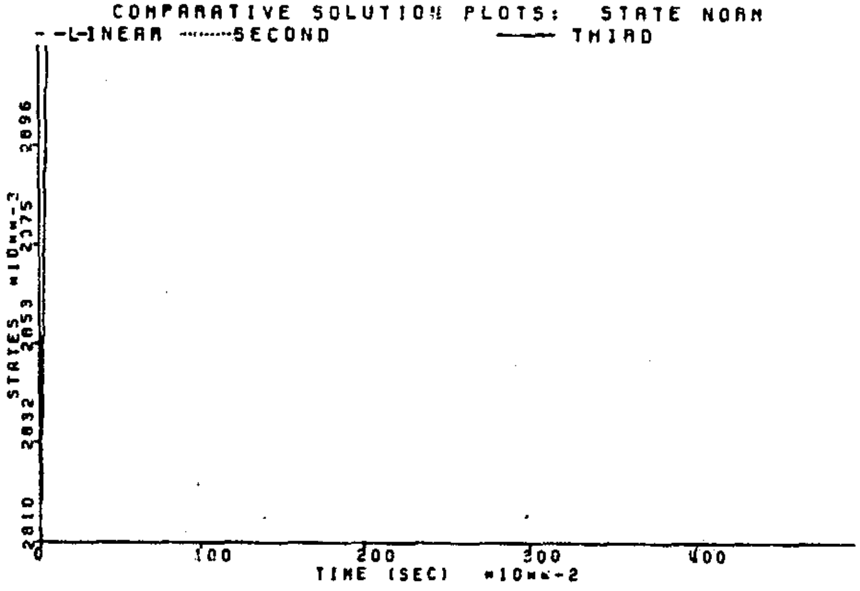


Figure B.48 Norm plots for x(0) = (2.375,-1.5)
(Linear and quadratic are unstable immediately)



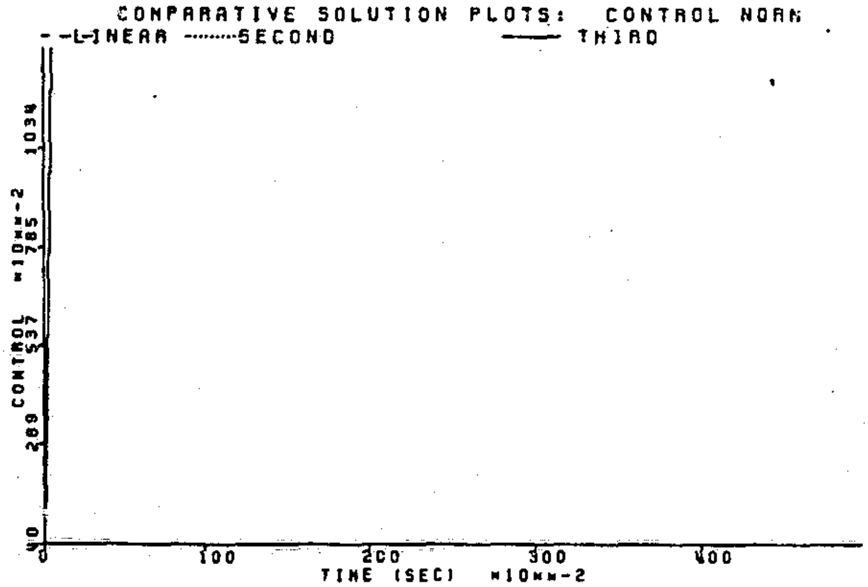


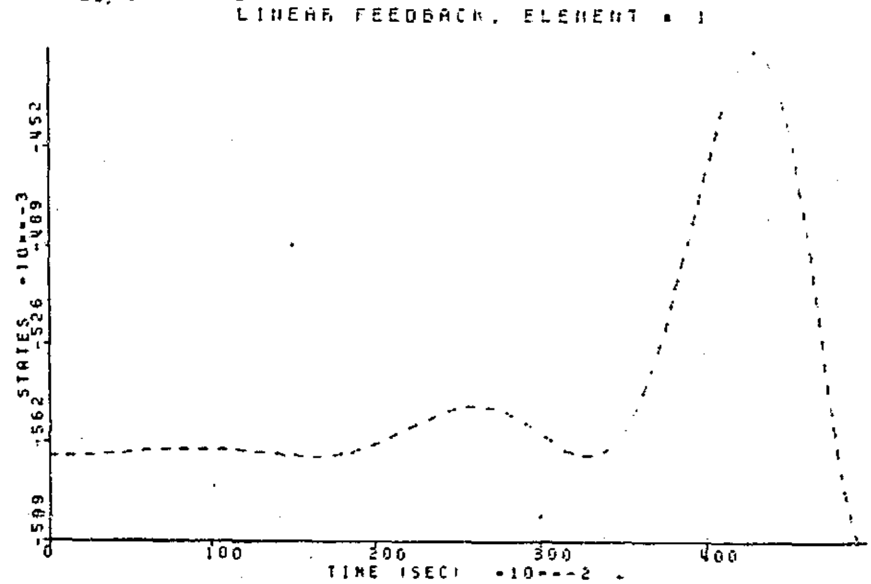
Figure 3.49 Norm plots for $x(0) = (2.376,-1.5)$
(All are unstable)

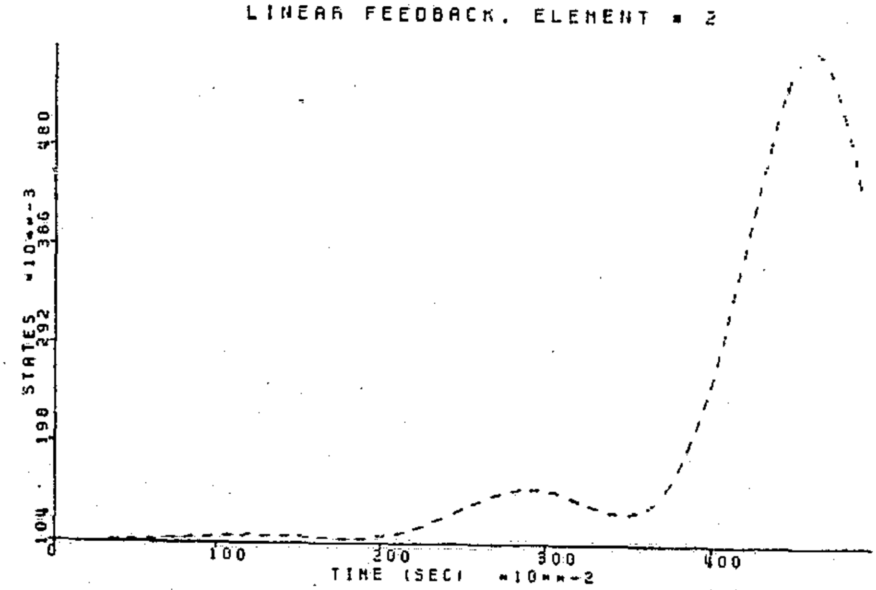
the second is useful over a much larger area in both quadrants. The third order performs best throughout most of the region. Also, a close look was taken at boundary points in the first and fourth quadrants and they were shown to be significant.

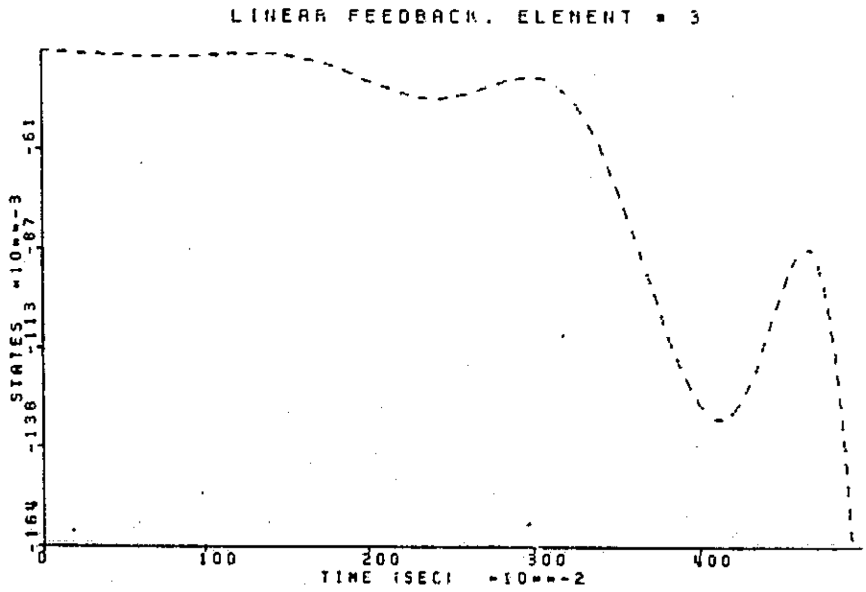
APPENDIX C

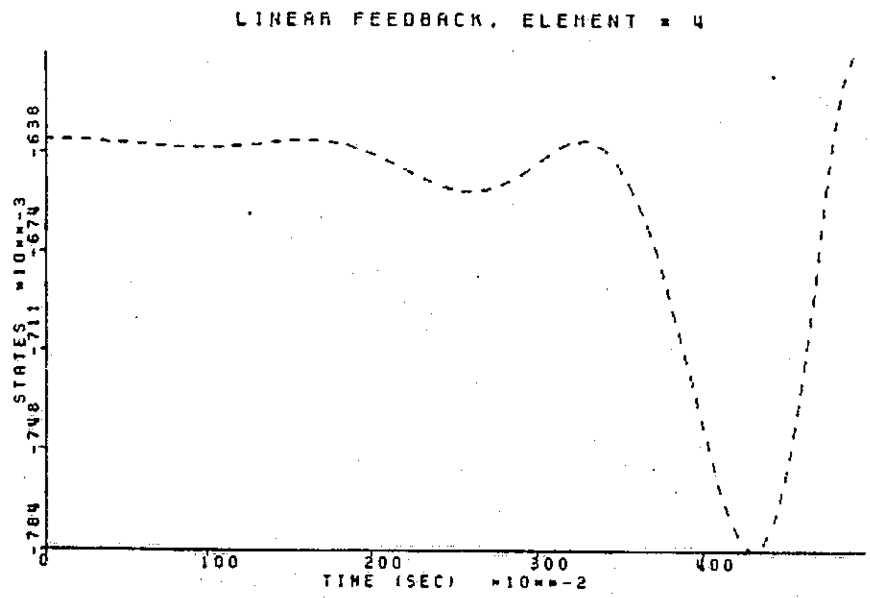
This appendix shows the data which is tabulated in Chapter 5, Tables 5.1 through 5.6 in graphical form.

ORIGINAL PAGE IS
OF POOR QUALITY

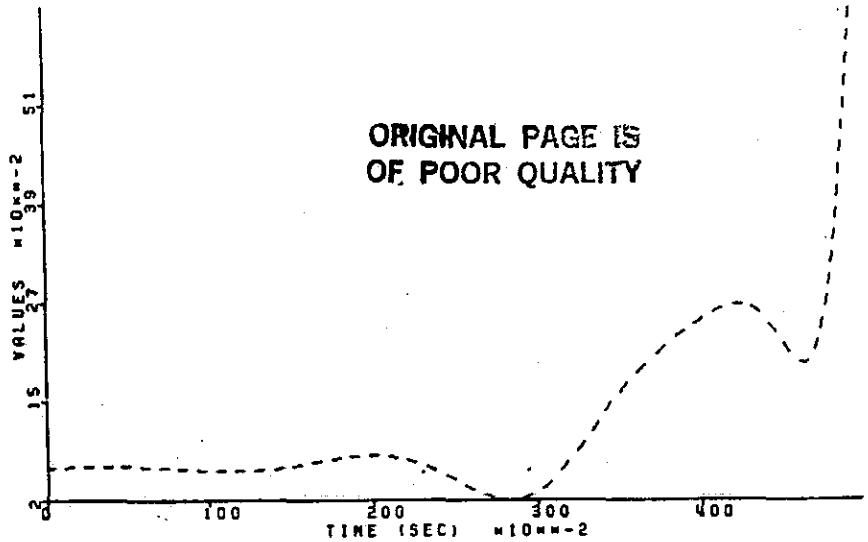




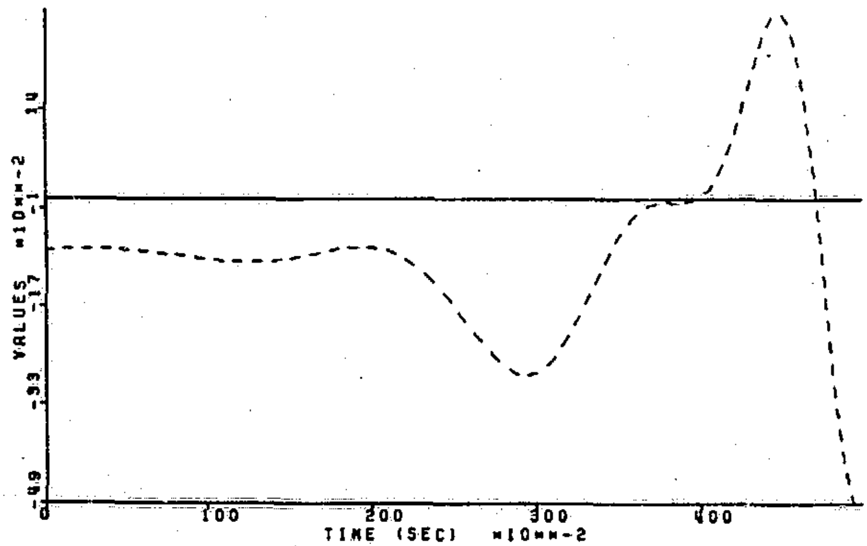

LINEAR FEEDBACK, ELEMENT = 3
STATES *10**-3
-61
-87
-113
-138
-164
0
100
200
300
400
TIME (SEC) *10**-2


LINEAR FEEDBACK, ELEMENT = 4
STATES *10**-3
-638
-674
-711
-748
-784
0
100
200
300
400
TIME (SEC) *10**-2

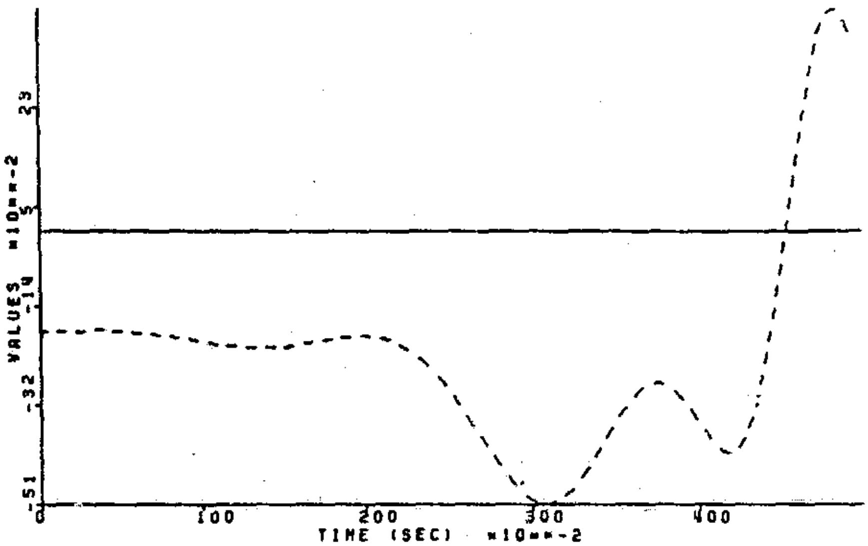
QUADRATIC FEEDBACK ELEMENT # 1



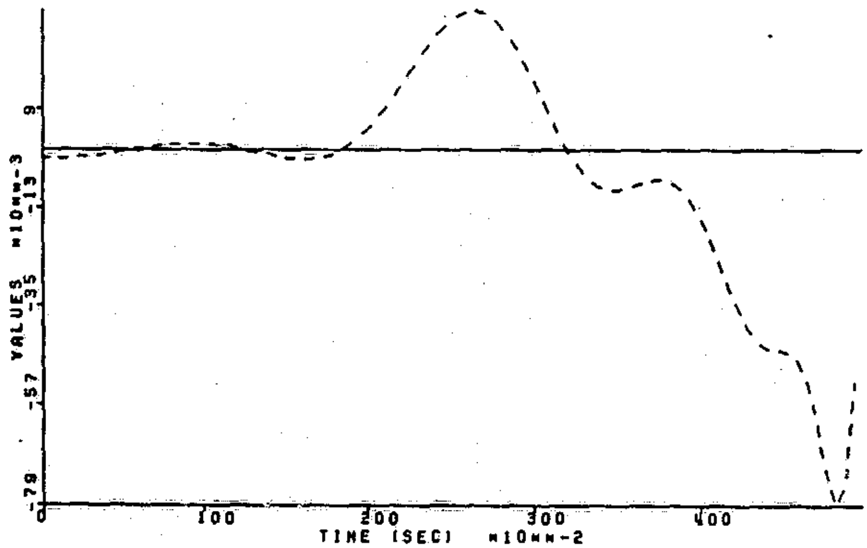
QUADRATIC FEEDBACK ELEMENT # 2



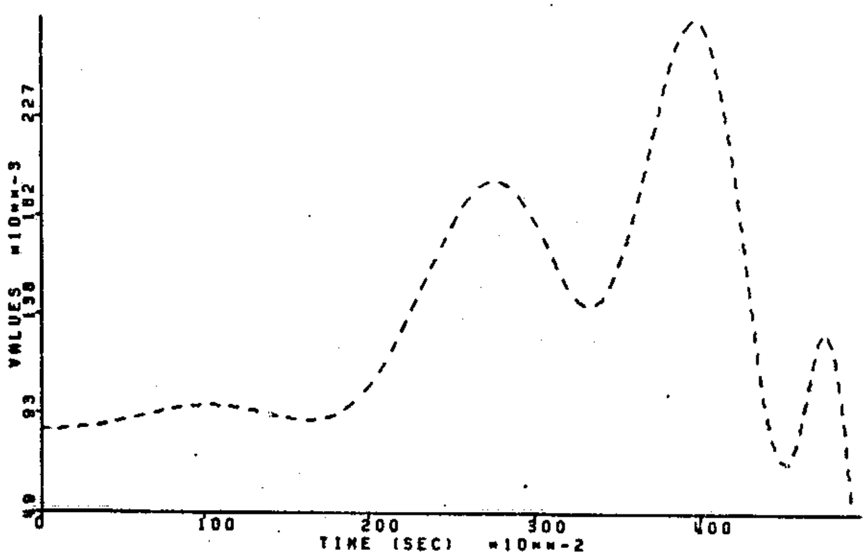
QUADRATIC FEEDBACK ELEMENT * 3



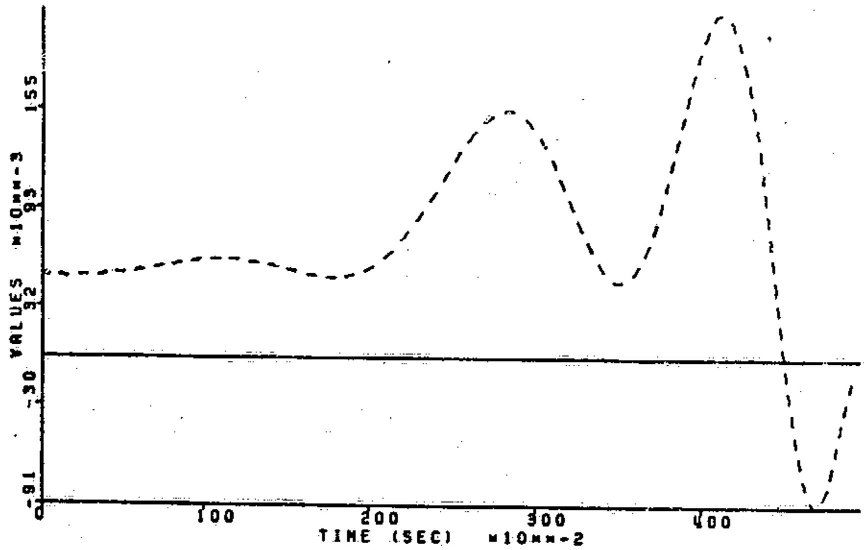
QUADRATIC FEEDBACK ELEMENT * 4



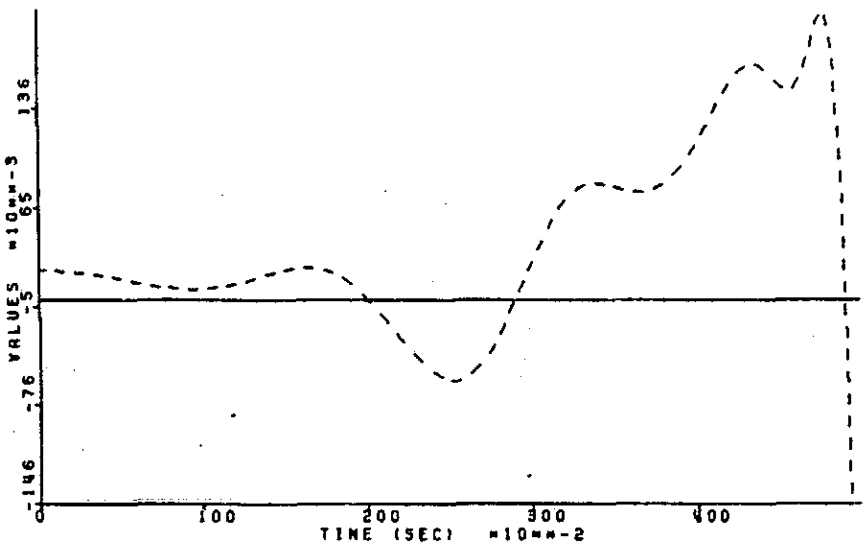
QUADRATIC FEEDBACK ELEMENT # 5



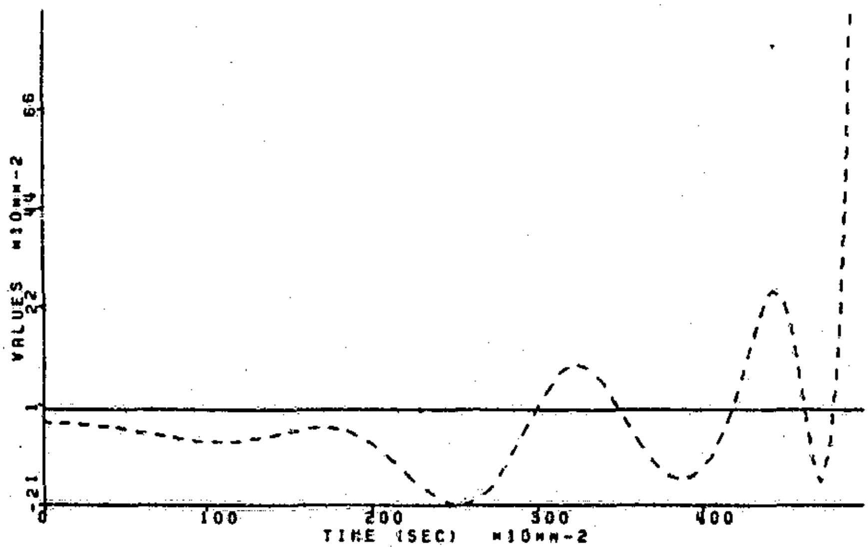
QUADRATIC FEEDBACK ELEMENT # 6



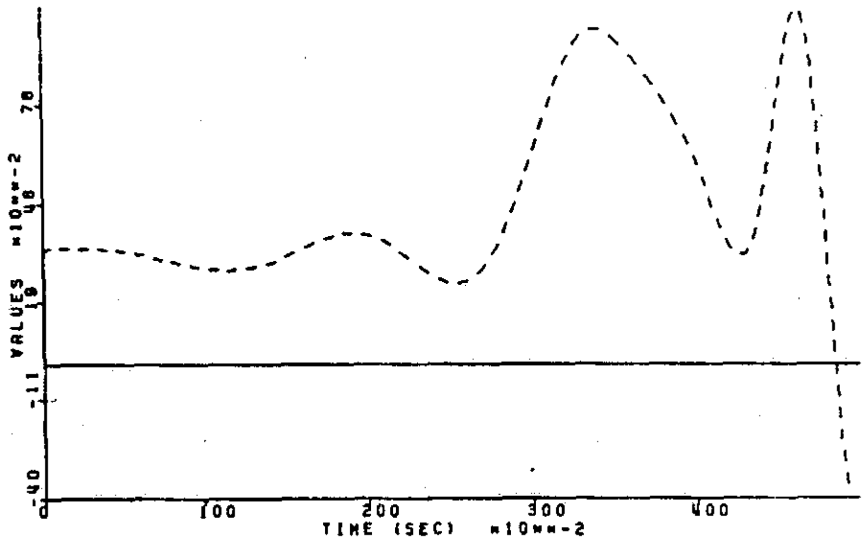
THIRD ORDER FEEDBACK ELEMENT 1



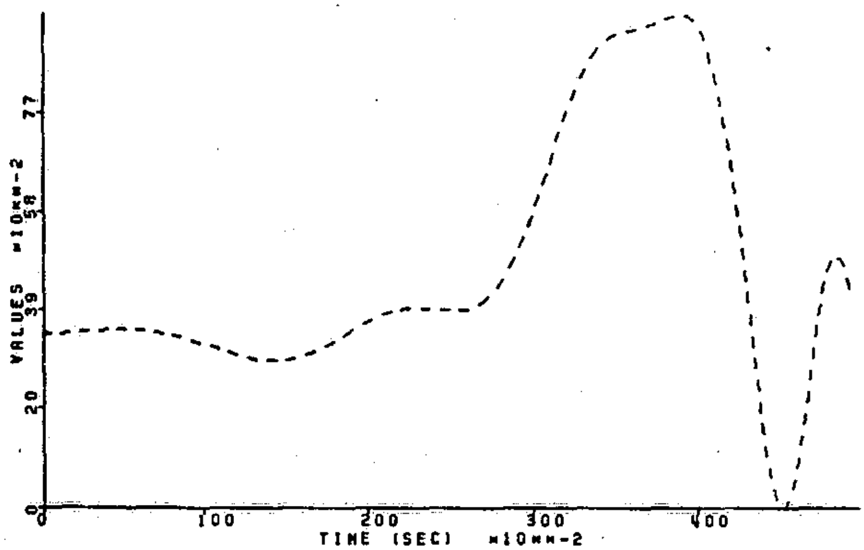
THIRD ORDER FEEDBACK ELEMENT 2

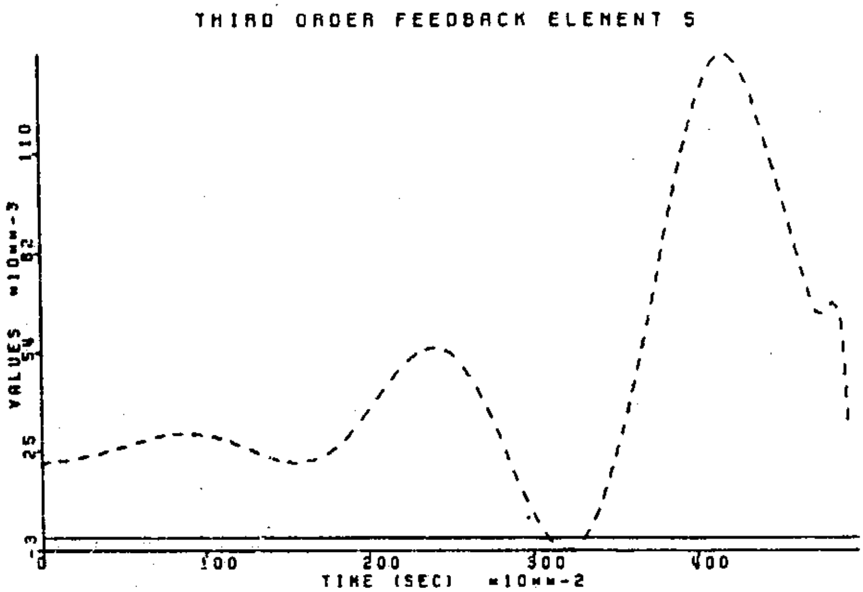


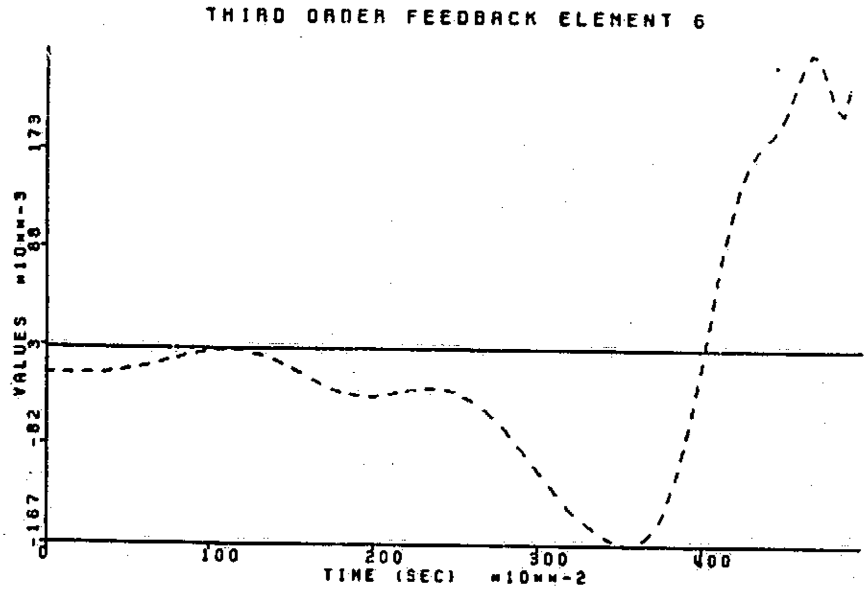
THIRD ORDER FEEDBACK ELEMENT 3



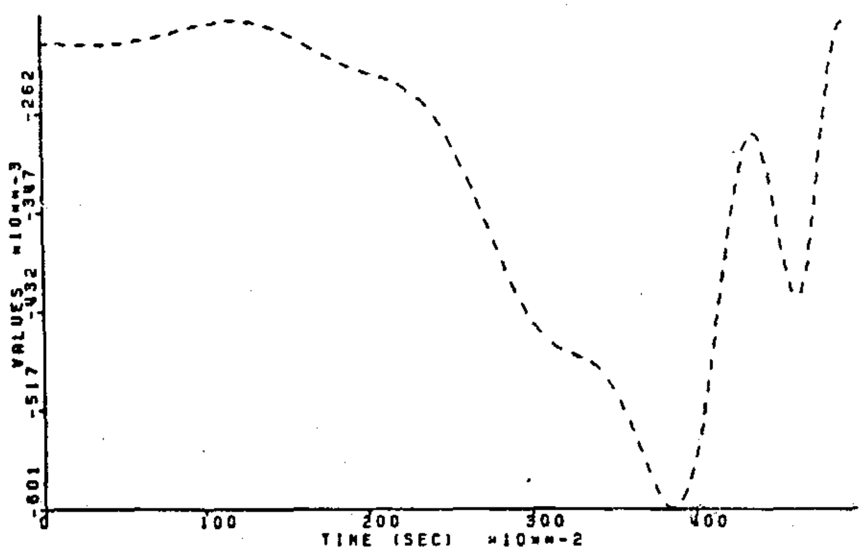
THIRD ORDER FEEDBACK ELEMENT 4



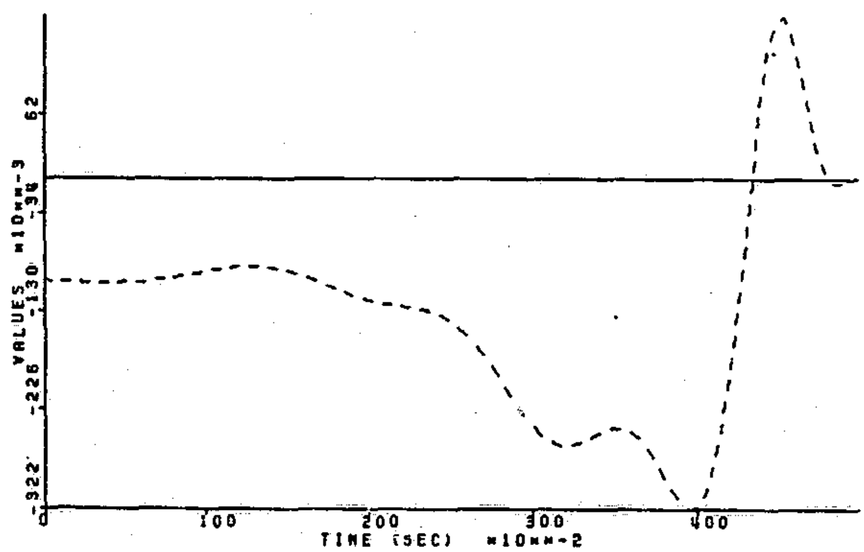

THIRD ORDER FEEDBACK ELEMENT 5
VALUES *10**-3
-3
25
54
82
110
0
100
200
300
400
TIME (SEC) *10**-2


THIRD ORDER FEEDBACK ELEMENT 6
VALUES *10**-3
-167
-82
3
88
173
0
100
200
300
400
TIME (SEC) *10**-2

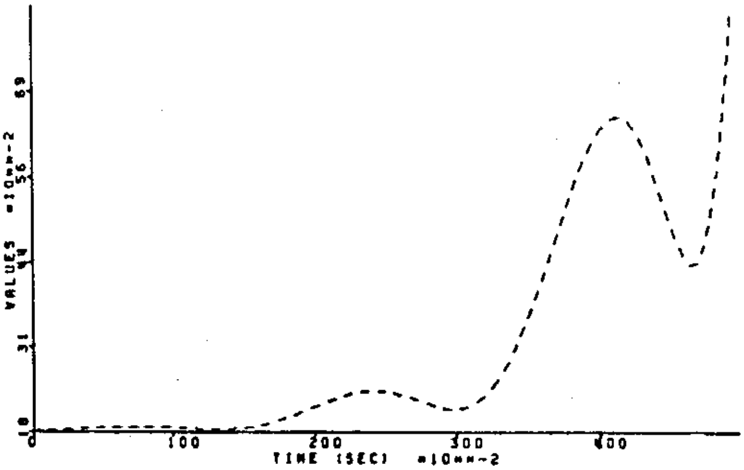
THIRD ORDER FEEDBACK ELEMENT 7



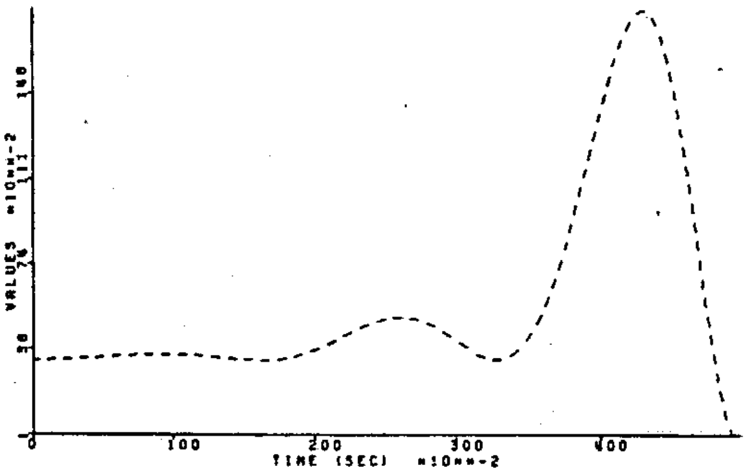
THIRD ORDER FEEDBACK ELEMENT 8



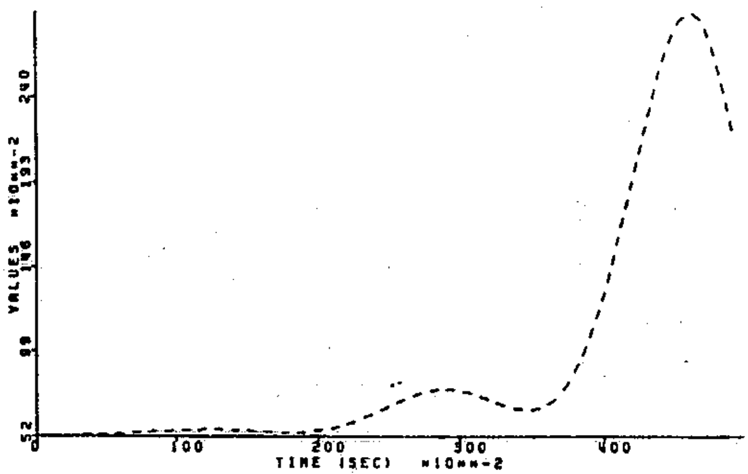
SECOND ORDER COST, ELEMENT = 1

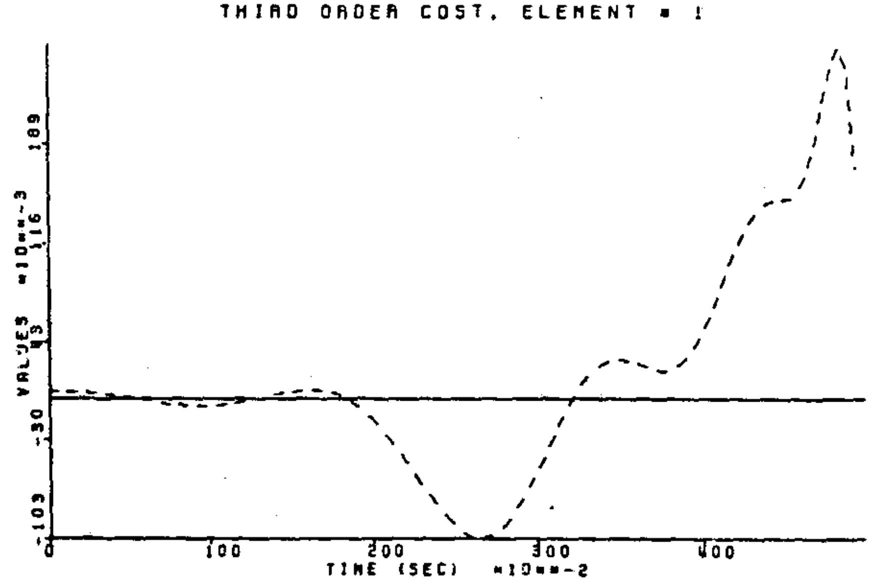


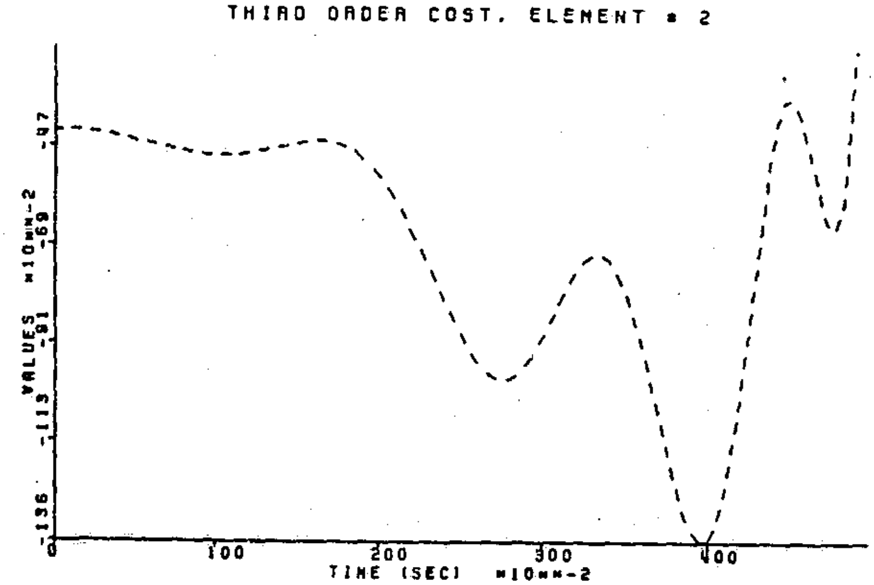
SECOND ORDER COST, ELEMENT = 2



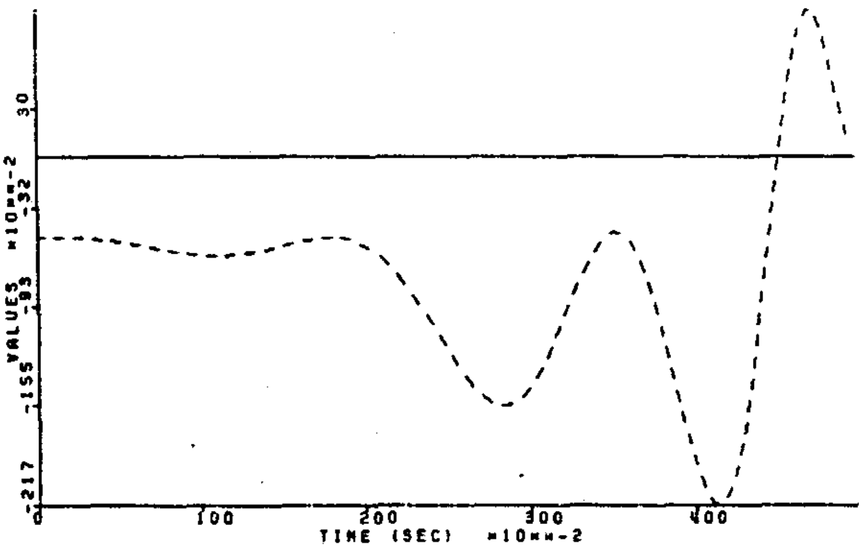
SECOND ORDER COST, ELEMENT = 3



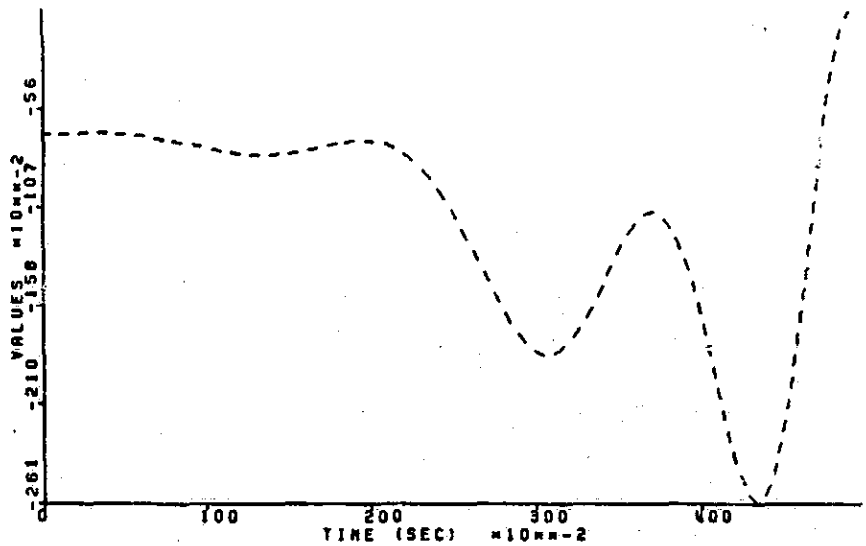

THIRD ORDER COST, ELEMENT = 1
VALUES *10**-3
189
116
43
-30
-103
0
100
200
300
400
TIME (SEC) *10**-2


THIRD ORDER COST, ELEMENT = 2
VALUES *10**-2
-47
-69
-91
-113
-136
0
100
200
300
400
TIME (SEC) *10**-2

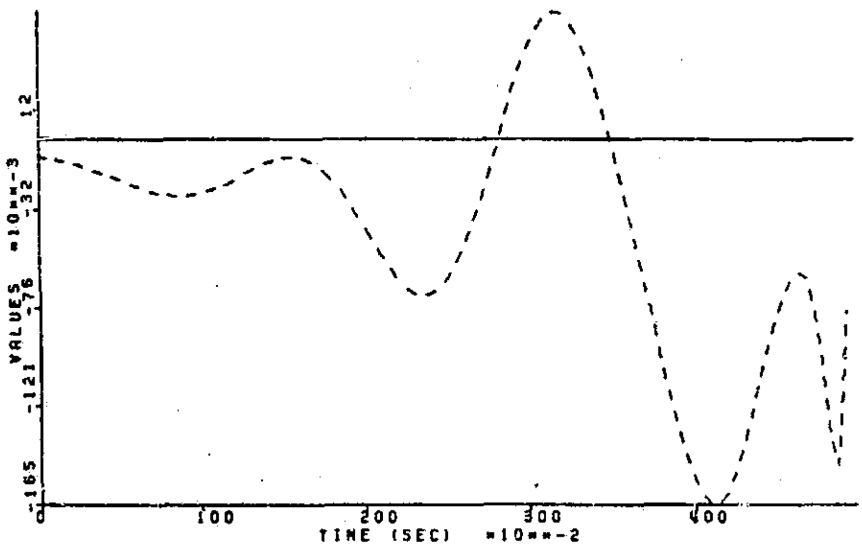
THIRD ORDER COST, ELEMENT # 3



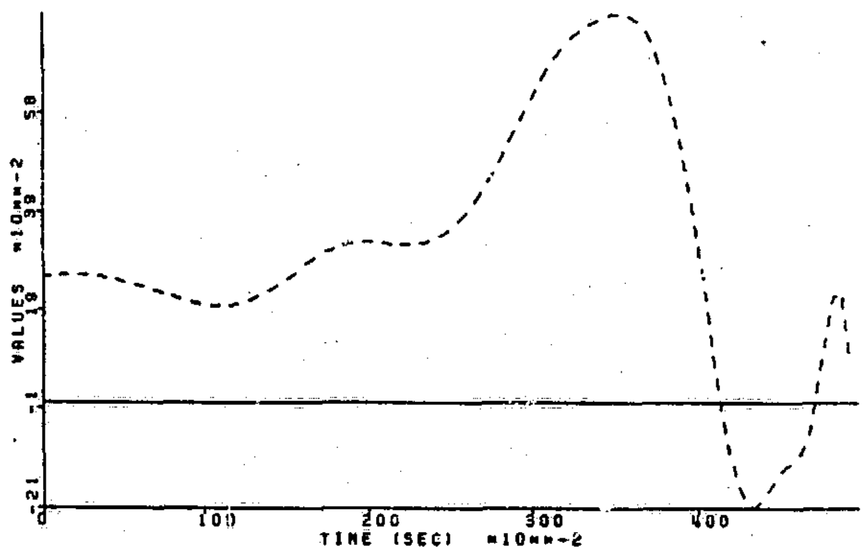
THIRD ORDER COST, ELEMENT # 4



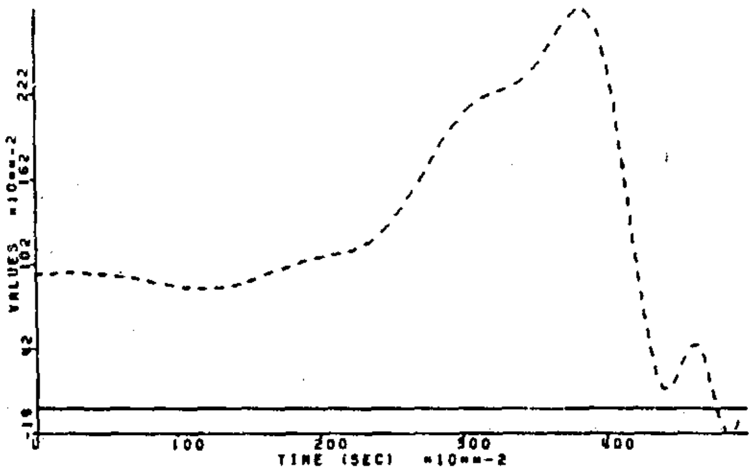
FOURTH ORDER COST, ELEMENT = 1



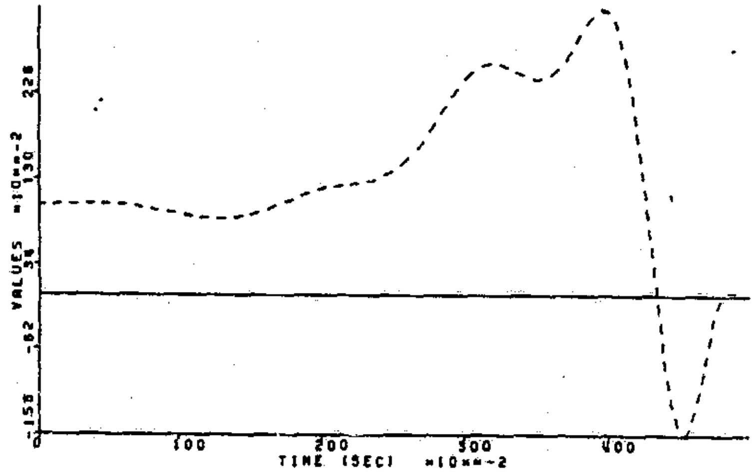
FOURTH ORDER COST, ELEMENT = 2



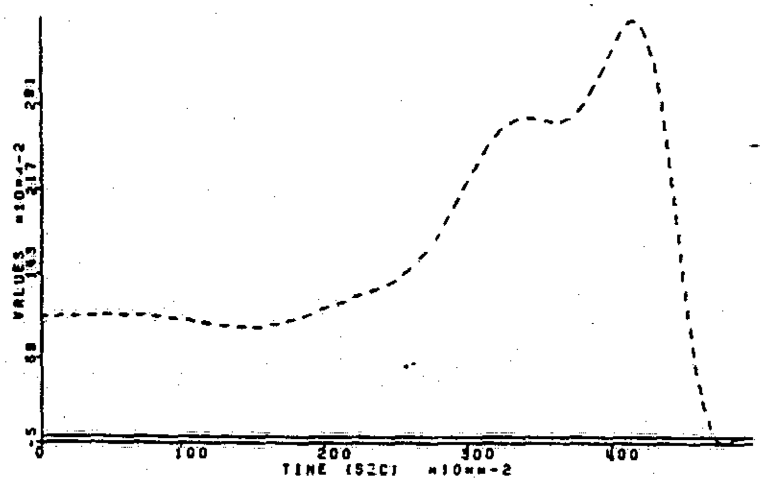
FOURTH ORDER COST, ELEMENT = 3



FOURTH ORDER COST, ELEMENT = 4



FOURTH ORDER COST, ELEMENT = 5



REFERENCES

1. M.R. Buric, "Optimal Regulation of Nonlinear Systems by Methods by Tensor Algebra", Ph.D Dissertation, Dept. of Electrical Engineering, University of Minnesota, January 1978.

2. D.L. Lukes, "Optimal Regulation of Nonlinear Dynamic Systems", SIAM Journal of Control, Vol. 7, No. 1, pp. 75-100, 1969.

3. J.P. Hill, "Solution of Nonlinear Optimal Control Problems Using the Algebraic Tensor: An Example", M.S.E.E. Thesis, Dept. of Electrical Engineering, University of Notre Dame, May 1982.

4. D.G. Luenberger, Optimization by Vector Space Methods, New York: Wiley and Sons, 1969.

5. E.B. Lee and L. Markus, Foundations of Optimal Control Theory, New York: Wiley and Sons, 1967.

6. M. Athans and P.L. Falb, Optimal Control, New York: McGraw-Hill, 1966.

7. S. Yurkovich, "Application of Tensor Ideas to Nonlinear Modeling and Control", M.S.E.E. Thesis, Dept. of Electrical Engineering, University of Notre Dame, January 1981.

8. S. Yurkovich, "Multilinear Modeling and Linear Analysis of Nonlinear Systems", Ph.D Dissertation, Dept. of Electrical Engineering, University of Notre Dame, May 1984.

9. D. Bugajski, "Monomial Reduction of Nonlinear Tensor Models", M.S.E.E. Thesis, Dept. of Electrical Engineering, University of Notre Dame, May 1984.

10. L. McWilliams, "Nonlinear Models by Layered Tensors", M.S.E.E. Thesis, Dept. of Electrical Engineering, University of Notre Dame, May 1984.

11. W. Greub, Linear Algebra, Fourth Edition, New York: Springer-Verlag, 1981.

12. W. Greub, Multilinear Algebra, Second Edition, New York: Springer-Verlag, 1978.

13. W. Merrill, et al., "HYTESS-A Hypothetical Turbofan Engine Simplified Simulation", NASA Technical Memorandum 83561, January 1984.

14. M. Vidyasagar, Non-linear Stability Theory, Prentice-Hall, Englewood Cliffs, NJ, 1978.