

156
74

INCORPORATING GLOBAL INFORMATION INTO LOCAL
NONLINEAR CONTROLLERS

by

Chris G. Stewart

Thesis submitted to Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for degree of

MASTER OF SCIENCE

in

Electrical Engineering

APPROVED:



Hugh F. Vanlandingham



William T. Baumann, Chairman



John S. Bay

September 1990

Blacksburg, Virginia

c.2

LD
5655
V855
1990
S739
c.2

INCORPORATING GLOBAL INFORMATION INTO LOCAL
NONLINEAR CONTROLLERS

by

Chris G. Stewart

Committee Chairman: William T. Baumann

Electrical Engineering

(ABSTRACT)

For a particular equilibrium point, the local performance is determined by the partial derivatives of the control law evaluated at the equilibrium point. For a linear controller, the derivatives are equal to the state-feedback gains and the gains on the external inputs. These gains can be changed to vary the local performance of the system. An extended-linear controller links together the desired local controllers of various equilibrium points producing a nonlinear controller with the desired characteristics in a neighborhood of the equilibrium curve.

Global performance is the behavior of the the system away from the equilibrium curve. Although the extended-linear controller has good local performance, the global performance might be poor or even unstable. This thesis uses cubic spline techniques to investigate the coupling of global information into local controllers without affecting the local performance. Although "stand alone" interpolative spline structures do not give the desired local performance, global information can be splined into linear and extended-linear controllers to provide both good global and good local performance.

Acknowledgements

First, I would like to thank God who makes everything possible. Next, I would like to thank Dr. Baumann, who made this project possible. I would also like to acknowledge the financial support of the DuPont Company, under a *Young Faculty Development Grant* awarded to Dr. Baumann, and the Air Force Office of Scientific Research, under contract # AFOSR-89-0495. Finally, I would like to thank Wendy, for her “wonderful” love, my parents for their eternal love, my twin sister, Marian, and big bro.

Table of Contents

	Page
Abstract	ii
Acknowledgements.....	iii
1.0 Introduction.....	1
2.0 Linear and Extended-Linear Controllers.....	5
2.1 Linear Controllers.....	5
2.2 Extended-Linear Controllers.....	11
3.0 Single Equilibrium Point Global Controllers.....	23
3.1 Splines.....	24
3.2 First Prototype Controller.....	26
Table of Contents	iv

3.3 Second Prototype Controller	38
3.4 Single Equilibrium Point Baseline Controller	47
4.0 Varying External Input Global Controllers	76
4.1 Extended S.E.P. Baseline Controller	76
4.2 Varying Equilibrium Point Baseline Controller	84
5.0 Conclusion	94
List of References.....	96
Appendix.....	98
Vita.....	126

1.0 INTRODUCTION

Global performance is the behavior of the system away from the equilibrium curve. A highly desired property of any system is to have good global performance without sacrificing the desired local performance. This thesis investigates the coupling of global information into local controllers without affecting the desired local performance.

This thesis will only discuss the extended-linearization method for solving nonlinear control problems because it applies to many systems under very mild assumptions. Other methods such as feedback-linearization, immersion, and Lyapunov functions are very powerful methods, but they apply to restrictive classes of systems. For example, in the single input case, feedback-linearization [7- 10] only applies to the nonlinear systems of the form

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}) + \mathbf{G}(\mathbf{X}) \cdot \phi(\mathbf{X}, \mathbf{U}) \quad (1.1)$$

where $X \in \mathbb{R}^n$, $U \in \mathbb{R}$, $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$, $G: \mathbb{R}^n \rightarrow \mathbb{R}^n$, and $\phi: \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$. The system must also satisfy a restrictive involutivity constraint. The idea is to find an invertible transformation from one state-control space to another state-control space such that the nonlinear system (1.1) takes the form of a linear system in the new state-control space. The feedback-linearization technique relies on the often unrealistic assumption of knowing the exact state of the nonlinear system.

The immersion method [11-14] is limited to systems that have the number of inputs greater than or equal to the number of outputs. Also, as with the feedback-linearization method, the immersion method requires the exact knowledge of the state of the nonlinear system. The objective is to find a nonlinear feedback that makes the input-output behavior of the nonlinear system identical to that of a linear system.

The Lyapunov method [15-17] requires that a Lyapunov function be determined, a formidable task that might not even have a solution. Additionally, the Lyapunov functions apply to a very particular class of systems that motivates their determination.

The extended-linearization method applies to a much more general class of systems under very mild assumptions, while producing a result which applies at all equilibrium points in a neighborhood of a nominal equilibrium point. References [1-3] explored the control of a general class of nonlinear systems by considering the family of linearizations of the systems about their set of equilibrium points. Given that a family of linear state-feedback controllers have been designed to meet some linear control objectives, the references have proven that these controllers can be pieced together into an overall, nonlinear, state-feedback controller. This “extended-linearization” method for designing

local nonlinear controllers does, under certain conditions, guarantee stability in a neighborhood of the equilibrium set, but this method does not guarantee good global performance. The global performance can be indirectly influenced by tuning the set of local controllers, but this sacrifices the desired local performance for good global performance, and essentially the global performance still remains unspecified.

As stated before, good local performance does not guarantee good global performance. Using linear state-feedback controllers or extended-linear controllers, a designer can “trade-off” good local performance for better global performance; however the need exists for a controller that obtains both good local and good global performance with no trade-off. To develop this controller, this research examined several variations of spline-based design techniques in an attempt to unify the desired local and global performance into one nonlinear controller framework.

Chapter two will review the linearization of nonlinear systems to form linear state-feedback controllers. Also chapter two will review extended-linearization. New issues will start to be addressed in chapter three which will discuss global control for a single equilibrium point (S.E.P.) situation. A S.E.P situation is where the external control is assumed to be constant, and the controller is designed for that particular constant value. This chapter will discuss the development of two prototypes before discussing the final S.E.P. baseline approach. Although the prototypes will show poor results, they will illustrate some of the limitations and restrictions of global controllers developed in the context of multivariable splines.

In chapter four, a varying external input to the system will be examined. This implies that for

each constant external input value, the system should obtain a different desired equilibrium point. The techniques of the S.E.P. baseline controller will be applied to a system that has a varying external input, but this “extended S.E.P. baseline” controller will produce poor results. However, its documentation will prove helpful in understanding global control. Finally, in chapter four, a successful varying equilibrium point (V.E.P.) baseline controller will be discussed. Chapter five will summarize the results.

2.0 LINEAR AND EXTENDED LINEAR CONTROLLERS

Given a nonlinear system, the state equation form is shown by

$$\dot{\mathbf{X}}(t) = \mathbf{F}[\mathbf{X}(t), \mathbf{U}(t)] \quad (2.1)$$

where $\mathbf{X}(t)$ is an $n \times 1$ state vector, $\mathbf{U}(t)$ is an $m \times 1$ input vector, and $\mathbf{F}[\mathbf{X}(t), \mathbf{U}(t)]$ is an $n \times 1$ function vector. We only consider the autonomous case where \mathbf{F} is a function of only the state vector and input vector and does not depend explicitly on t .

2.1 Linear Controllers

An equilibrium point is a point in state-control space when the trajectories of the states have become stationary, i.e.

$$\mathbf{F}[\mathbf{X}^\circ, \mathbf{U}^\circ] = 0 \quad (2.2)$$

The set of state vectors and input vectors that satisfy condition (2.2) constitutes an equilibrium curve or surface. For each equilibrium state vector on the equilibrium curve or surface, denoted \mathbf{X}° , there corresponds an equilibrium input vector, \mathbf{U}° , which causes the system to remain stationary. If we pick a nominal operating point, $(\mathbf{X}^\circ, \mathbf{U}^\circ)$, on the equilibrium curve or surface, then by using a Taylor series expansion, the nonlinear system can be expanded about this operating point such that

$$\begin{aligned} \dot{x}_i(t) = f_i(\mathbf{X}^\circ, \mathbf{U}^\circ) + \sum_{j=1}^n \left. \frac{\partial f_i(\mathbf{X}, \mathbf{U})}{\partial x_j} \right|_{\mathbf{X}^\circ, \mathbf{U}^\circ} (x_j - x_j^\circ) + \\ \sum_{j=1}^m \left. \frac{\partial f_i(\mathbf{X}, \mathbf{U})}{\partial u_j} \right|_{\mathbf{X}^\circ, \mathbf{U}^\circ} (u_j - u_j^\circ) + \text{Higher order terms} \end{aligned} \quad (2.3)$$

for $i = 1, 2, \dots, n$. Note that $\dot{x}_i(t)$ are components of $\dot{\mathbf{X}}(t)$. The functions $f_i(\mathbf{X}^\circ, \mathbf{U}^\circ)$ are components of \mathbf{F} evaluated at the operating point and are equal to zero for all i due to equation (2.2). If we neglect higher order terms and let

$$\Delta x_j = x_j - x_j^\circ \quad ; \quad (2.4)$$

and

$$\Delta u_j = u_j - u_j^\circ \quad ; \quad (2.5)$$

and

$$\Delta \dot{x}_i = \dot{x}_i - f_i(\mathbf{X}^\circ, \mathbf{U}^\circ) = \dot{x}_i \quad ; \quad (2.6)$$

then

$$\Delta \dot{x}_i = \sum_{j=1}^n \left. \frac{\partial f_i(\mathbf{X}, \mathbf{U})}{\partial x_j} \right|_{\mathbf{X}^\circ, \mathbf{U}^\circ} \Delta x_j + \sum_{j=1}^m \left. \frac{\partial f_i(\mathbf{X}, \mathbf{U})}{\partial u_j} \right|_{\mathbf{X}^\circ, \mathbf{U}^\circ} \Delta u_j \quad (2.7)$$

Combining these equations yields

$$\Delta \dot{\mathbf{X}} = \mathbf{D}_1 \mathbf{F}(\mathbf{X}^\circ, \mathbf{U}^\circ) \cdot \Delta \mathbf{X} + \mathbf{D}_2 \mathbf{F}(\mathbf{X}^\circ, \mathbf{U}^\circ) \cdot \Delta \mathbf{U} \quad (2.8)$$

where

$$\mathbf{D}_1 \mathbf{F}(\mathbf{X}, \mathbf{U}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (2.9)$$

$$\mathbf{D}_2 \mathbf{F}(\mathbf{X}, \mathbf{U}) = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \frac{\partial f_n}{\partial u_2} & \cdots & \frac{\partial f_n}{\partial u_m} \end{bmatrix} \quad (2.10)$$

Note that $D_1F(\mathbf{X}^\circ, \mathbf{U}^\circ)$ and $D_2F(\mathbf{X}^\circ, \mathbf{U}^\circ)$ vary with the operating point. Therefore at different operating points, there are different linearizations. Assuming that the equilibrium curve is smooth and continuous, then the equilibrium curve generates a continuous family of linearizations of the form (2.8).

The linearization of (2.1) about an equilibrium point allows the use of linear state-feedback to control the nonlinear system. State-feedback has the form

$$\mathbf{U} = \mathbf{H} \cdot \mathbf{W} - \mathbf{G} \cdot \mathbf{X} \quad (2.11)$$

where \mathbf{W} is an $m \times 1$ external input vector, \mathbf{H} is an $m \times m$ input gain matrix, and \mathbf{G} is an $m \times n$ feedback gain matrix. Generally, \mathbf{H} is the identity matrix. For different values of \mathbf{G} , the system achieves different control objectives, such as eigenvalue placement, amount of overshoot, or rise time. Intuitively, as long as the closed-loop system is operated sufficiently close to the equilibrium point, the behavior of the system should meet the design objectives. When the system operates away from the equilibrium point, the system might perform poorly.

Throughout this thesis, the different approaches will be demonstrated and compared using a model of a continuously stirred tank reactor that has two states and one input. When several chemicals are added to the tank and stirred, a reaction occurs to produce a desired chemical output. The percent of conversion to the final output is state x_1 , and the temperature at which it occurs is state x_2 . As the temperature at which the reaction occurs is increased, more of the chemicals are converted, but also the plant's lifetime decreases. The plant operator must choose a reasonable conversion rate that balances these two factors. An additional consideration is that if the temperature gets too high,

then the plant might catastrophically fail. The dimensionless model for a constant volume, non-adiabatic, continuously stirred tank reactor is given by the equations

$$\dot{x}_1 = -x_1 + D \cdot (1 - x_1) \cdot e^{\frac{x_2}{(1+x_2/\gamma)}} \quad (2.12)$$

$$\dot{x}_2 = -x_2 + B \cdot D \cdot (1 - x_1) \cdot e^{\frac{x_2}{(1+x_2/\gamma)}} - u \cdot (x_2 - x_c) \quad (2.13)$$

where x_1 is the conversion, x_2 is the dimensionless temperature, u is the dimensionless heat transfer coefficient, x_c is the dimensionless cooling jacket temperature, D is the Damkohler number, and B is the dimensionless heat of reaction. For this example, $x_c = 0.0$, $D = 0.05$, $B = 8.0$, and $\gamma = 2100$. Figure 2.1 shows the equilibrium curve for this system.

For chapters two and three, the external input into the controller, w , is held constant such that the system will attain the equilibrium point; $x_1^o = 0.5$, and $x_2^o = 3.0$. This point, corresponding to moderate conversion and temperature, is unstable, and requires compensation to achieve it. In chapter four, the external input varies.

A continuous-time linear controller was designed to place the eigenvalues of the closed-loop linearization about this operating point at -2.0 . The form of the controller is

$$U(x_1, x_2) = H \cdot w - G_1 \cdot x_1 - G_2 \cdot x_2 \quad (2.14)$$

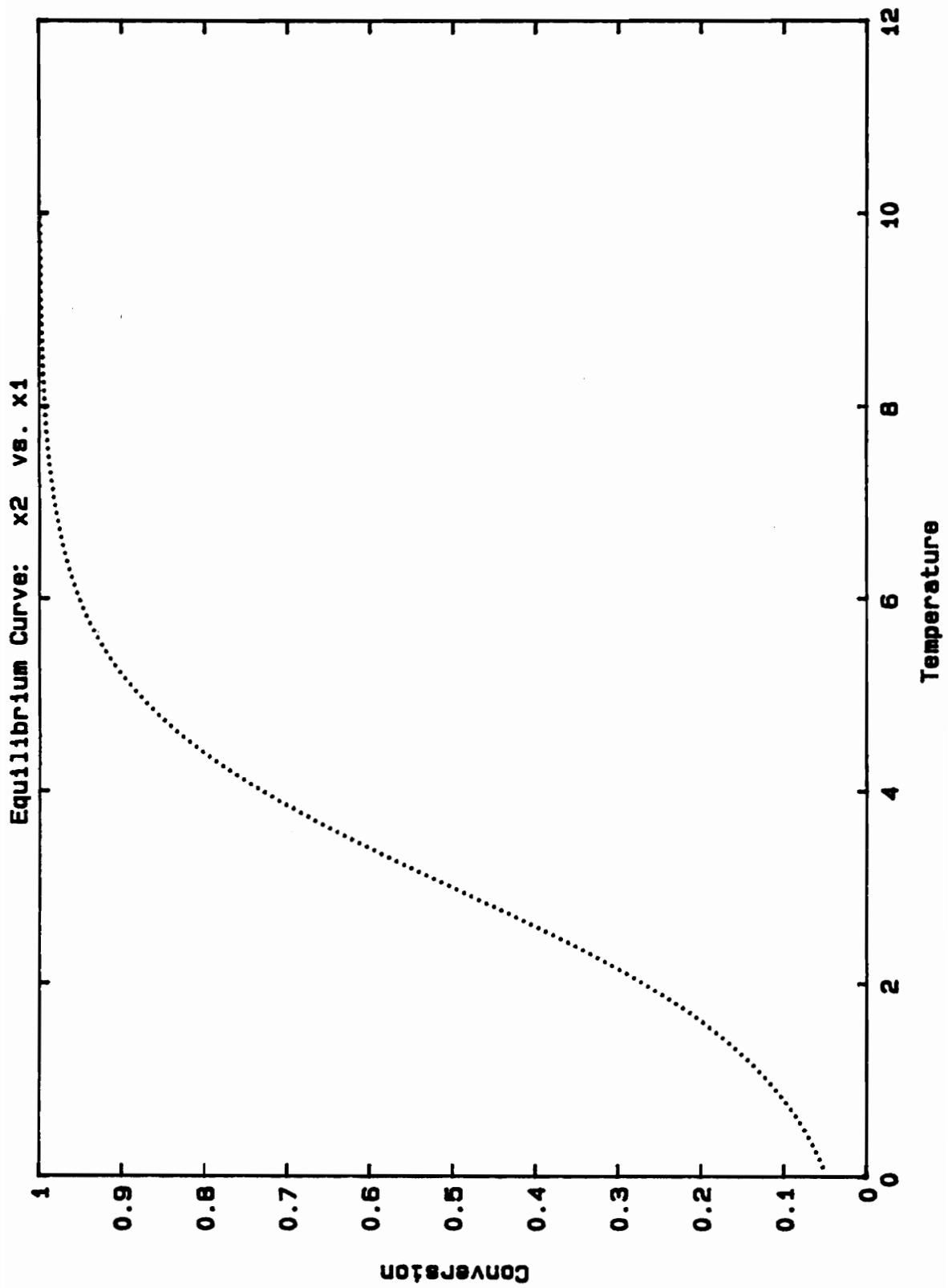


Figure 2.1

where $H = 1.0$, $w = -2.989$, $G_1 = 2.667$, and $G_2 = -1.552$. The phase trajectories of this controller are shown in figure 2.2. For initial conditions in the neighborhood of $\mathbf{X} = [0.1 \ 5.0]^T$, the system has transients with large overshoots in both temperature and conversion. Global control information is needed for this region.

2.2 Extended-Linear Controller

Consider a nonlinear system, suppressing t -arguments, described by

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, \mathbf{U}) \quad (2.15)$$

where \mathbf{X} is an $n \times 1$ vector, \mathbf{U} is an $m \times 1$ vector, \mathbf{F} is continuously differentiable, and $\mathbf{F}(0, 0) = 0$. As stated in section 2.1, this system, at least locally, has a continuous family of equilibrium points, and thus a continuous family of linearizations. Given a family of linear state-feedback controllers, the extended-linear controller links together the linear controllers into an overall, nonlinear, state feedback controller. The resulting extended-linear controller has the property that at each equilibrium point, the linearization of the closed-loop nonlinear system equals the desired closed-loop linearization designed for the linear controller. Therefore, as long as the closed-loop system is operated in a region sufficiently close to the equilibrium curve, the behavior of the system should meet the desired linear design objectives. However, the extended-linear controller does not account for states far from the equilibrium curve which might result in poor global performance.

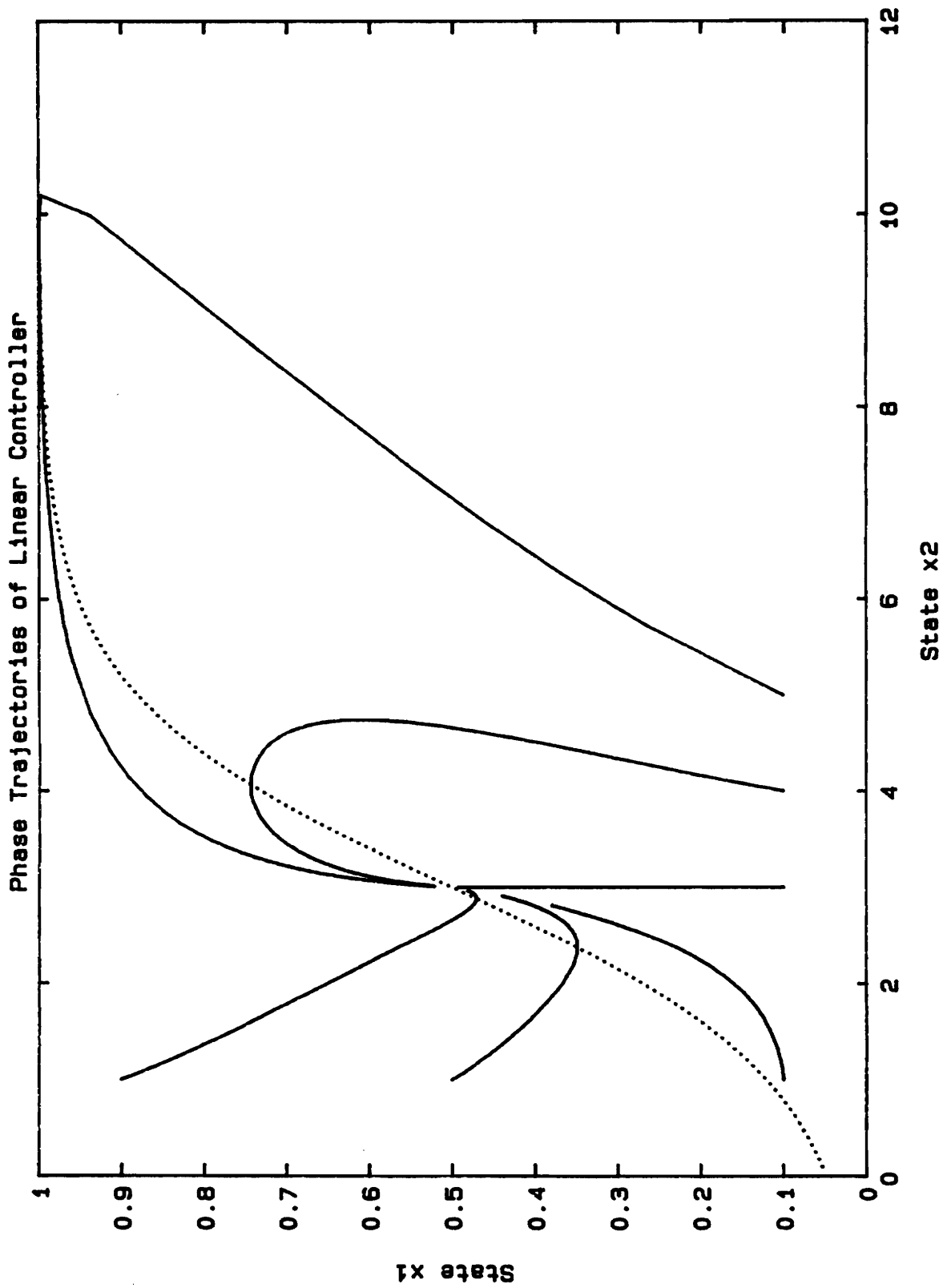


Figure 2.2

The extended-linearization method may be viewed as a theoretical formulation and extension of the technique of gain scheduling. The equilibrium points, $(\mathbf{X}^\circ, \mathbf{U}^\circ)$, of the system (2.15) will be parametrized by a vector $\alpha \in \mathfrak{U} \subseteq \mathfrak{R}^m$ such that

$$\mathbf{F}(\mathbf{X}^\circ(\alpha), \mathbf{U}^\circ(\alpha)) = 0 \quad (2.16)$$

for all α in a neighborhood of the origin such that $\mathbf{X}^\circ(0)=0$, $\mathbf{U}^\circ(0)=0$, and $\partial\mathbf{U}^\circ(0)/\partial\alpha$ is invertible.

At each equilibrium point, the open-loop linearization is

$$\Delta\dot{\mathbf{X}} = D_1\mathbf{F}(\mathbf{X}^\circ(\alpha), \mathbf{U}^\circ(\alpha)) \cdot \Delta\mathbf{X} + D_2\mathbf{F}(\mathbf{X}^\circ(\alpha), \mathbf{U}^\circ(\alpha)) \cdot \Delta\mathbf{U} \quad (2.17)$$

Designing a feedback controller of the form

$$\Delta\mathbf{U} = \mathbf{H}(\alpha) \cdot \Delta\mathbf{W} - \mathbf{G}(\alpha) \cdot \Delta\mathbf{X} \quad (2.18)$$

results in a desired closed-loop system of the form

$$\begin{aligned} \Delta\dot{\mathbf{X}} &= \left[D_1\mathbf{F}(\mathbf{X}^\circ, \mathbf{U}^\circ) - D_2\mathbf{F}(\mathbf{X}^\circ, \mathbf{U}^\circ) \cdot \mathbf{G}(\alpha) \right] \cdot \Delta\mathbf{X} \\ &+ \left[D_2\mathbf{F}(\mathbf{X}^\circ, \mathbf{U}^\circ) \cdot \mathbf{H}(\alpha) \right] \cdot \Delta\mathbf{W} \end{aligned} \quad (2.19)$$

Now consider a nonlinear feedback controller for the original system of the form

$$\mathbf{U} = \mathbf{S}(\mathbf{X}, \mathbf{W}) \quad (2.20)$$

The closed-loop system is

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, \mathbf{S}(\mathbf{X}, \mathbf{W})) \quad (2.21)$$

and has an equilibrium set $(\mathbf{X}^c(\beta), \beta)$, parametrized by constant values of the input, such that

$$\mathbf{F}(\mathbf{X}^c(\beta), \beta) = \mathbf{0} \quad (2.22)$$

The closed-loop and open-loop equilibrium points are related by a function $\mathbf{a}(\cdot)$ such that

$$\mathbf{X}^c(\beta) = \mathbf{X}^o(\mathbf{a}(\beta)) \quad (2.23)$$

$$\mathbf{S}(\mathbf{X}^c(\beta), \beta) = \mathbf{U}^o(\mathbf{a}(\beta)) \quad (2.24)$$

where $\alpha = \mathbf{a}(\beta)$. The linearization of the closed-loop nonlinear system is given by

$$\begin{aligned} \Delta \dot{\mathbf{X}} = & \left[D_1 \mathbf{F}(\mathbf{X}^o, \mathbf{U}^o) + D_2 \mathbf{F}(\mathbf{X}^o, \mathbf{U}^o) \cdot D_1 \mathbf{S}(\mathbf{X}^c, \beta) \right] \cdot \Delta \mathbf{X} \\ & + \left[D_2 \mathbf{F}(\mathbf{X}^o, \mathbf{U}^o) \cdot D_2 \mathbf{S}(\mathbf{X}^c, \beta) \right] \cdot \Delta \mathbf{W} \end{aligned} \quad (2.25)$$

The extended-linearization problem is to determine a nonlinear function \mathbf{S} such that the linearization of the closed-loop nonlinear system, equation (2.25), is equal to the desired closed-loop system, equation (2.19).

Theorem: Suppose the nonlinear system (2.15) and a function $\mathbf{G}(\cdot): \mathbb{R}^m \rightarrow \mathbb{R}^{m \times n}$ satisfy

$$\begin{aligned} \text{rank} [D_1 \mathbf{F}(0, 0)] &= n \\ \text{rank} [D_1 \mathbf{F}(0, 0) - D_2 \mathbf{F}(0, 0) \cdot \mathbf{G}(0)] &= n \end{aligned}$$

then there exist functions $\mathbf{S}(\cdot, \cdot): \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ (with $\mathbf{S}(0, 0) = 0$) and $\mathbf{H}(\cdot): \mathbb{R}^m \rightarrow \mathbb{R}^{m \times n}$ (with $\mathbf{H}(0)$ invertible) such that (2.25) is equal to (2.19) with $\alpha = \mathbf{a}(\beta)$ at each closed-loop equilibrium point. Note that this applies only in a neighborhood of the nominal equilibrium point ($\mathbf{W}=0, \mathbf{X}=0$).

The proof is given in reference [1]. An important result developed in the proof is that if the function $\mathbf{a}(\beta)$, relating the open and closed-loop equilibrium points, is chosen to be the identity function, then \mathbf{H} is uniquely determined by \mathbf{G} . Thus, not all the control laws of the form (2.18) can be scheduled. If the function $\mathbf{a}(\beta)$ is equal to the identity function, then the gradient of \mathbf{S} on the equilibrium curve is

$$D_1 \mathbf{S}(\mathbf{X}^c(\beta), \beta) = -\mathbf{G}(\alpha) \Big|_{\alpha=\beta} \quad (2.26a)$$

$$D_2 \mathbf{S}(\mathbf{X}^c(\beta), \beta) = \mathbf{G}(\alpha) \cdot \frac{\partial \mathbf{X}^o(\alpha)}{\partial \alpha} + \frac{\partial \mathbf{U}^o(\alpha)}{\partial \alpha} = \mathbf{H}(\alpha) \Big|_{\alpha=\beta} \quad (2.26b)$$

where the quantities on the right are all known open-loop quantities.

Equations (2.26a) and (2.26b) apply to the controller only on the equilibrium curve. The next

step is to extend the function \mathbf{S} off the curve. Let the $(m+n)$ -tuple of variables (z_1, \dots, z_{m+n}) be a permutation of the $m+n$ state and control variables (\mathbf{X}, \mathbf{W}) such that

$$\left[\det \frac{\partial}{\partial \beta} \left\{ \begin{array}{l} \mathbf{Z} \\ \mathbf{X} = \mathbf{X}^c(\beta) \\ \mathbf{W} = \beta \end{array} \right\} \right] \Big|_{\beta=0} \neq 0 \quad (2.27)$$

where $\mathbf{Z} = [z_1, \dots, z_m]$. This implies that there exists a vector function $\mathcal{F}(\cdot): \mathbb{R}^m \rightarrow \mathbb{R}^m$ such that

$$\beta = \mathcal{F} \left(\begin{array}{l} \mathbf{Z} \\ \mathbf{X} = \mathbf{X}^c(\beta) \\ \mathbf{W} = \beta \end{array} \right) \quad (2.28)$$

The vector \mathbf{Z} can be viewed as a set of coordinates for the equilibrium curve, and thus associates each point in the state-control space with an equilibrium point. The most obvious choice is $\mathbf{Z} = \mathbf{W}$, which implies that $\mathcal{F}(\cdot)$ is the identity operator.

A simple way to define the value for $\mathbf{S}(\mathbf{X}, \mathbf{W})$ when the point (\mathbf{X}, \mathbf{W}) is not on the equilibrium curve is as follows. Project (\mathbf{X}, \mathbf{W}) onto the equilibrium curve holding \mathbf{Z} constant. The value of \mathbf{S} at the projected point is $\mathbf{U}^o(\mathcal{F}(\mathbf{Z}))$. Let

$$D_{z_i} \mathbf{S}(\mathbf{X}^c(\beta), \beta) = \mathbf{K}_i(\beta) \quad (2.29)$$

where $D_{z_i} \mathbf{S}$ is the partial derivative of \mathbf{S} with respect to the variable z_i , and (2.29) is evaluated using (2.26a) and (2.26b). Assuming the derivatives, $D_{z_i} \mathbf{S}$, for $i = m+1, \dots, m+n$, are constant on

surfaces of constant \mathbf{Z} , then

$$D_{z_i} S(\mathbf{X}^o(\beta), \beta) = \mathbf{K}_i(\mathcal{F}(\mathbf{Z})) \quad (2.30)$$

The function $S(\mathbf{X}, \mathbf{W})$ is then

$$S(\mathbf{X}, \mathbf{W}) = \mathbf{U}^o(\mathcal{F}(\mathbf{Z})) + \sum_{i=m+1}^{m+n} \mathbf{K}_i(\mathcal{F}(\mathbf{Z})) \cdot (z_i - z_i|_{\substack{\mathbf{X}=\mathbf{X}^o(\mathcal{F}(\mathbf{Z})) \\ \mathbf{W}=\mathcal{F}(\mathbf{Z})}}) \quad (2.31)$$

For most systems, the controller (2.31) can not be computed in closed form because it may not be possible to compute the functions $\mathcal{F}(\cdot)$, $\mathbf{U}^o(\cdot)$, $\mathbf{K}_i(\cdot)$, and $\mathbf{X}^o(\cdot)$ in closed form. However, the theorem insures that a controller exists that provides the desired closed-loop linearization. Even though (2.31) can not be computed in closed form, an approximate controller can be produced. To design an approximate controller, the values of $\mathbf{X}^o(\alpha)$, $\mathbf{U}^o(\alpha)$, $\partial\mathbf{U}^o(\alpha)/\partial\alpha$, $D_1\mathbf{F}(\mathbf{X}^o(\alpha), \mathbf{U}^o(\alpha))$, and $D_2\mathbf{F}(\mathbf{X}^o(\alpha), \mathbf{U}^o(\alpha))$ would be computed or measured at a number of equilibrium points. At each of these points, $D_1\mathbf{F}(\cdot, \cdot)$ and $D_2\mathbf{F}(\cdot, \cdot)$ would be used to compute $\mathbf{G}(\alpha)$. Piecewise functions such as linear splines that interpolate the $\mathbf{X}^o(\alpha)$, $\mathbf{U}^o(\alpha)$, $\partial\mathbf{U}^o(\alpha)/\partial\alpha$, $\mathbf{G}(\alpha)$ data would be used in place of the exact functions in (2.31). The approximate controller would converge to the desired exact controller as the number of points at which these functions are known increases.

Returning to the continuously stirred tank reactor example, if the equilibrium points are parametrized by $\alpha = x_2^o$, then the equilibrium state and control are given by

$$\mathbf{X}^o(\alpha) = \begin{bmatrix} x_1^o(\alpha) \\ \alpha \end{bmatrix} \quad \text{and} \quad \mathbf{U}^o(\alpha) = [u^o(\alpha)] \quad (2.32)$$

where

$$x_1^o(\alpha) = \frac{D \cdot e^{\frac{\alpha}{1+\alpha/v}}}{1 + D \cdot e^{\frac{\alpha}{1+\alpha/v}}} \quad (2.33)$$

and

$$u^o(\alpha) = \frac{-\alpha + B \cdot x_1^o(\alpha)}{\alpha} \quad (2.34)$$

An approximate controller was designed using local information about the equilibrium points corresponding to $\alpha = 1, 2, 3, 4,$ and 5 . To provide the same performance about these equilibrium points as the linear controller, the eigenvalues of the closed-loop linearizations were chosen to be -2.0 for all five points. The following values were computed

α	$G_1(\alpha)$	$G_2(\alpha)$	$u^o(\alpha)$	$\frac{\partial u^o(\alpha)}{\partial \alpha}$
1	-5.16286	-2.86324	-0.04318	-0.1152
2	0.73404	-1.85239	0.07763	0.2470
3	2.66663	-1.55178	0.33333	0.2203
4	4.41469	-1.16275	0.46080	0.0271
5	2.57403	-0.25310	0.40799	-0.1134

The approximated functions, $U^\circ(\alpha)$, $\partial U^\circ(\alpha)/\partial\alpha$, and $G(\alpha)$, were formed using linear splines to interpolate the above data. The form of the controller, corresponding to (2.31), scheduled on the state x_2 is

$$S(X^\circ, W^\circ) = u^\circ(x_2) - G_1(x_2) \cdot (x_1 - x_1^\circ(x_2)) + D_2S(X^\circ(x_2), x_2) \cdot (w - x_2) \quad (2.35)$$

where (2.26b) was used to compute $D_2S(X^\circ(x_2), x_2)$. The functions $x_1^\circ(\alpha)$ in equation (2.35) and $\partial x_1(\alpha)/\partial\alpha$ in equation (2.26b) were computed in closed form using equation (2.33), but they could easily have been interpolated like the specified functions in the above table.

The resulting phase trajectories are shown in figure 2.3. Note that for initial conditions in the region of $X = [0.1 \ 4.0]^T$, the system is unstable. The extended-linear controller can be “tuned” by changing the local objectives, i.e. the gain matrix G , on portions of the equilibrium curve. For example, if when $\alpha=4$, the eigenvalues are placed at -4.0 , and when $\alpha=5$, the eigenvalues are placed at -6.0 , then the resulting phase trajectories are shown in figure 2.4. Figure 2.5 shows the same trajectories using a different scale. Although the system is now stable in the $X = [0.1 \ 4.0]^T$ region, the desired local performance has been sacrificed to obtain better global performance. For this example, if an equilibrium point is chosen at $X^\circ = [0.88 \ 5.0]^T$, then the local performance in the region of this equilibrium point is governed by the eigenvalues of -6.0 instead of the desired eigenvalues of -2.0 . The next chapter will try to achieve the same or better global performance without sacrificing the desired local performance.

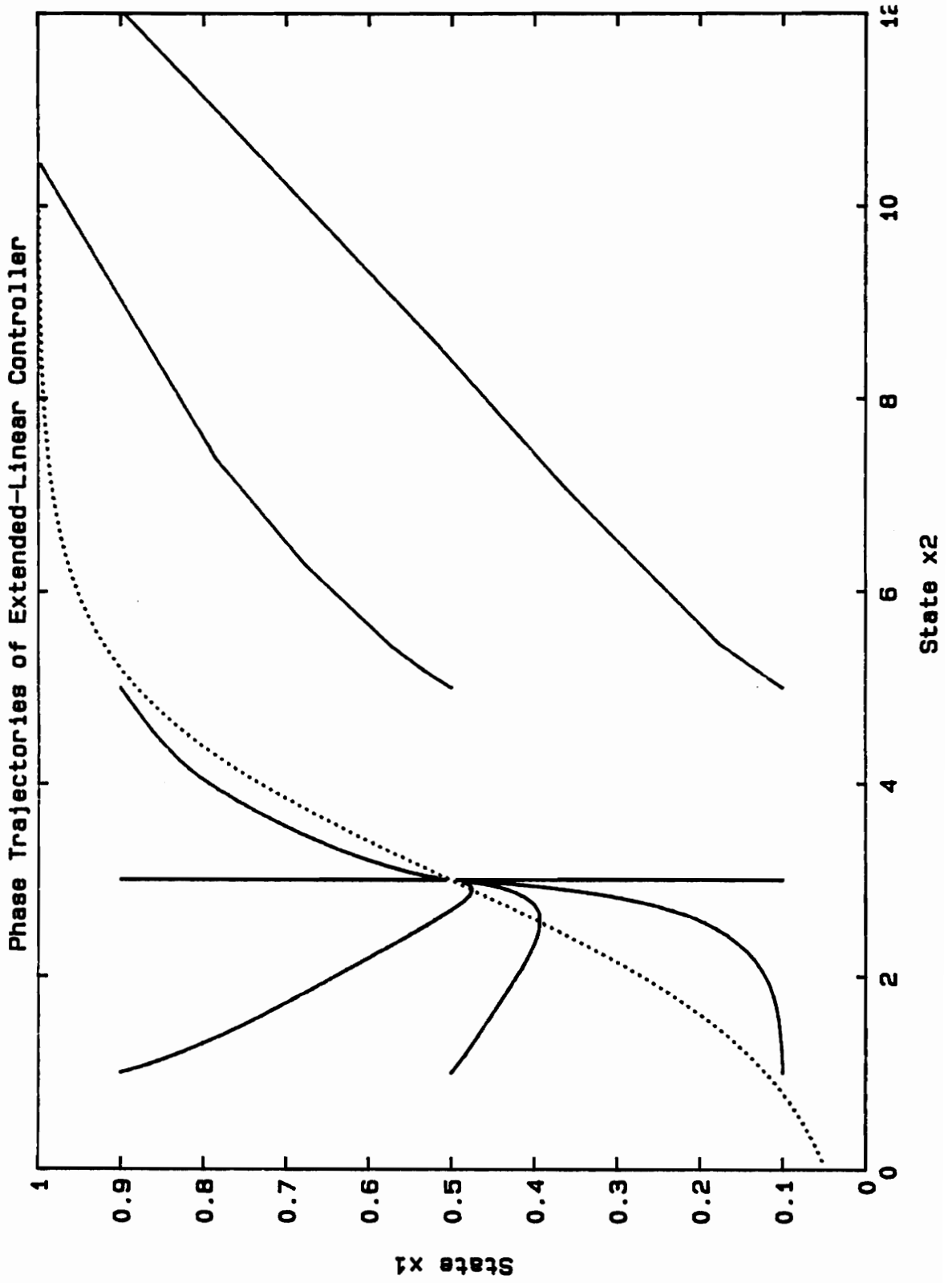


Figure 2.3

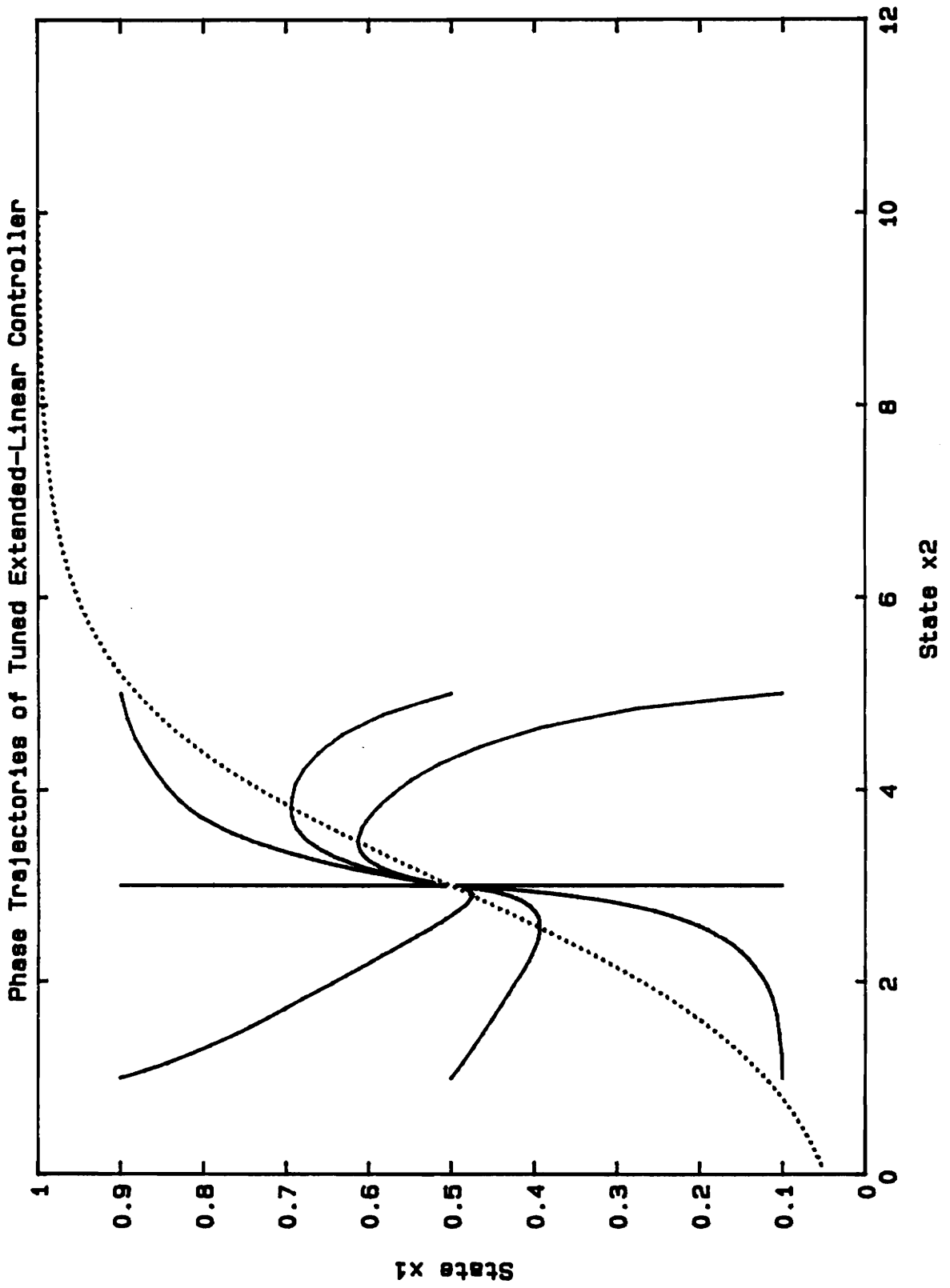


Figure 2.4

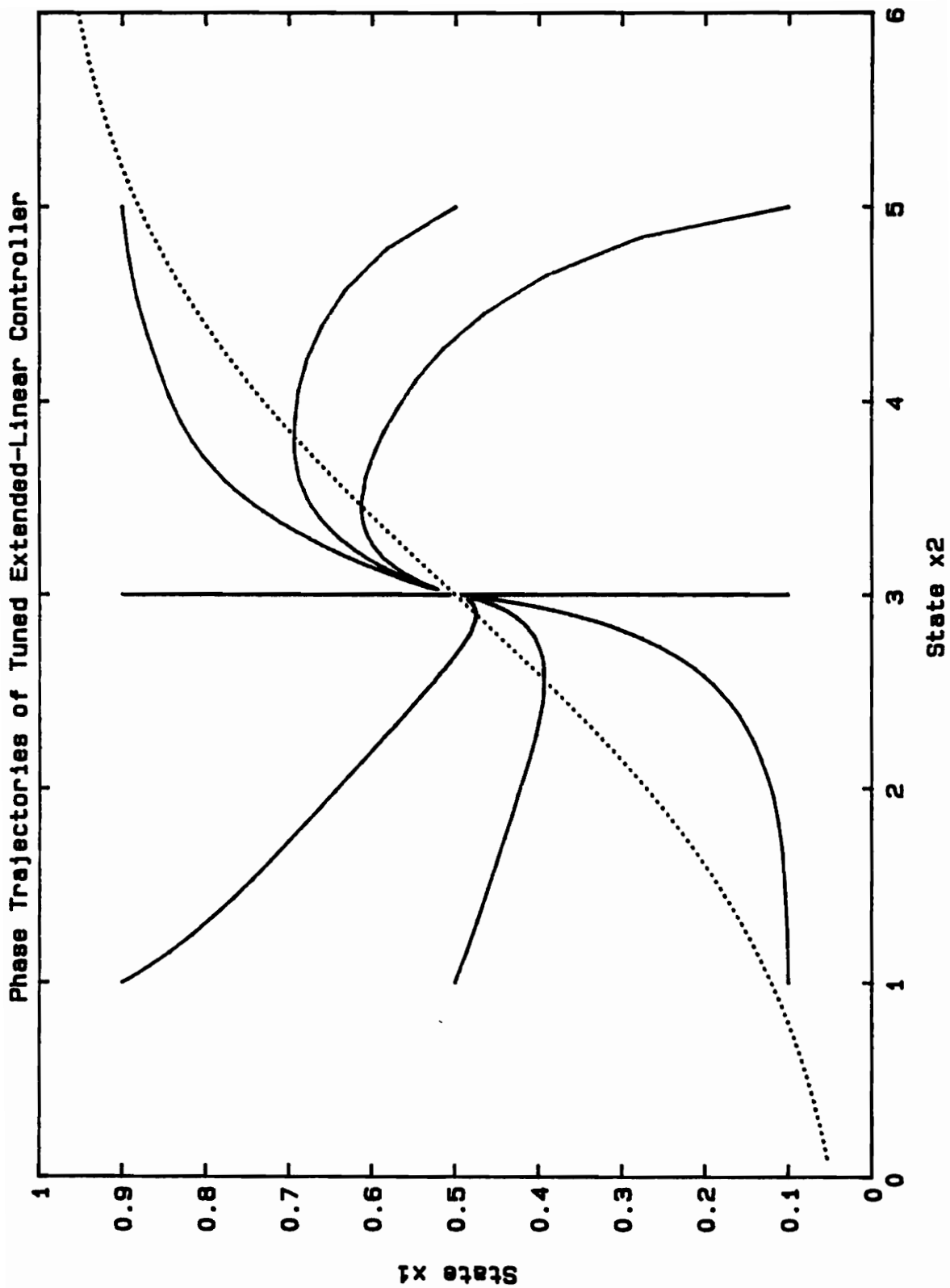


Figure 2.5

3.0 SINGLE EQUILIBRIUM POINT GLOBAL CONTROLLERS

As stated in the previous chapter, the extended-linearization controller provides excellent local performance, but the global performance remains unspecified. A system using the extended-linearization controller could have very poor global performance when the control input or states stray far from the equilibrium curve. To obtain good global performance and the same local control performance of a linear controller, a spline structure, which incorporates both the local and global control functions, is examined as a controller. The structure is defined only in the region of interest in the state-control space. Although this research examined several different approaches using splines, there were basically two prototypes before developing the final baseline approach. This chapter will first review cubic splines and then examine the prototypes to illustrate the problems that may occur in a spline based design. Next the baseline approach will be discussed. Also, for notational clarity, this chapter will deal only with two state variables. Additionally, the external control will be held constant which will result in only one equilibrium point. In the next chapter, the discussion will be extended to include a time-varying external control input.

3.1 Splines

Before proceeding into the development of the prototypes, a review of splines is in order. A spline is a function that performs piecewise polynomial interpolation. Lower-degree polynomials are joined together in a continuous fashion so that the resulting piecewise polynomial function interpolates the data set. The lowest degree piecewise polynomial interpolation is a “broken line” technique which is also known as linear interpolation. Although linear interpolation works in many cases, the technique lacks the smoothness that usually exists with physical processes. To overcome this lack of smoothness and to obtain better approximation of the data set, the technique of splines was developed. Cubic splines are a set of functions that interpolate the data and satisfy the following three properties over the interval $[a,b]$:

$$S(x) \in C^2[a,b]; \text{ that is, } S(x), S'(x), S''(x) \text{ are continuous on } [a,b]. \quad (3.1a)$$

$$S(x_k) = f(x_k) = f_k, \quad 1 \leq k \leq n; \text{ that is, } S(x) \text{ interpolates } f(x) \text{ on } [a,b]. \quad (3.1b)$$

$$S(x) \text{ is a cubic spline polynomial on each subinterval } [x_k, x_{k+1}], \quad 1 \leq k \leq n-1. \quad (3.1c)$$

Higher-order splines are defined similarly. For example, a quartic spline would be a function $S(x)$ where $S(x) \in C^3[a,b]$, and $S(x)$ is a fourth-degree polynomial on each subinterval $[x_k, x_{k+1}]$. Also note that $S(x)$ will be a *different* cubic on each subinterval; so to distinguish between the subintervals, $S_p(x)$ will denote the cubic polynomial such that $S(x) = S_p(x)$ for x in $[x_p, x_{p+1}]$, $1 \leq p \leq n-1$.

For example, consider a data set where $n = 4$. In this case the following ten equations must be satisfied

$$S_1(x_1) = f_1; \quad S_2(x_2) = f_2; \quad S_3(x_3) = f_3; \quad S_3(x_4) = f_4; \quad (3.2a)$$

$$S_1(x_2) = S_2(x_2); \quad S_2(x_3) = S_3(x_3); \quad (3.2b)$$

$$S'_1(x_2) = S'_2(x_2); \quad S'_2(x_3) = S'_3(x_3); \quad (3.2c)$$

$$S''_1(x_2) = S''_2(x_2); \quad S''_2(x_3) = S''_3(x_3); \quad (3.2d)$$

There are twelve unknowns associated with these equations, $\{f_1, f_2, f_3, f_4; f'_1, f'_2, f'_3, f'_4; f''_1, f''_2, f''_3, f''_4\}$. This system of equations is underdetermined which allows the solver to pick two more additional constraints. This fact will become important in the development of the prototypes. If the two additional constraints are such that the endpoints' second derivatives are set equal to zero, i.e. $S''_1(x_1) = 0$ and $S''_3(x_4) = 0$, then the splines are called natural cubic splines. Note that if n is increased, the number of unknowns is always two more than the number of equations.

Cubic spline interpolation can easily be extended into two dimensions by using cardinal cubic splines. Cardinal cubic splines are characterized by the property that on the knots x_1, x_2, \dots, x_n ; the spline function, $c_i(x_k)$, equals one when $i = k$ and zero otherwise, i.e. $c_i(x_k) = \delta_{ik}$ for $1 \leq i \leq n$. In two dimensions, cardinal cubic splines are used in the following fashion. On the knots (x_i, y_j) , there exists a single-variable cardinal spline $c_i(x)$ in the x direction and another, $c_j(y)$, in the y direction. The product of these cardinal cubic splines, $C_{ij}(x, y) = c_i(x) \cdot c_j(y)$, equals one at the specified grid point and zero at every other grid point. Consider a function such that $f(x_i, y_j) = f_{i,j}$ for $\{(x_i, y_j) \mid i=1, \dots, I; j=1, \dots, J\}$. This function can be approximated by the piecewise

polynomial spline function

$$f(x, y) \approx S(x, y) = \sum_{i=1}^I \sum_{j=1}^J f_{ij} \cdot C_{ij}(x, y)$$

or

$$f(x, y) \approx S(x, y) = \sum_{i=1}^I \sum_{j=1}^J f_{ij} \cdot c_i(x) \cdot c_j(y) \quad (3.3)$$

Cardinal splines can also be used to obtain higher-dimensional analogies of cubic splines.

Given $f(x, y, z)$, the cubic spline approximation function is

$$S(x, y, z) = \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K f_{ijk} \cdot c_i(x) \cdot c_j(y) \cdot c_k(z) \quad (3.4)$$

for the region gridded by $\{ (x_i, y_j, z_k) \mid i=1, \dots, I; j = 1, \dots, J; k = 1, \dots, K \}$.

3.2 First Prototype Controller

The first step in the control law design is to define the boundaries of the state-control space of interest and to determine the number of grid points needed. Only a coarse grid is acceptable since the application of even a coarse grid to several states represents a complicated controller. At each non-equilibrium grid point, a control value is calculated using some global criterion. In this thesis, the criterion will be to point the trajectory of the system toward the desired equilibrium point. At the equilibrium grid point, the necessary control to cause the system to remain stationary is calculated.

The resulting information, denoted by f_{ij} , is splined together to form a global controller.

If these calculated control values are simply splined together using standard techniques, such as natural cubic splines or basis splines, then the resulting control function might exhibit poor local performance about the equilibrium point. This is because the local response is determined by the derivative of the control law at the equilibrium point, and this is not specified if only natural cubic splines or basis splines are used. To avoid this problem, the spline functions should have the same partial derivatives at the equilibrium points as the desired local linear controllers. This will cause the global controller to have the desired local performance in some neighborhood of the equilibrium point.

The control values, f_{ij} , computed for each point on the grid $\{ (x_1^i, x_2^j) \mid i=1, \dots, I; j = 1, \dots, J \}$ are interpolated using piecewise cubic polynomials. These piecewise polynomials are formed by cardinal cubic splines. As stated in section 3.1, cardinal cubic splines are characterized by the property that on each knot, (x_1^i, x_2^j) , there exists a single-variable cardinal spline, $c_i(x_1)$, in the x_1 direction and another, $c_j(x_2)$, in the x_2 direction. The product of these cardinal cubic splines, $C_{ij}(x_1, x_2) = c_i(x_1) \cdot c_j(x_2)$, equals one at the specified grid point and zero at every other grid point. Note that $c_i(x_1)$ may be a *different* piecewise cubic polynomial in the x_1 direction for different values of "j". Also $c_j(x_2)$ may be a *different* piecewise cubic polynomial in the x_2 direction for different values of "i". To distinguish between these different polynomials, let $c_{ij}^1(x_1)$ denotes the cardinal cubic spline in the x_1 direction and let $c_{ij}^2(x_2)$ denote the cardinal cubic spline in the x_2 direction. The control law is constructed by multiplying the cardinal spline associated with each grid point by the desired control value for the grid point and summing over all the grid points

$$U(x_1, x_2) = \sum_{i=1}^I \sum_{j=1}^J f_{ij} \cdot c_{ij}^1(x_1) \cdot c_{ij}^2(x_2) \quad (3.5)$$

As stated in section 3.1, the spline technique does not uniquely determine a piecewise polynomial because the number of unknowns is two more than the number of equations for each direction. Additionally, the first prototype controller does not require that the cardinal cubic splines' second derivatives be equal at the equilibrium grid point and its adjacent grid points. This specification relaxes the condition stated in (3.1a). This will allow freedom in specifying the equilibrium grid point's partial derivatives. To get a unique representation for the control law and to incorporate the desired derivative information at the equilibrium point, more conditions need to be specified. Assume the equilibrium point occurs at the grid point with index (i_e, j_e) . If $j \neq j_e$, then $c_{ij}^1(x_1)$ is taken to be a natural cubic spline. If $j = j_e$, and the desired derivative in the x_1 direction at the equilibrium point is D_{x_1} (the gain of the desired linear feedback associated with x_1), then $c_{i j_e}^1(x_1)$ is taken to be the unique piecewise cubic polynomial that satisfies

$$c_{i j_e}^1(x_1^k) = \delta_{ik} \quad (3.6)$$

$$\frac{\partial c_{i j_e}^1(x_1^k)}{\partial x_1} = \frac{D_{x_1}}{h_{i_e j_e}} \delta_{i i_e} \delta_{ik} \quad \text{for } \{ i = i_e - 1, i_e, \text{ or } i_e + 1 \} \quad (3.7)$$

$$c_{i j_e}^1(x_1^k) \triangleq \text{natural cubic splines} \quad \text{for } \{ i \neq i_e - 1, i_e, \text{ or } i_e + 1 \} \quad (3.8)$$

for $1 \leq k \leq I$. Refer to figure 3.3 for samples of cardinal cubic splines. The $c_{ij}^2(x_2)$ are defined in a similar fashion for $1 \leq k \leq J$.

Returning to the continuously stirred tank reactor example, the following notational state changes will alleviate notational ambiguity

$$x = x_1 \quad (3.9)$$

$$y = x_2 \quad (3.10)$$

The region of interest is restricted to $\{ (x, y) \mid 0.1 \leq x \leq 0.9 ; 1.0 \leq y \leq 5.0 \}$. The region is gridded by $x_1=0.1, x_2=0.5, x_3=0.9$; and $y_1=1.0, y_2=3.0, y_3=5.0$ where $I=3$ and $J=3$. Refer to figure 3.1. The desired equilibrium point was chosen to be $X^o = [0.5 \ 3.0]^T$. At the exterior grid points, the control necessary to point the trajectory of the system toward the equilibrium point were calculated. These control values represent the constants, $f_{i,j}$, in equation (3.5), and the relevant global information about the system. Refer to figure 3.2.

For each subinterval in the x-direction, the form of the cardinal cubic spline is given by

$$c_{i,j}^1(x) = ax^3 + bx^2 + cx + d \quad (3.11)$$

When $j \neq 2$, the standard technique for solving natural cubic splines is used to solve for $\{ a, b, c, d \}$.

When $j=2$, the coefficients are determined by solving the following linear equations

for $i = 1$ and $0.1 \leq x < 0.5$:

$$c_{12}^1(x_1) = ax_1^3 + bx_1^2 + cx_1 + d = 1 \quad (3.12)$$

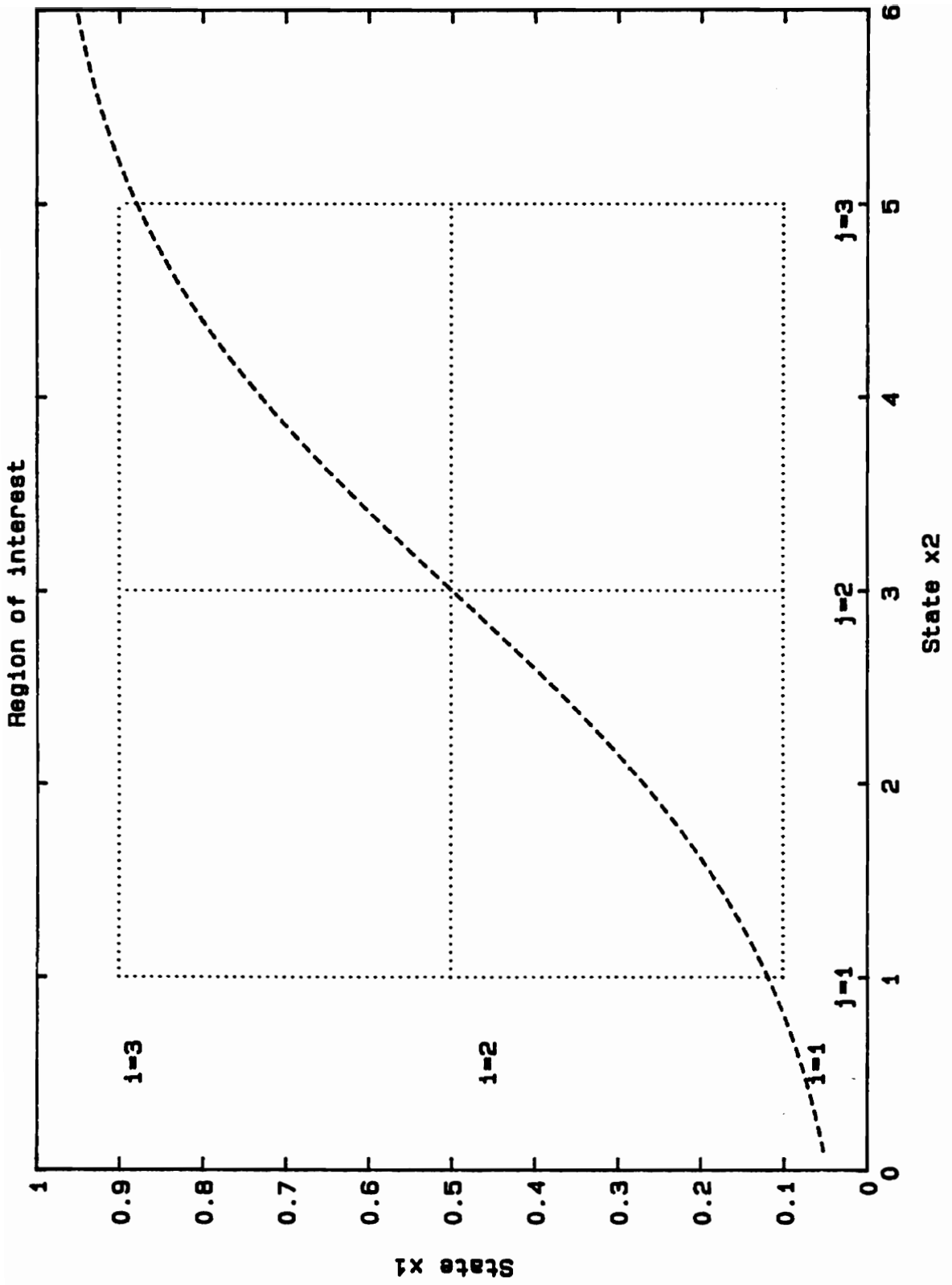


Figure 3.1

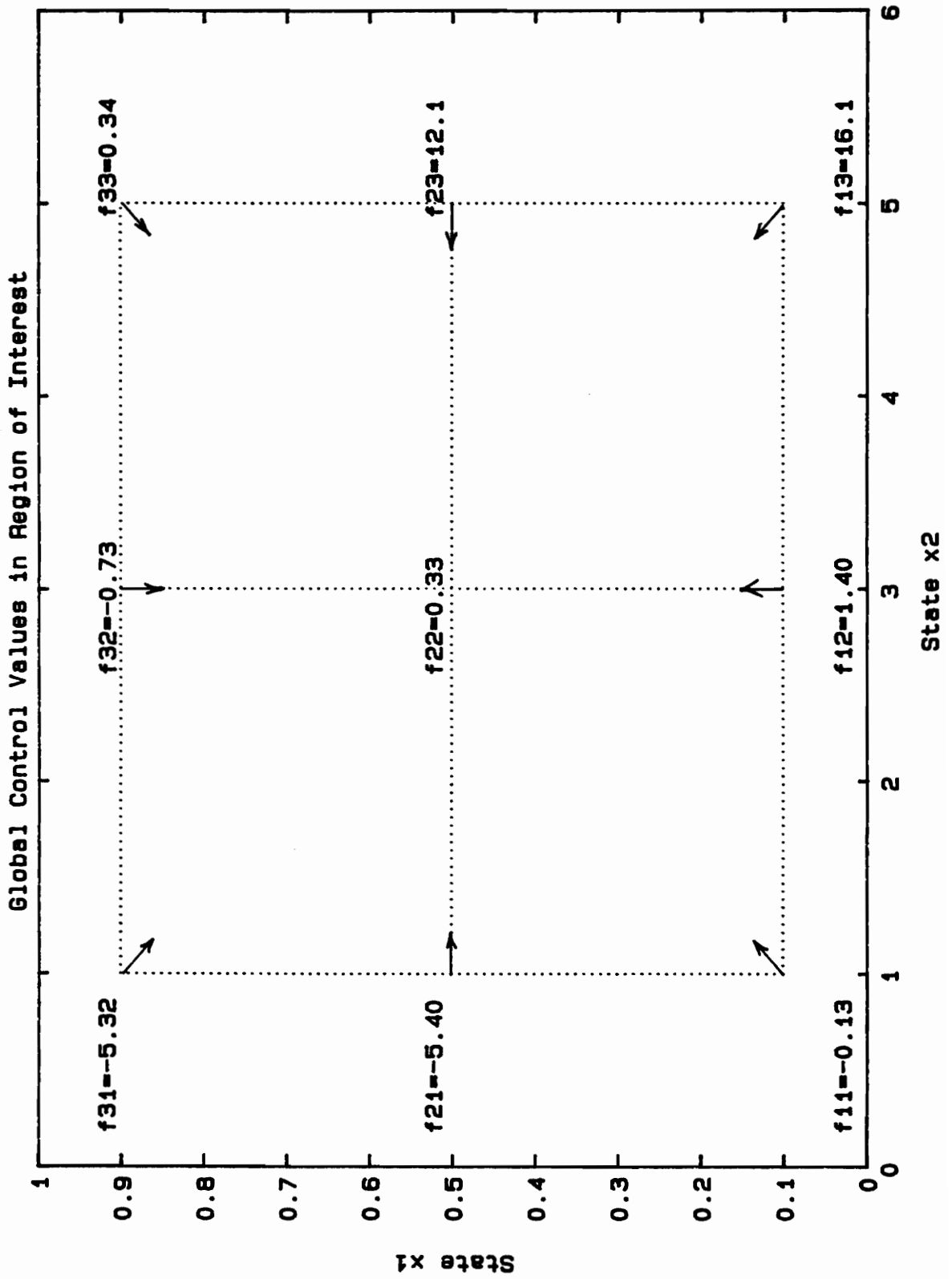


Figure 3.2

$$c_{12}^1(x_2) = ax_2^3 + bx_2^2 + cx_2 + d = 0 \quad (3.13)$$

$$\frac{dc_{12}^1(x_1)}{dx} = 3 \cdot ax_1^2 + 2 \cdot bx_1 + c = 0 \quad (3.14)$$

$$\frac{dc_{12}^1(x_2)}{dx} = 3 \cdot ax_2^2 + 2 \cdot bx_2 + c = 0 \quad (3.15)$$

for $i = 1$ and $0.5 \leq x \leq 0.9$: $c_{12}^1(x) = 0 \quad (3.16)$

for $i = 2$ and $0.1 \leq x < 0.5$:

$$c_{22}^1(x_1) = ax_1^3 + bx_1^2 + cx_1 + d = 0 \quad (3.17)$$

$$c_{22}^1(x_2) = ax_2^3 + bx_2^2 + cx_2 + d = 1 \quad (3.18)$$

$$\frac{dc_{22}^1(x_1)}{dx} = 3 \cdot ax_1^2 + 2 \cdot bx_1 + c = 0 \quad (3.19)$$

$$\frac{dc_{22}^1(x_2)}{dx} = 3 \cdot ax_2^2 + 2 \cdot bx_2 + c = \frac{D_{x_1}}{f_{22}} \quad (3.20)$$

for $i = 2$ and $0.5 \leq x \leq 0.9$:

$$c_{22}^1(x_2) = ax_2^3 + bx_2^2 + cx_2 + d = 1 \quad (3.21)$$

$$c_{22}^1(x_3) = ax_3^3 + bx_3^2 + cx_3 + d = 0 \quad (3.22)$$

$$\frac{dc_{22}^1(x_2)}{dx} = 3 \cdot ax_2^2 + 2 \cdot bx_2 + c = \frac{D_{x_1}}{f_{22}} \quad (3.23)$$

$$\frac{dc_{22}^1(x_3)}{dx} = 3 \cdot ax_3^2 + 2 \cdot bx_3 + c = 0 \quad (3.24)$$

for $i = 3$ and $0.1 \leq x < 0.5$:

$$c_{32}^1(x) = 0 \quad (3.25)$$

for $i = 3$ and $0.5 \leq x \leq 0.9$:

$$c_{32}^1(x_2) = ax_2^3 + bx_2^2 + cx_2 + d = 0 \quad (3.26)$$

$$c_{32}^1(x_3) = ax_3^3 + bx_3^2 + cx_3 + d = 1 \quad (3.27)$$

$$\frac{dc_{32}^1(x_2)}{dx} = 3 \cdot ax_2^2 + 2 \cdot bx_2 + c = 0 \quad (3.28)$$

$$\frac{dc_{32}^1(x_3)}{dx} = 3 \cdot ax_3^2 + 2 \cdot bx_3 + c = 0 \quad (3.29)$$

As stated before, $x_1 = 0.1$, $x_2 = 0.5$ and $x_3 = 0.9$ in all the above equations. Additionally, $D_{x_1} = 2.667$ and $f_{22} = 0.3333$. Intuitively, given two points and a derivative at each point, a cardinal cubic spline always exists that interpolates the data. The above cardinal cubic splines are shown in figure 3.3.

When the control is limited to a value of less than ten, then the first prototype controller obtains better phase trajectories with less overall control than the linear controller. Note that all global information is first incorporated into the controller before the controller is limited to a value of less than ten. Future controllers discussed in this thesis will also be limited to ten and will exhibit the same property. Refer to figures 3.19 through 3.21 for a comparison of the linear controller versus the S.E.P. baseline controller.

The phase trajectories resulting from the first prototype design are shown in figure 3.4. The trajectories associated with the initial conditions around $X = [0.1 \ 5.0]^T$ have much better global performance than the linear controller and a moderate improvement over the tuned extended-linear controller. However, the first prototype controller fails to give the desired local performance. Although there does exist a region of good local performance around the equilibrium point, the region is so small that the impact the region has on the system's performance is negligible. An explanation for the controller's failure is given by figure 3.5. For a global controller to have the desired local performance of a linear controller, the partials of the control function with respect to the state variables,

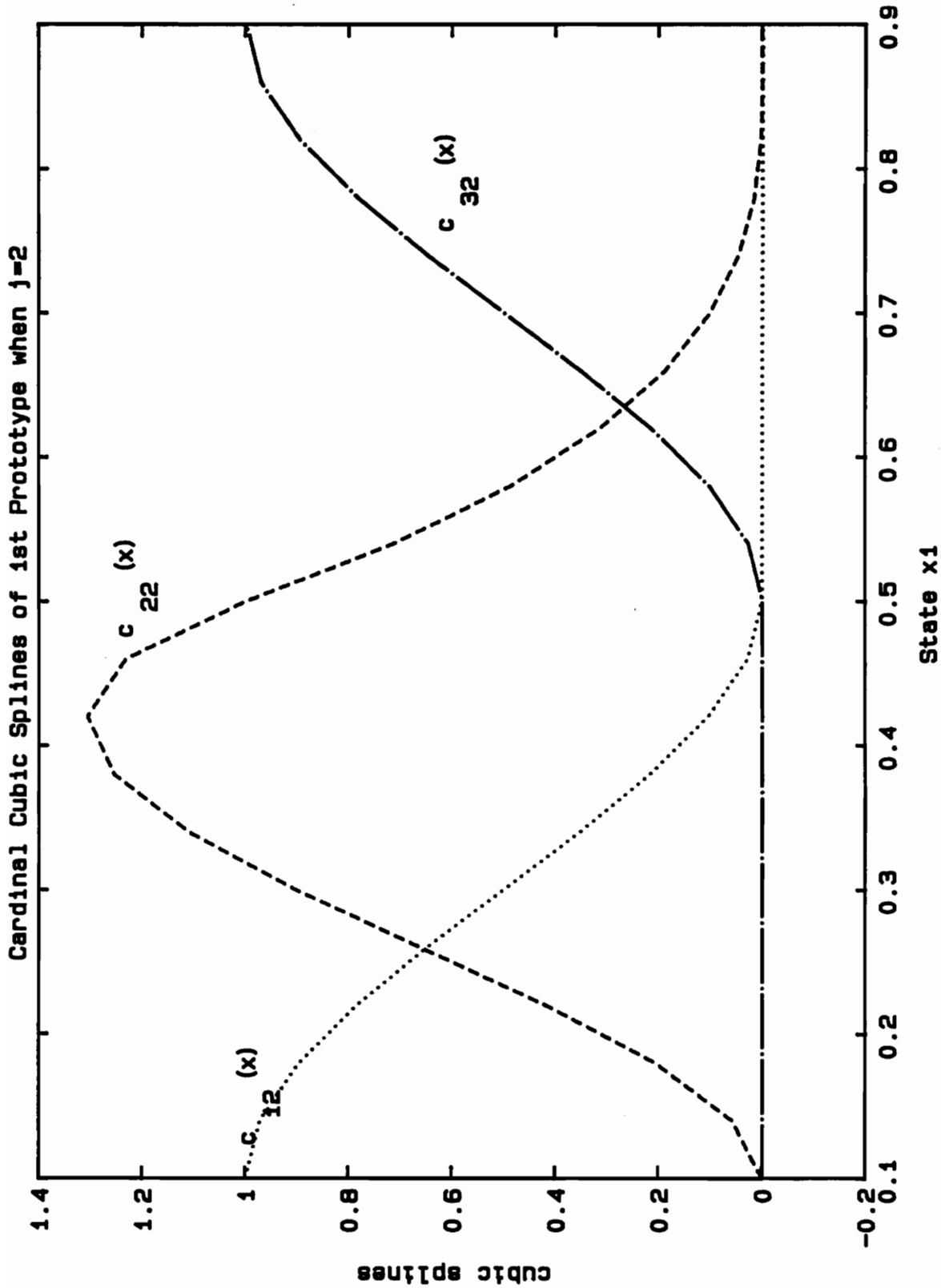


Figure 3.3

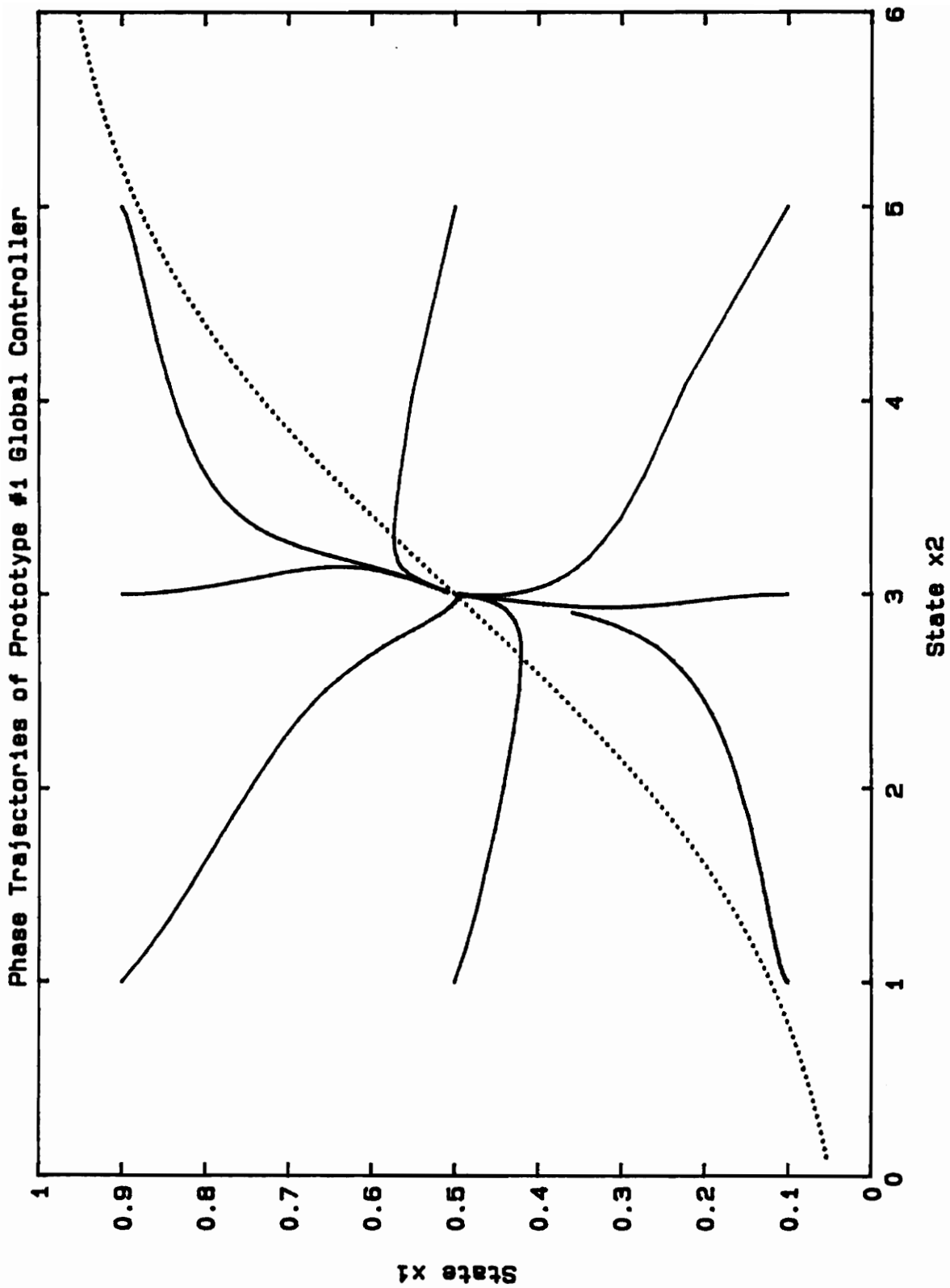


Figure 3.4

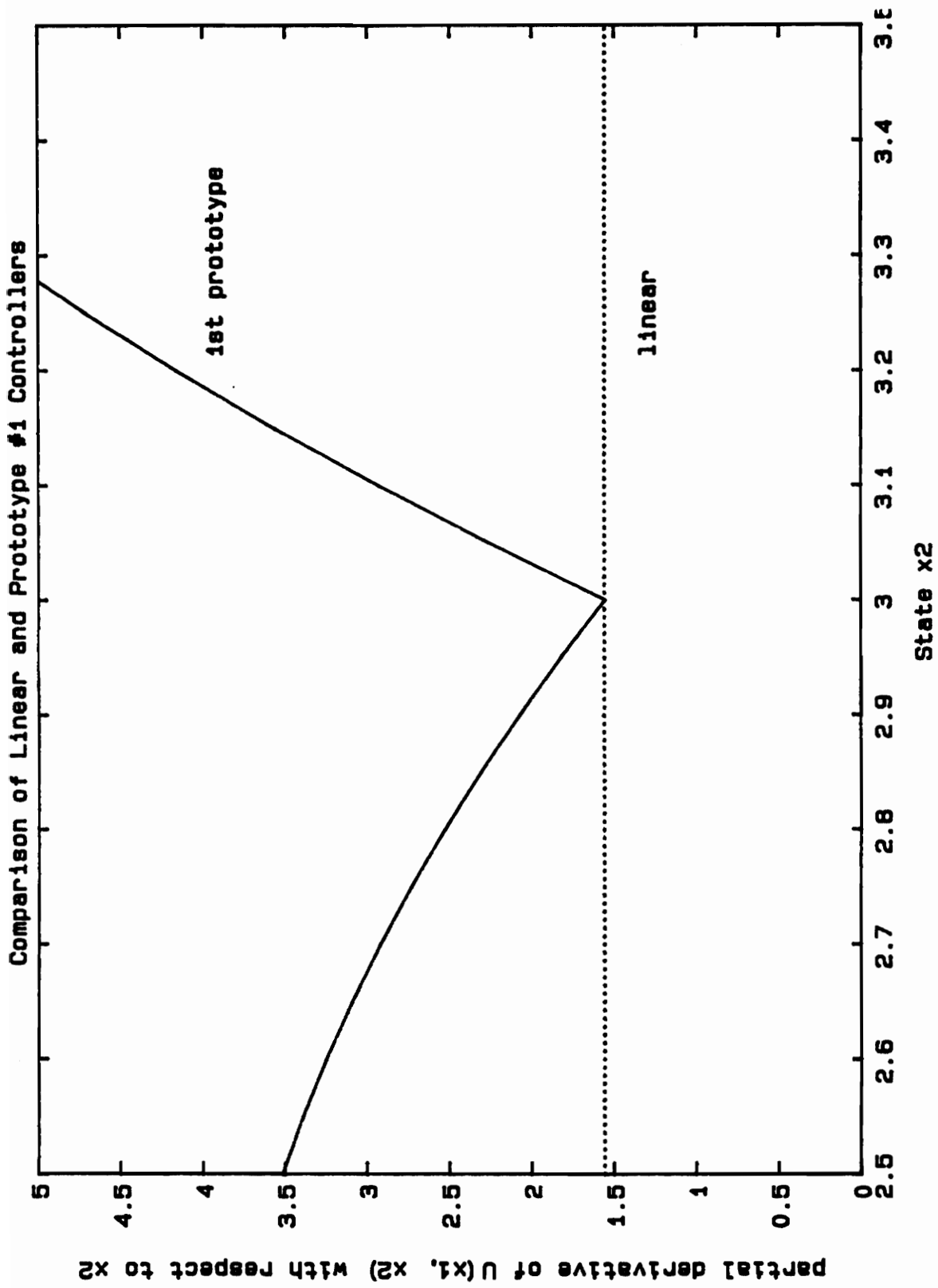


Figure 3.5

$\partial U(x_1, x_2)/\partial x_1$ and $\partial U(x_1, x_2)/\partial x_2$, must match their respective linear feedback gains, G_1 and G_2 . As seen in figure 3.5, the first prototype matches the x_2 feedback gain with the desired partial only at the equilibrium point, and any significant distance away from the equilibrium point, the two values are very different. Therefore the first prototype controller has good local performance only in a unsatisfactory small region of the state space about the equilibrium point.

3.2 Second Prototype Controller

The first prototype shows excellent global performance, and does show the desired local linear performance. However, the local linear performance is apparent in only a tiny neighborhood of the state-control space near the equilibrium point. In fact the first prototype shows no significant improvement over the “straight” interpolative global controller.

The second prototype was developed to improve the first prototype’s local performance. To keep $\partial U(x_1, x_2)/\partial x_1$ and $\partial U(x_1, x_2)/\partial x_2$ from changing over a significant distance from the equilibrium point, the second prototype specifies the second partial derivatives, $\partial^2 U(x_1, x_2)/\partial x_1^2$ and $\partial^2 U(x_1, x_2)/\partial x_2^2$, to be zero at the desired equilibrium grid point. Additionally in contrast to the first prototype controller, the second prototype controller does not specify $\partial U(x_1, x_2)/\partial x_1$ and $\partial U(x_1, x_2)/\partial x_2$ to be zero at grid points adjacent to the equilibrium grid point. These conditions give a unique representation for the second prototype controller.

Again, assume the equilibrium point occurs at the grid point with index (i_e, j_e) . If $j \neq j_e$, then $c_{ij}^1(x_1)$ is taken to be a natural cubic spline. If $j = j_e$, and the desired derivative in the x_1 direction

at the equilibrium point is D_{x_1} (the gain of the desired linear feedback associated with x_1), then $c_{i_j e}^1(x_1)$ is taken to be the unique piecewise cubic polynomial that satisfies

$$c_{i_j e}^1(x_1^k) = \delta_{ik} \quad (3.30)$$

$$\frac{\partial c_{i_j e}^1(x_1^k)}{\partial x_1} = \frac{D_{x_1}}{f_{i_e j e}} \delta_{i i_e} \quad \text{for } \{ i = i_e - 1, i_e, \text{ or } i_e + 1 \} \text{ and } \{ k = i_e \} \quad (3.31)$$

$$\frac{\partial^2 c_{i_j e}^1(x_1^k)}{\partial x_1^2} = 0 \quad \text{for } \{ i = i_e - 1, i_e, \text{ or } i_e + 1 \} \text{ and } \{ k = i_e \} \quad (3.32)$$

$$c_{i_j e}^1(x_1^k) \triangleq \text{natural cubic splines for } \{ i \neq i_e - 1, i_e, \text{ or } i_e + 1 \} \quad (3.33)$$

for $1 \leq k \leq I$. The $c_{i_j}^2(x_2)$ are defined in a similar fashion for $1 \leq k \leq J$.

Returning to the continuously stirred tank reactor example, assume again that $x = x_1$ and $y = x_2$. For each subinterval in the x -direction, the form of the cardinal cubic spline is given by

$$c_{i_j}^1(x) = ax^3 + bx^2 + cx + d \quad (3.34)$$

When $j \neq 2$, the standard technique for solving natural cubic splines is used to solve for $\{ a, b, c, d \}$.

When $j = 2$, then the coefficients are determined by solving the following set of linear equations

for $i = 1$ and $0.1 \leq x < 0.5$:

$$c_{12}^1(x_1) = ax_1^3 + bx_1^2 + cx_1 + d = 1 \quad (3.35)$$

$$c_{12}^1(x_2) = ax_2^3 + bx_2^2 + cx_2 + d = 0 \quad (3.36)$$

$$\frac{dc_{12}^1(x_2)}{dx} = 3 \cdot ax_2^2 + 2 \cdot bx_2 + c = 0 \quad (3.37)$$

$$\frac{d^2 c_{12}^1(x_2)}{dx^2} = 6 \cdot ax_2 + 2 \cdot b = 0 \quad (3.38)$$

for $i = 1$ and $0.5 \leq x \leq 0.9$: $c_{12}^1(x) = 0 \quad (3.39)$

for $i = 2$ and $0.1 \leq x < 0.5$:

$$c_{22}^1(x_1) = ax_1^3 + bx_1^2 + cx_1 + d = 0 \quad (3.40)$$

$$c_{22}^1(x_2) = ax_2^3 + bx_2^2 + cx_2 + d = 1 \quad (3.41)$$

$$\frac{dc_{22}^1(x_2)}{dx} = 3 \cdot ax_2^2 + 2 \cdot bx_2 + c = \frac{D_{x_1}}{f_{22}} \quad (3.42)$$

$$\frac{d^2}{dx^2} c_{22}^1(x_2) = 6 \cdot ax_2 + 2 \cdot b = 0 \quad (3.43)$$

for $i = 2$ and $0.5 \leq x \leq 0.9$:

$$c_{22}^1(x_2) = ax_2^3 + bx_2^2 + cx_2 + d = 1 \quad (3.44)$$

$$c_{22}^1(x_3) = ax_3^3 + bx_3^2 + cx_3 + d = 0 \quad (3.45)$$

$$\frac{dc_{22}^1(x_2)}{dx} = 3 \cdot ax_2^2 + 2 \cdot bx_2 + c = \frac{D_{x_1}}{f_{22}} \quad (3.46)$$

$$\frac{d^2}{dx^2} c_{22}^1(x_2) = 6 \cdot ax_2 + 2 \cdot b = 0 \quad (3.47)$$

for $i = 3$ and $0.1 \leq x < 0.5$: $c_{32}^1(x) = 0 \quad (3.48)$

for $i = 3$ and $0.5 \leq x \leq 0.9$:

$$c_{32}^1(x_2) = ax_2^3 + bx_2^2 + cx_2 + d = 0 \quad (3.49)$$

$$c_{32}^1(x_3) = ax_3^3 + bx_3^2 + cx_3 + d = 1 \quad (3.50)$$

$$\frac{d^2 c_{32}^1(x_2)}{dx^2} = 6 \cdot ax_2 + 2 \cdot b = 0 \quad (3.51)$$

$$\frac{dc_{32}^1(x_2)}{dx} = 3 \cdot ax_2^2 + 2 \cdot bx_2 + c = 0 \quad (3.52)$$

As stated before, $x_1 = 0.1$, $x_2 = 0.5$ and $x_3 = 0.9$ in all the above equations. Additionally, $D_{x_1} = 2.667$ and $f_{22} = 0.3333$. These cardinal cubic splines are given in figure 3.6.

The phase trajectories of the resulting design are given in figure 3.7. As with the first prototype, the trajectories associated with the initial conditions around $\mathbf{X} = [0.1 \ 5.0]^T$ have much better global performance than the linear controller and a moderate improvement over the tuned extended-linear controller. Additionally, the second prototype controller improves on the first prototype's local performance, but the second prototype still fails to give the desired local performance. Although the region of local performance has been greatly extended over the first prototype's region, the local region is still too small to significantly alter the system's performance near the equilibrium point. An explanation for the controller's failure is given by figure 3.8. The partial of the control function with respect to the state variable x_2 , $\partial U(x_1, x_2)/\partial x_2$, fails to match the respective feedback gain of the linear system, G_2 , over a significant distance away from the equilibrium point. Also figure 3.9 shows that the partial of the control function with respect to the state variable x_1 , $\partial U(x_1, x_2)/\partial x_1$, plotted over the same domain matches the respective feedback gain, G_1 , only at the

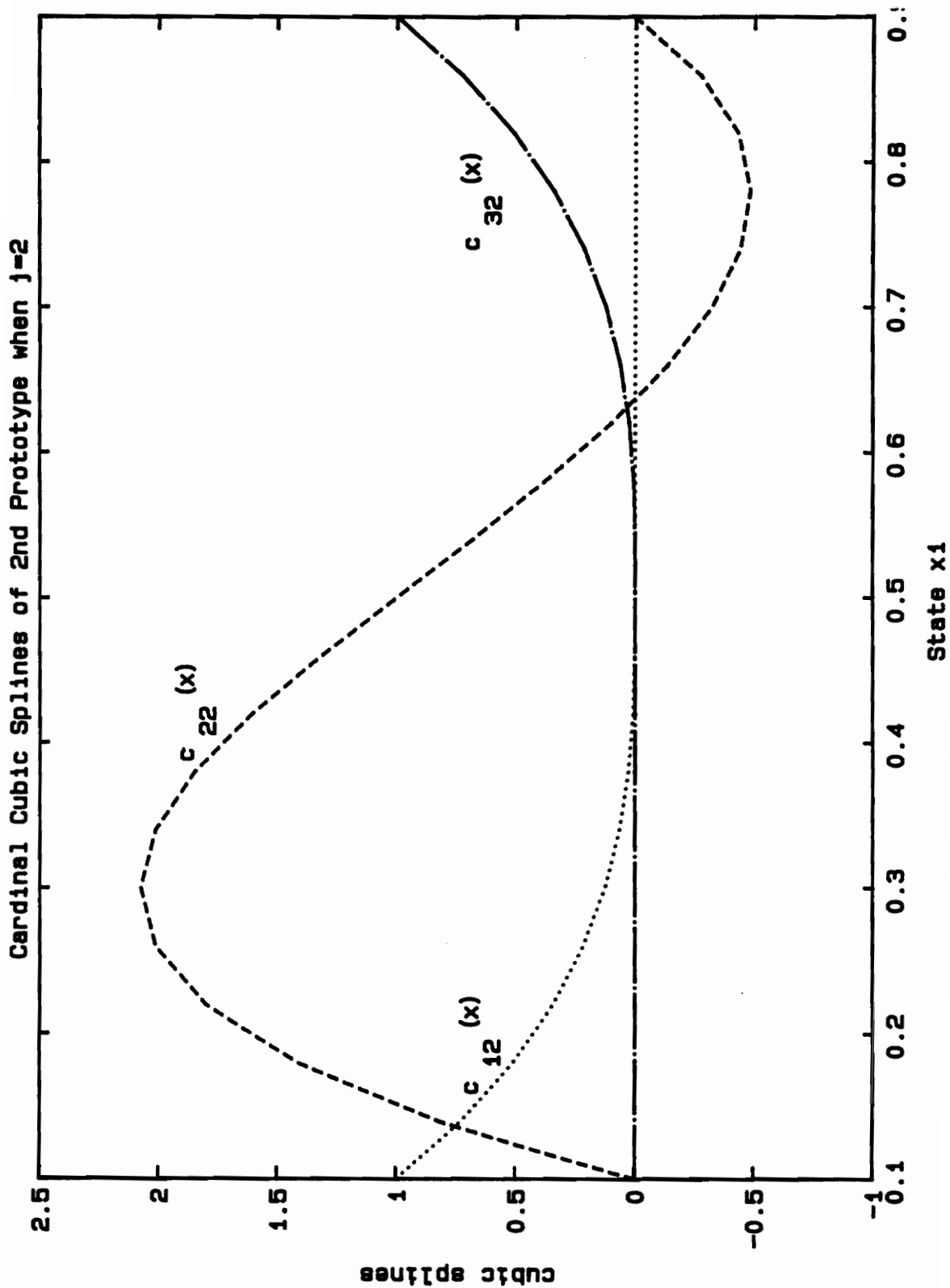


Figure 3.6

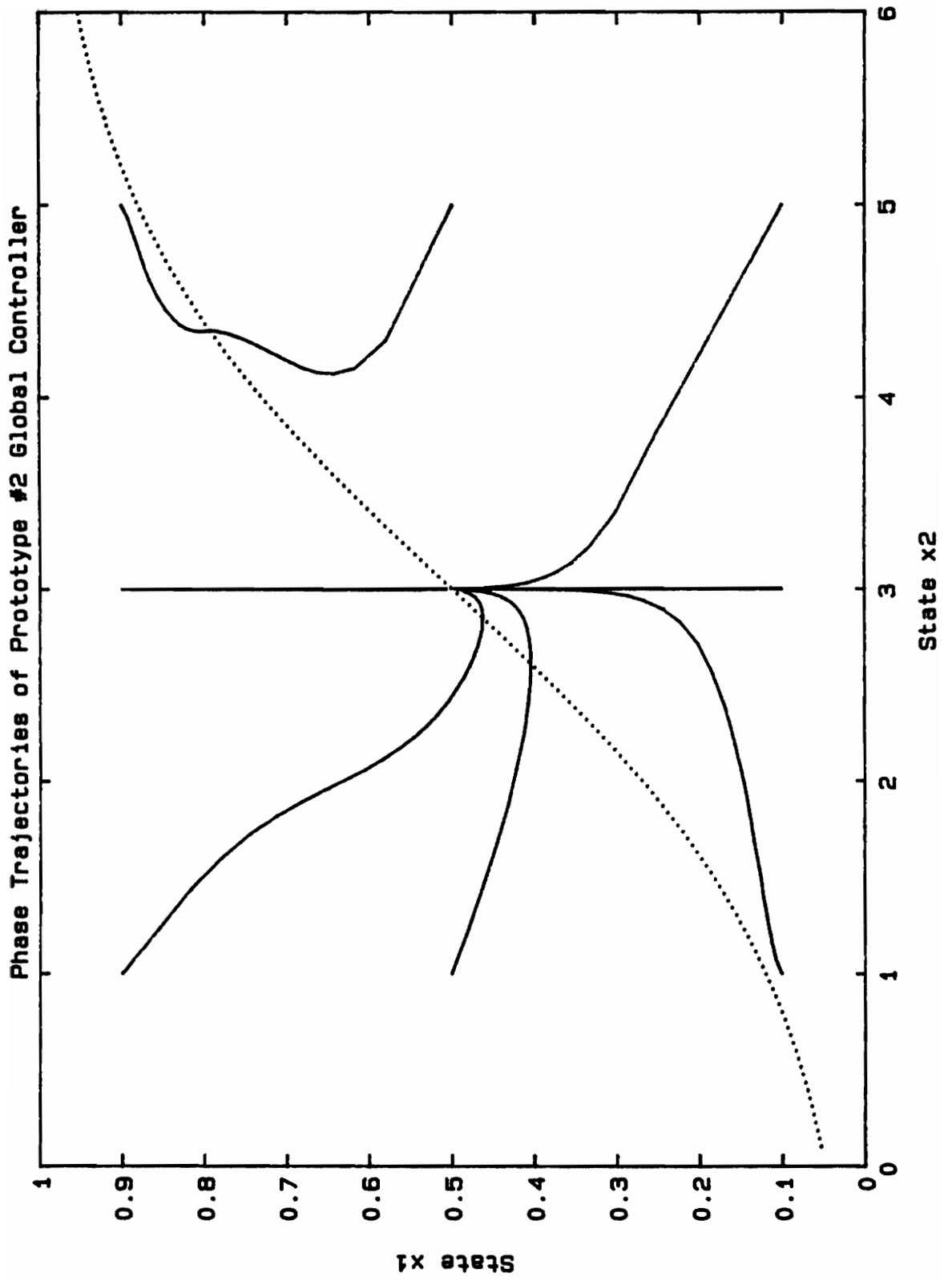


Figure 3.7

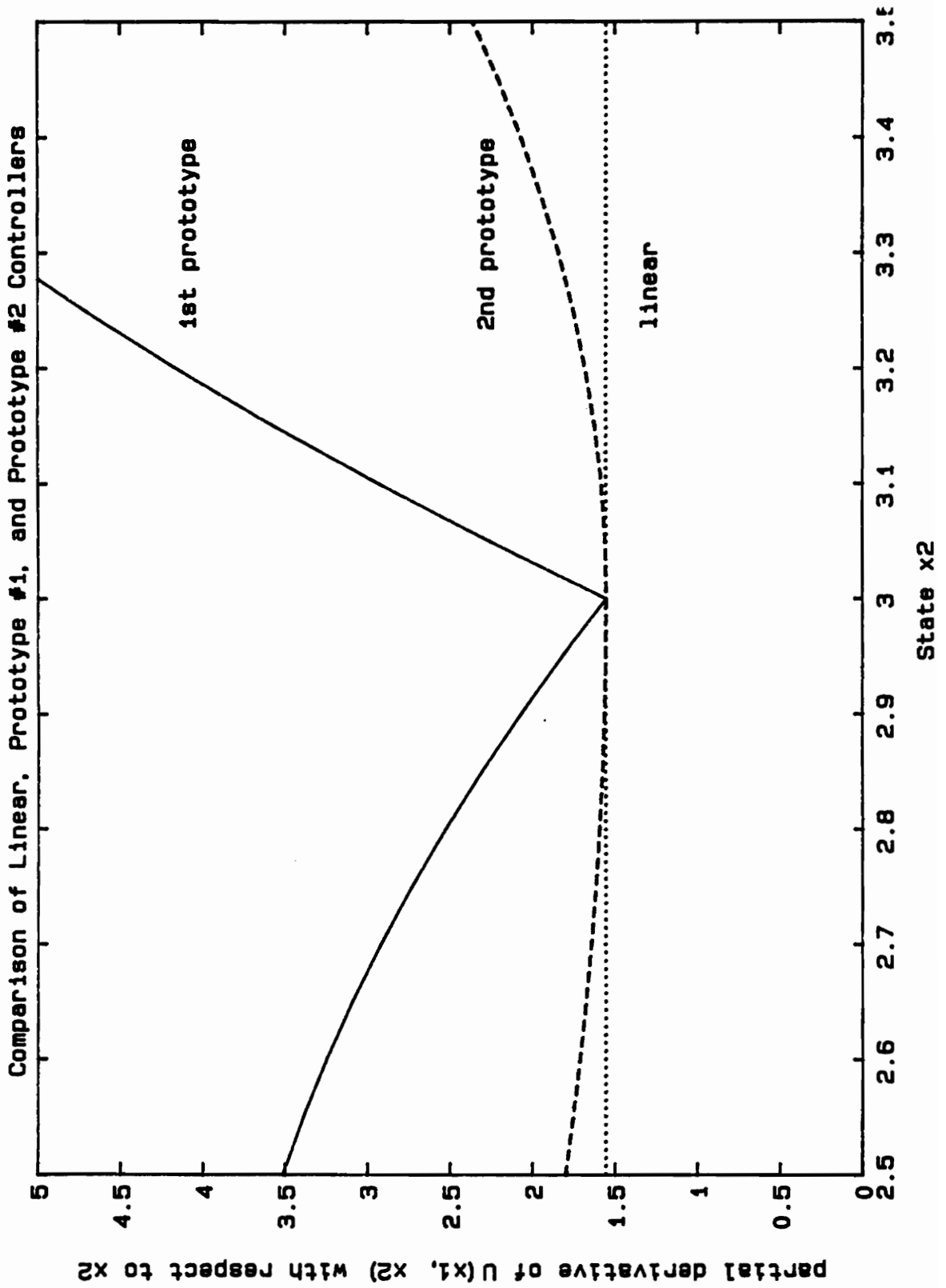


Figure 3.8

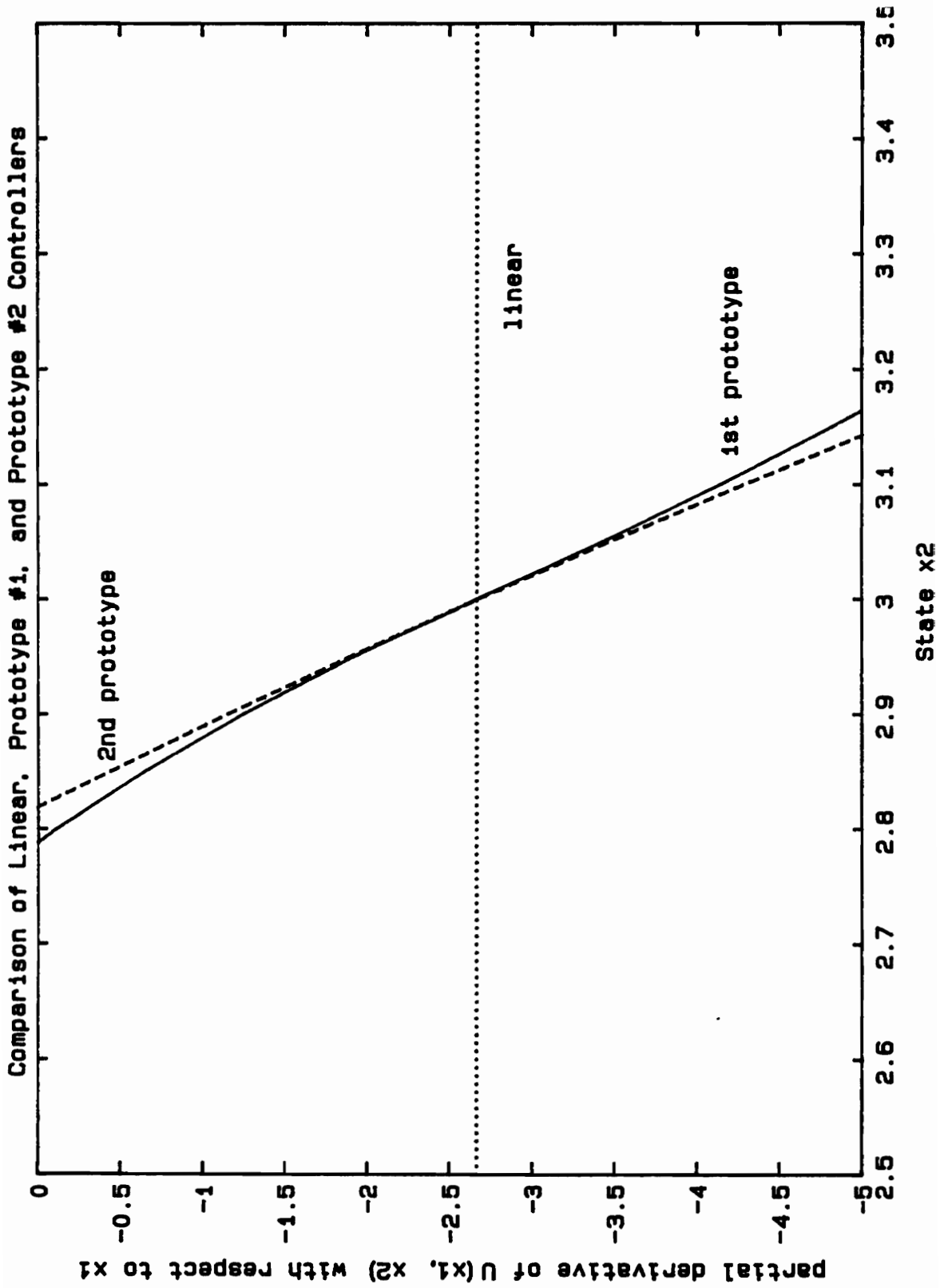


Figure 3.9

equilibrium point.

Examination of the phase trajectories reveals that a spurious equilibrium point exists at $\mathbf{X}^o = [0.88 \ 4.32]^T$. This problem is not unique for the second prototype's design procedure, but it is equally possible for the linear controller and the first prototype controller already discussed. Whenever the closed-loop control, $\mathbf{S}(x_1^o(\alpha), \alpha)$, equals the desired equilibrium control, $u^o(\alpha)$, another equilibrium point will occur. For the continuously stirred tank reactor, figure 3.10 shows the $u^o(\alpha)$ curve, where $\alpha = x_2$. Figure 3.11 shows the $u^o(\alpha)$ curve and three different $\mathbf{S}(x_1^o(\alpha), \alpha)$ curves as functions of x_2 . When $f_{33} = 0.34$, the closed loop control equals the desired equilibrium control at $x_2 = 4.32$ which results in the spurious equilibrium point. If f_{33} is increased from a value of 0.34 to 5.0, then the right portion of the $\mathbf{S}(x_1^o(\alpha), \alpha)$ curve is increase slightly above the $u^o(\alpha)$ curve pushing the spurious equilibrium point to $\mathbf{X} = [0.7 \ 3.8]^T$. When f_{33} is increased to a value of 10.0, then the spurious equilibrium point is removed entirely. This procedure represents a global "fix-up" to remove unwanted equilibrium points.

3.4 S.E.P. Baseline Controller

Next the baseline approach will be discussed. Although the first two methods show the desired local performance specified, this performance extends over a very unsatisfactory small region close to the equilibrium point. The second prototype extended this linear region, but only by a small amount. The next approach yields a satisfactory linear region and still gives the global performance desired.

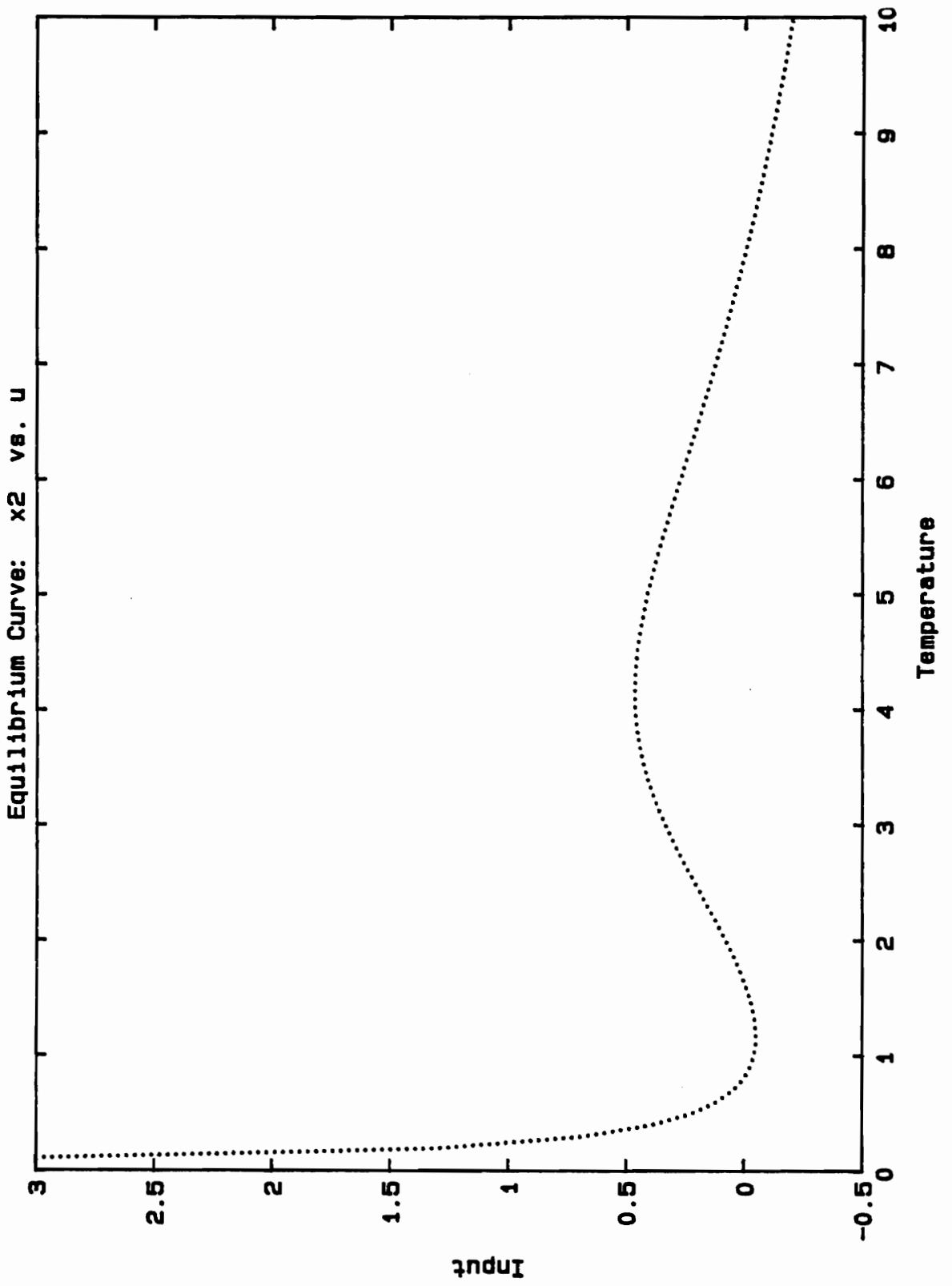


Figure 3.10

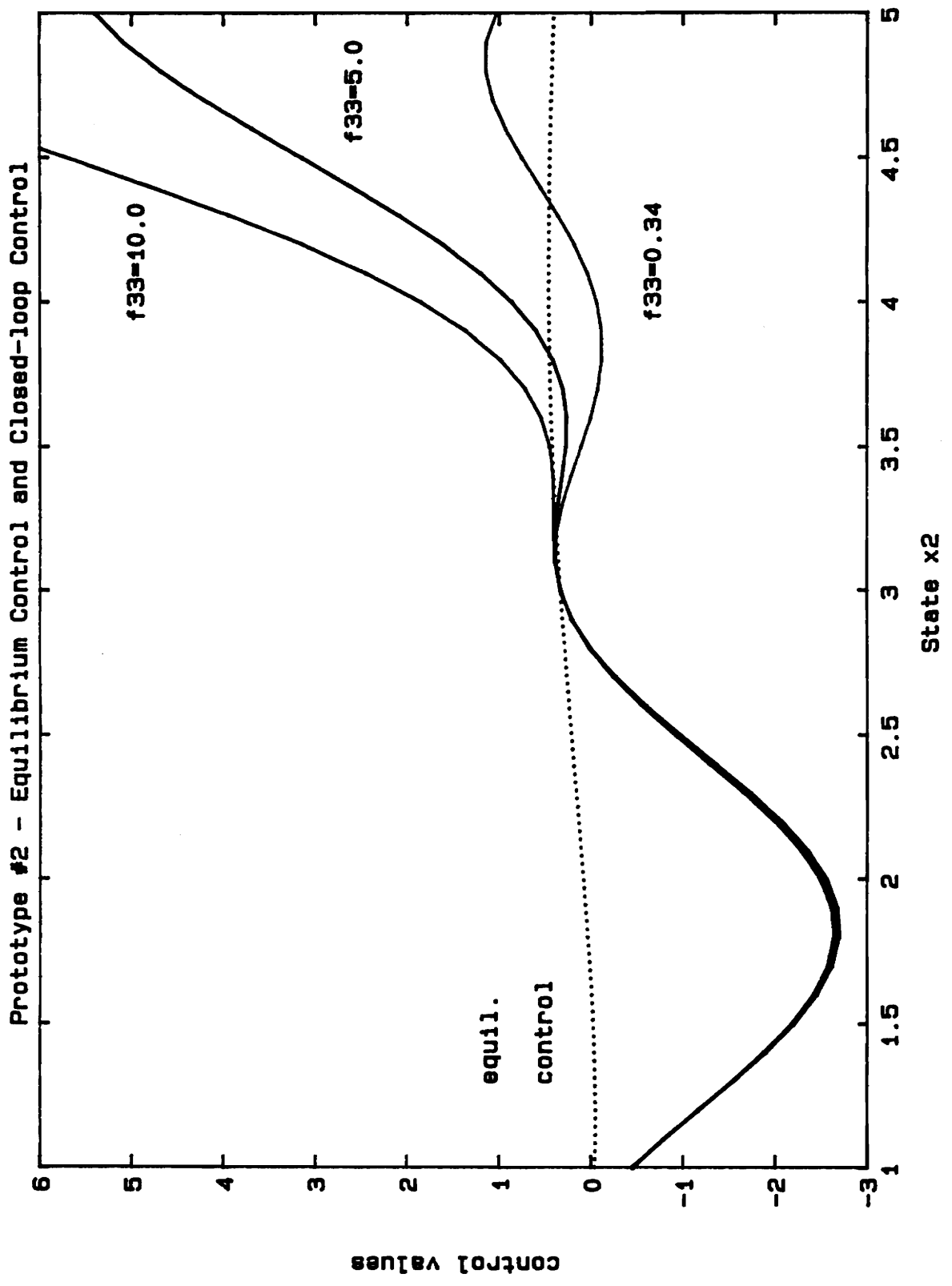


Figure 3.11

The idea is to start with the structure of a linear controller and spline global information to the structure. At each grid point, the control value of the linear controller, denoted l_{ij} , is calculated. These control values are subtracted from the global values, f_{ij} , in equation (3.5). The resulting values, $b_{ij} = f_{ij} - l_{ij}$, are then splined together. Note that at the equilibrium point, the resulting value is zero, i.e. $b_{ieje} = 0$. The spline structure is added to the linear controller where the sum is still a spline structure. The overall nonlinear controller is

$$U(x_1, x_2) = w - G_1 \cdot x_1 - G_2 \cdot x_2 + U_g(x_1, x_2) \quad (3.53)$$

where

$$U_g(x_1, x_2) = \left[\sum_{i=1}^I \sum_{j=1}^J b_{ij} \cdot c_{ij}^1(x_1) \cdot c_{ij}^2(x_2) \right] \quad (3.54)$$

An important difference exists between the baseline controller and the prototype controllers. The baseline controller, although still a cubic spline structure, can not be obtained using the cardinal spline technique previously discussed. This is because a straight linear controller is not possible to obtain using the cardinal spline decomposition. Although the linear controller is a cubic spline structure, albeit with many coefficients equal to zero, the following condition exists

$$\frac{\partial^2 U(x_1, x_2)}{\partial x_1 \partial x_2} = 0 \quad \forall (x_1, x_2) \quad (3.55)$$

Condition (3.55) is not possible using the cardinal spline technique except in the most trivial case. For

the cardinal cubic spline technique, condition (3.55) can be written as

$$\frac{\partial^2 U(x_1, x_2)}{\partial x_1 \partial x_2} = \sum_{i=1}^I \sum_{j=1}^J f_{ij} \cdot \frac{d c_{ij}^1(x_1)}{dx_1} \cdot \frac{d c_{ij}^2(x_2)}{dx_2} = 0 \quad (3.56)$$

Equation (3.56) does not apply for all (x_1, x_2) and specifically does not apply at the equilibrium grid point. Therefore, condition (3.55) can not be satisfied using the cardinal spline technique.

Again $c_{ij}^1(x_1)$ and $c_{ij}^2(x_2)$ do not uniquely determine a control law. To uniquely determine a control law, the cardinal cubic splines are determined using the same techniques as the second prototype except for two major difference. First, the local information in the S.E.P. baseline controller is established by the linear control information; and, therefore, the global information should not contain any local information. This implies that the partial derivatives with respect to the state variables of the global control information, $\partial U_g(x_1, x_2)/\partial x_1$ and $\partial U_g(x_1, x_2)/\partial x_2$, must be zero at the equilibrium points. The second major difference is that the second partial derivatives are used to minimize the effect of a grid point on other grid points adjacent to them. Again assume the equilibrium point occurs at the grid point with index (i_e, j_e) , then $c_{ij}^1(x_1)$ is taken to be the unique piecewise cubic polynomial that satisfies

$$c_{ij}^1(x_1^k) = \delta_{ik} \quad (3.57)$$

$$\frac{\partial c_{ij}^1(x_1^k)}{\partial x_1} = 0 \quad \text{for } \{ i = i_e - 1 \text{ or } i_e + 1 \} \text{ and } \{ k = i_e \} \quad (3.58)$$

$$\frac{\partial^2 c_{ij}^1(x_1^k)}{\partial x_1^2} = 0 \quad \text{for } \{ i = i_e - 1 \text{ or } i_e + 1 \} \text{ and } \{ k = i_e \} \quad (3.59)$$

$$\frac{\partial c_{ij}^1(x_1^k)}{\partial x_1} = 0 \quad \text{for } \{ i = i_e \} \text{ and } \{ k = i_e - 1 \text{ or } i_e + 1 \} \quad (3.60)$$

$$\frac{\partial^2 c_{ij}^1(x_1^k)}{\partial x_1^2} = 0 \quad \text{for } \{ i = i_e \} \text{ and } \{ k = i_e - 1 \text{ or } i_e + 1 \} \quad (3.61)$$

$$c_{i_e j_e}^1(x_1^k) \triangleq \text{natural cubic splines for } \{ i \neq i_e - 1, i_e, \text{ or } i_e + 1 \} \quad (3.62)$$

for $1 \leq k \leq I$. The $c_{ij}^2(x_2)$ are defined in a similar fashion for $1 \leq k \leq J$. Note that since $b_{i_e j_e} = 0$, the spline functions, $c_{i_e j_e}^1(x_1)$ and $c_{i_e j_e}^2(x_2)$, have no influence on the spline structure and are not defined in the above specifications.

An additional consideration apparent with the S.E.P. baseline controller is the ability to spline only the global information truly needed. The structure of the linear controller extends throughout the state space, and only where the linear controller fails does global information need to be specified. The values, b_{ij} , should be set to zero where the linear controller has sufficient control to provide adequate global performance.

Returning to the continuously stirred tank reactor, the region of global “fix-up” is only needed at $\mathbf{X} = [0.1 \ 5.0]^T$ and $\mathbf{X} = [0.5 \ 5.0]^T$. In all other places, the linear controller is sufficient to provide adequate global control. The form of the controller is given by

$$U(x_1, x_2) = -2.989 - 2.667 \cdot x_1 + 1.552 \cdot x_2 + \left[\sum_{i=1}^3 \sum_{j=1}^3 b_{ij} \cdot c_{ij}^1(x_1) \cdot c_{ij}^2(x_2) \right] \quad (3.63)$$

However, cardinal splines are only needed for two global grid points already mentioned. Therefore the form of the controller is given by

$$U(x_1, x_2) = -2.989 - 2.667 \cdot x_1 + 1.552 \cdot x_2 + b_{13} \cdot c_{13}^1(x_1) \cdot c_{13}^2(x_2) + b_{23} \cdot c_{23}^1(x_1) \cdot c_{23}^2(x_2) \quad (3.64)$$

where $b_{13}=11.555$ and $b_{23}=7.601$.

Assuming $x = x_1$, then the coefficients of $c_{13}^1(x)$ and $c_{23}^1(x)$ are determined by solving the following set of linear equations

for $i = 1, j = 3$, and $0.1 \leq x < 0.5$:

$$c_{13}^1(x_1) = ax_1^3 + bx_1^2 + cx_1 + d = 1 \quad (3.65)$$

$$c_{13}^1(x_2) = ax_2^3 + bx_2^2 + cx_2 + d = 0 \quad (3.66)$$

$$\frac{dc_{13}^1(x_2)}{dx} = 3 \cdot ax_2^2 + 2 \cdot bx_2 + c = 0 \quad (3.67)$$

$$\frac{d^2 c_{13}^1(x_2)}{dx^2} = 6 \cdot ax_2 + 2 \cdot b = 0 \quad (3.68)$$

for $i = 1, j = 3$, and $0.5 \leq x \leq 0.9$: $c_{13}^1(x) = 0 \quad (3.69)$

for $i = 2, j = 3$, and $0.1 \leq x < 0.5$:

$$c_{23}^1(x_1) = ax_1^3 + bx_1^2 + cx_1 + d = 0 \quad (3.70)$$

$$c_{23}^1(x_2) = ax_2^3 + bx_2^2 + cx_2 + d = 1 \quad (3.71)$$

$$\frac{dc_{23}^1(x_1)}{dx} = 3 \cdot ax_1^2 + 2 \cdot bx_1 + c = 0 \quad (3.72)$$

$$\frac{d^2 c_{23}^1(x_1)}{dx^2} = 6 \cdot ax_1 + 2 \cdot b = 0 \quad (3.73)$$

for $i = 2, j = 3$, and $0.5 \leq x \leq 0.9$:

$$c_{23}^1(x_2) = ax_2^3 + bx_2^2 + cx_2 + d = 1 \quad (3.74)$$

$$c_{23}^1(x_3) = ax_3^3 + bx_3^2 + cx_3 + d = 0 \quad (3.75)$$

$$\frac{dc_{23}^1(x_3)}{dx} = 3 \cdot ax_3^2 + 2 \cdot bx_3 + c = 0 \quad (3.76)$$

$$\frac{d^2 c_{23}^1(x_3)}{dx^2} = 6 \cdot ax_3 + 2 \cdot b = 0 \quad (3.77)$$

As stated before, $x_1 = 0.1$, $x_2 = 0.5$ and $x_3 = 0.9$ in all the above equations. The above cardinal cubic splines are shown in figure 3.12.

Assuming $y = x_2$, then the coefficients of $c_{13}^2(y)$ and $c_{23}^2(y)$ are determined by solving the following set of linear equations

$$\text{for } i = 1, j = 3, \text{ and } 1.0 \leq y < 3.0: \quad c_{13}^2(y) = 0 \quad (3.78)$$

for $i = 1, j = 3$, and $3.0 \leq y \leq 5.0$:

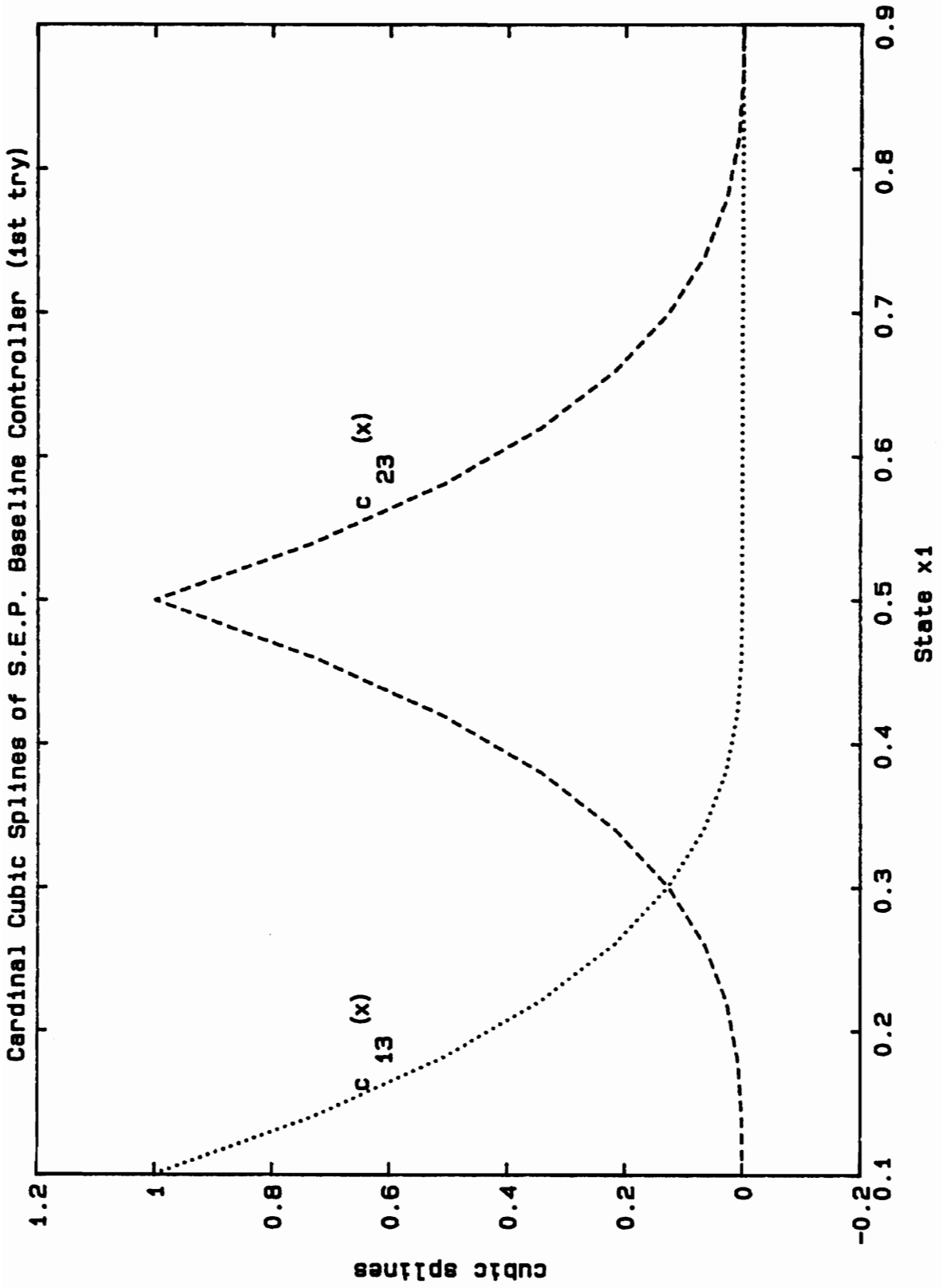


Figure 3.12

$$c_{13}^2(y_2) = ay_2^3 + by_2^2 + cy_2 + d = 0 \quad (3.79)$$

$$c_{13}^2(y_3) = ay_3^3 + by_3^2 + cy_3 + d = 1 \quad (3.80)$$

$$\frac{dc_{13}^2(y_2)}{dy} = 3 \cdot ay_2^2 + 2 \cdot by_2 + c = 0 \quad (3.81)$$

$$\frac{d^2 c_{13}^2(y_2)}{dy^2} = 6 \cdot ay_2 + 2 \cdot b = 0 \quad (3.82)$$

for $i = 2, j = 3$, and $1.0 \leq y < 3.0$: $c_{23}^2(y) = 0 \quad (3.83)$

for $i = 1, j = 3$, and $3.0 \leq y \leq 5.0$:

$$c_{23}^2(y_2) = ay_2^3 + by_2^2 + cy_2 + d = 0 \quad (3.84)$$

$$c_{23}^2(y_3) = ay_3^3 + by_3^2 + cy_3 + d = 1 \quad (3.85)$$

$$\frac{dc_{23}^2(y_2)}{dy} = 3 \cdot ay_2^2 + 2 \cdot by_2 + c = 0 \quad (3.86)$$

$$\frac{d^2 c_{23}^2(y_2)}{dy^2} = 6 \cdot ay_2 + 2 \cdot b = 0 \quad (3.87)$$

Note that $y_1 = 1.0$, $y_2 = 3.0$ and $y_3 = 5.0$ in all the above equations. The above cardinal cubic spline is shown in figure 3.13.

The control value of a linear controller as a function of the two state variables is a control surface represented by a plane in three dimensions. The tilt of the plane is determined by the feedback gains, G_1 and G_2 . The linear control portion of the S.E.P. baseline controller is the “tilted plane” seen in figure 3.14. The figure also shows the separation of local control, given by the plane, and global control, given by the peaks. The controller is limited to a value of ten after the global information has been added which results in a plateau above the $X = [0.1 \ 5.0]^T$ region. The phase trajectories in figure 3.15 show that the S.E.P. baseline controller has good global performance.

The specifying of conditions (3.57) to (3.62) do not necessarily represent the best possible design approach for using cubic splines, however they do give a good generic framework. Departing from using natural cubic splines, the system designer can tailor the controller by choosing cubic splines that best fit the data set, for which conditions (3.57) to (3.62) are just one method. For the continuously stirred tank reactor example, a better controller occurs if the spline $c_{13}^1(x)$ is changed to the function that solves the following set of linear equations

for $i = 1$, $j = 3$, and $0.1 \leq x < 0.5$:

$$c_{13}^1(x_1) = ax_1^3 + bx_1^2 + cx_1 + d = 1 \quad (3.88)$$

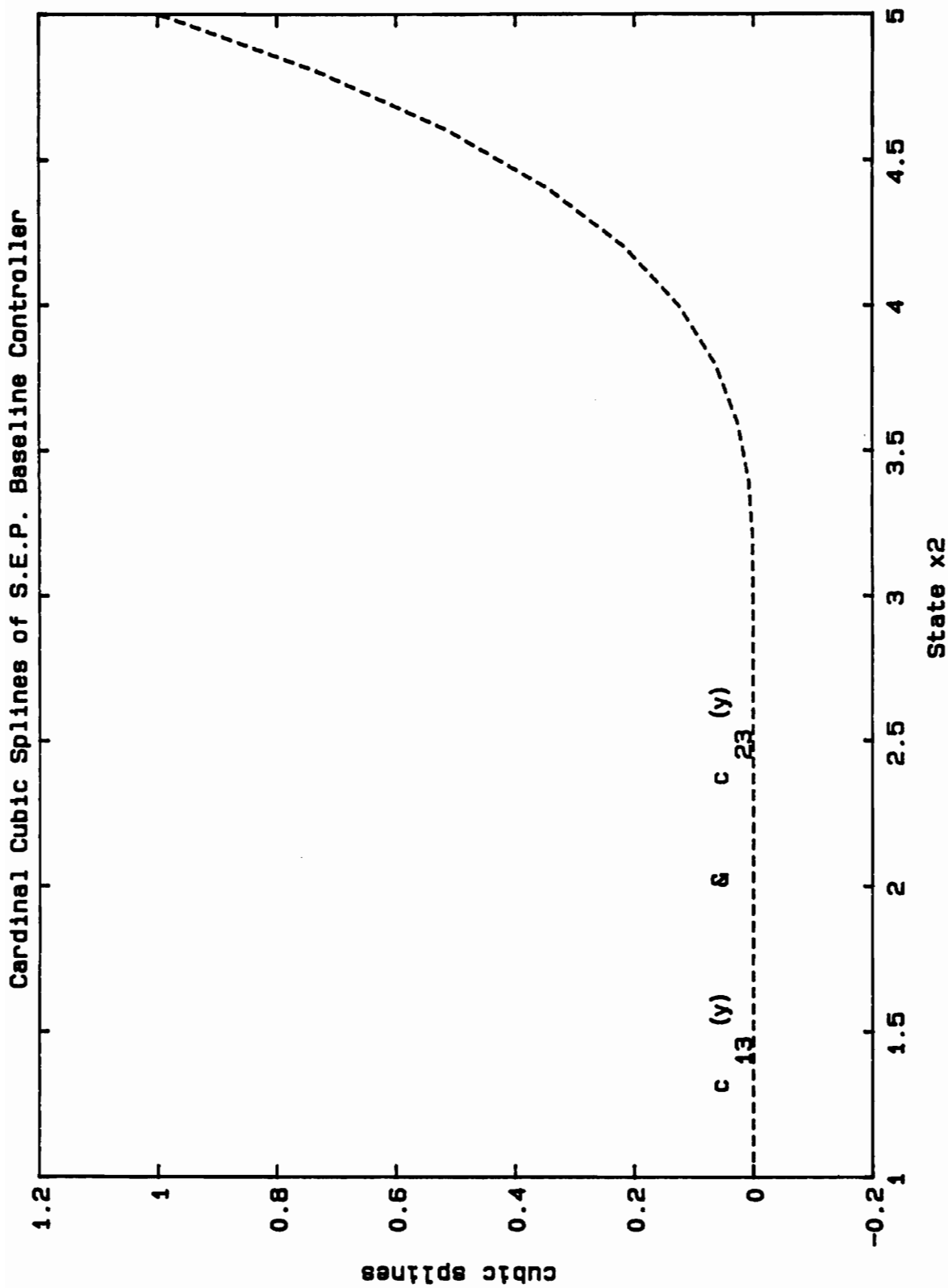


Figure 3.13

Control Surface of S.E.P. Baseline Controller (1st Try)

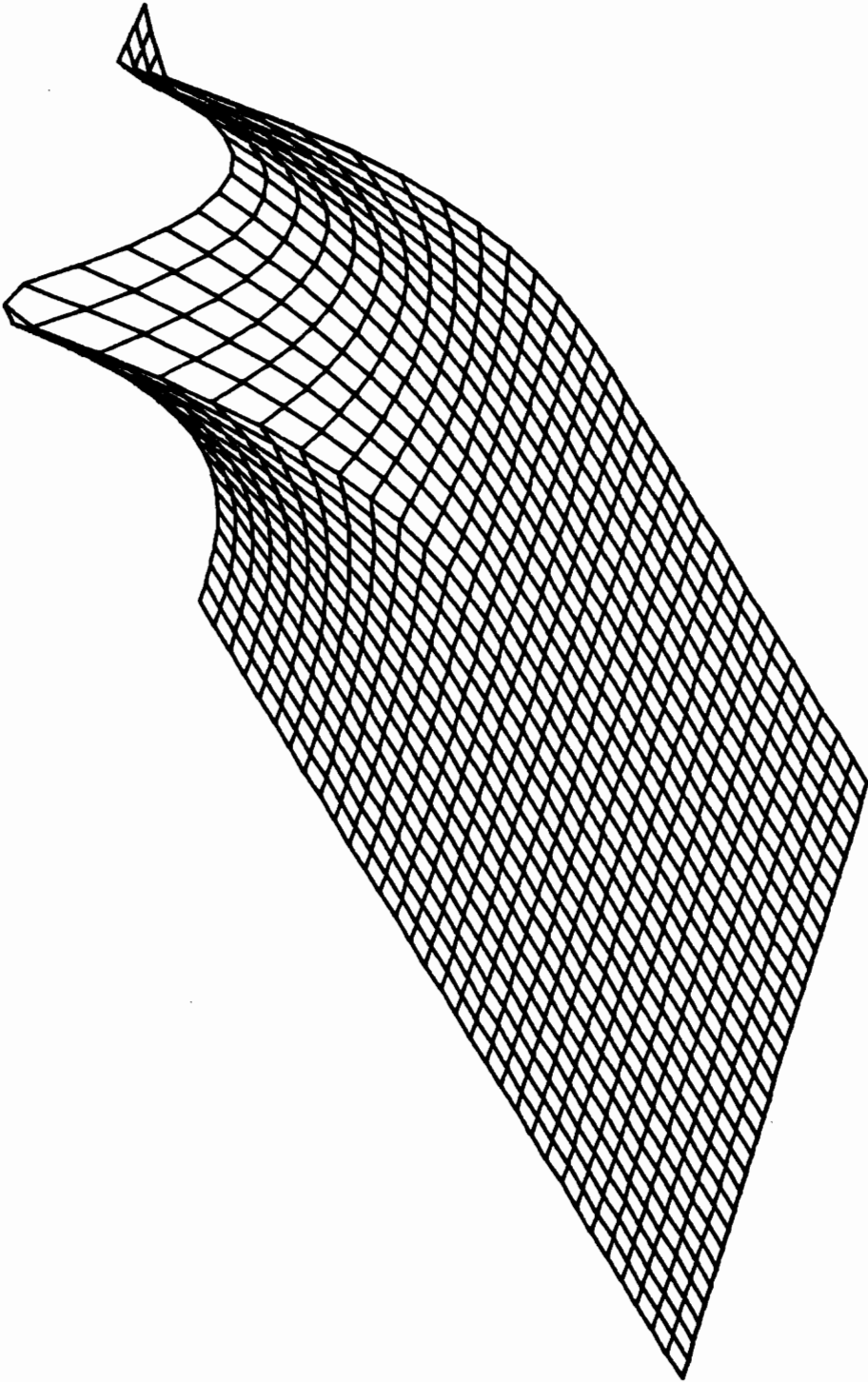


Figure 3.14

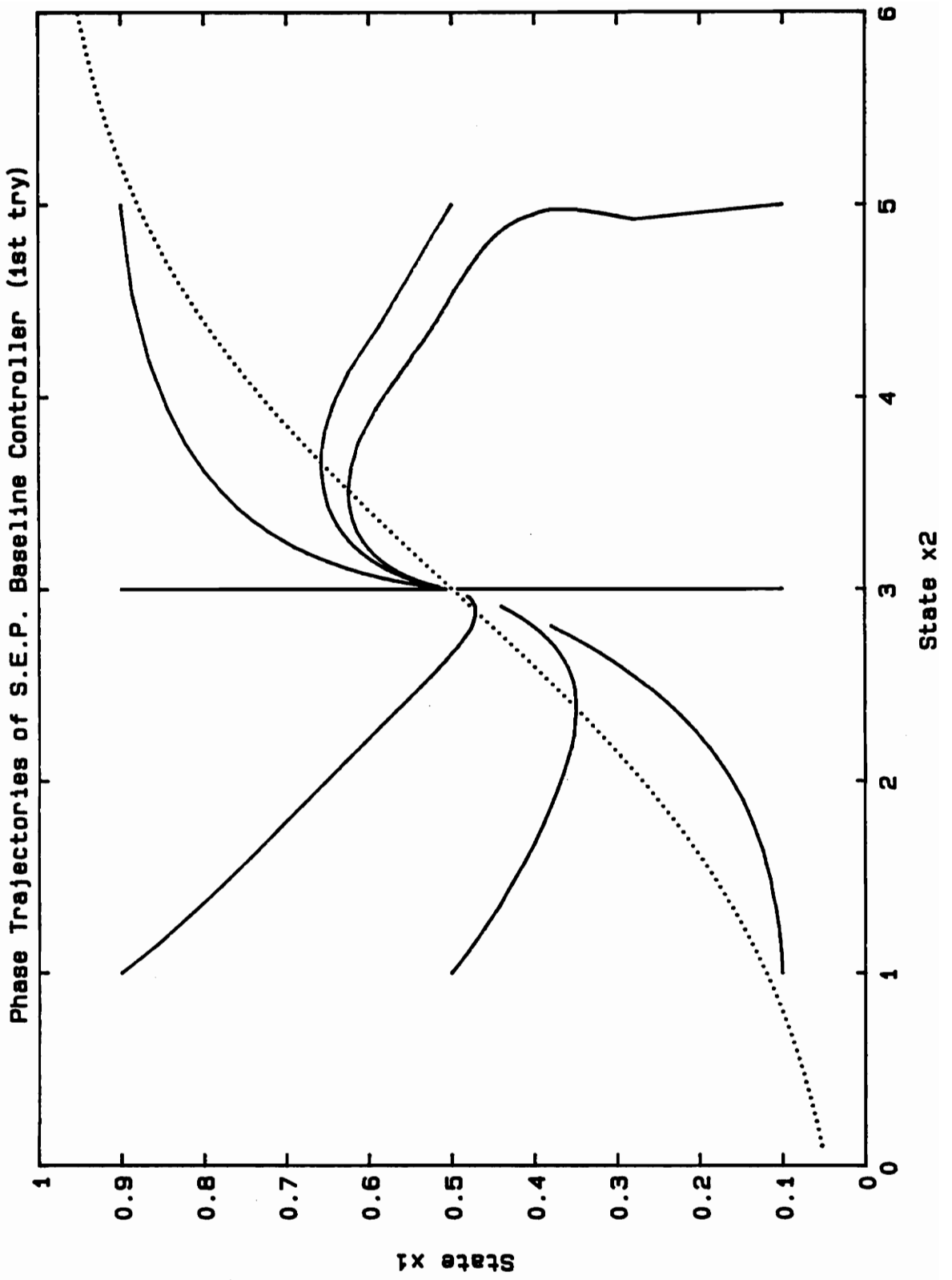


Figure 3.15

$$c_{13}^1(x_2) = ax_2^3 + bx_2^2 + cx_2 + d = 0 \quad (3.89)$$

$$\frac{dc_{13}^1(x_1)}{dx} = 3 \cdot ax_1^2 + 2 \cdot bx_1 + c = 0 \quad (3.90)$$

$$\frac{d^2 c_{13}^1(x_1)}{dx^2} = 6 \cdot ax_1 + 2 \cdot b = 0 \quad (3.91)$$

for $i = 1$, $j = 3$, and $0.5 \leq x \leq 0.9$: $c_{13}^1(x) = 0 \quad (3.92)$

Again, $x_1 = 0.1$, $x_2 = 0.5$ and $x_3 = 0.9$ in all the above equations. The above cardinal cubic spline is shown in figure 3.16. Equations (3.70) to (3.87) still apply for the other cubic splines, $c_{23}^1(x)$, $c_{13}^2(y)$, and $c_{23}^2(y)$. Figure 3.17 shows the control surface for the S.E.P. baseline controller that results using the new cubic spline for $c_{13}^1(x)$. Figure 3.18 shows the phase trajectories for this controller.

Figure 3.19 and 3.20 compare the state trajectories versus time of the linear, tuned extended-linear, and S.E.P. baseline controllers. The S.E.P. baseline controller performs considerably better than the linear controller and moderately better than the tuned extended-linear controller. As mention in section 3.2, when the S.E.P. baseline controller is limited to a value less than ten, the system has better phase trajectories and uses less overall control than the system with the linear controller. Refer to figure 3.21. This is because the linear controller needs a large amount of control to “turn” the system

Cardinal Cubic Splines of S.E.P. Baseline Controller

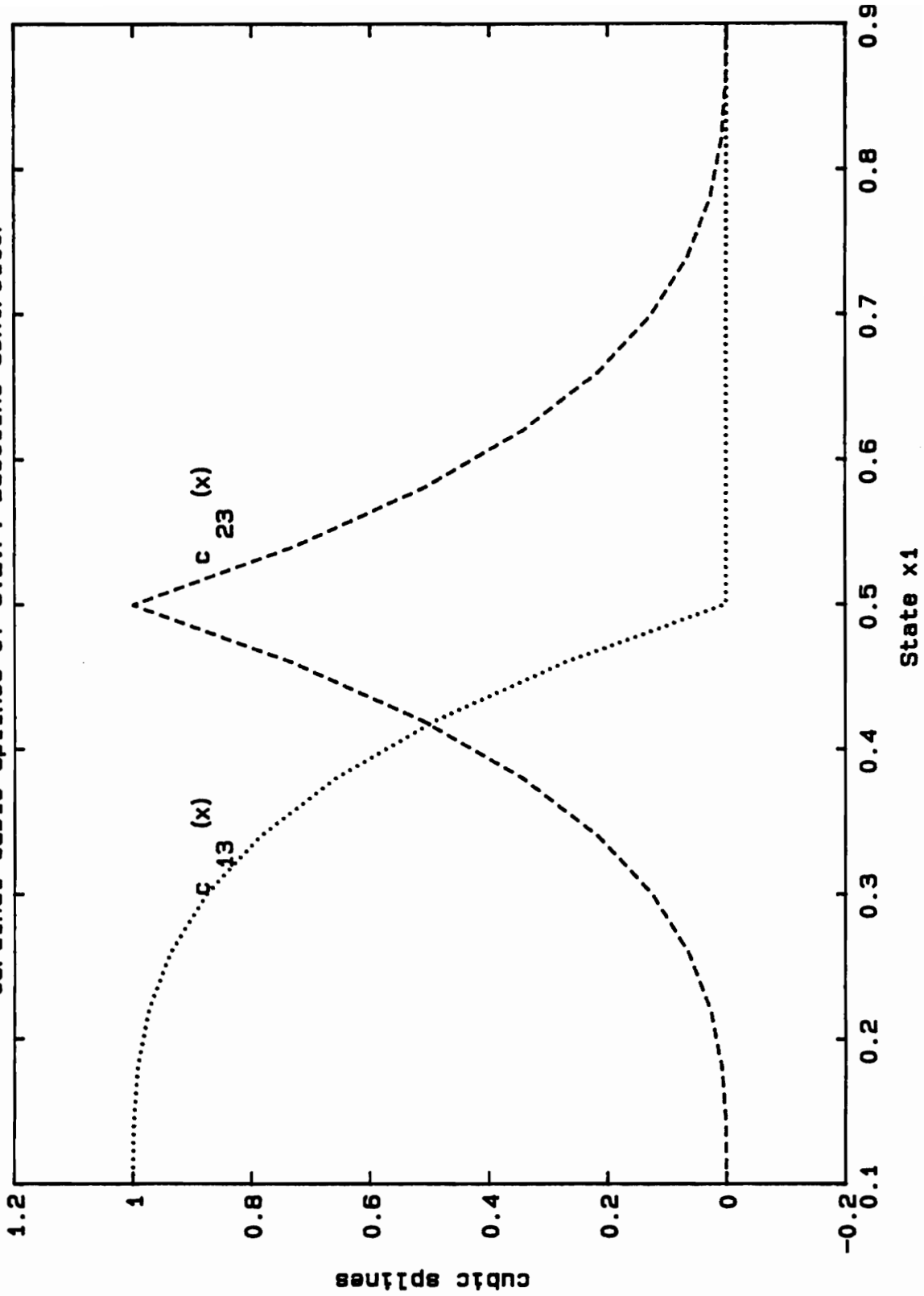


Figure 3.16

Control Surface of S.E.P. Baseline Controller

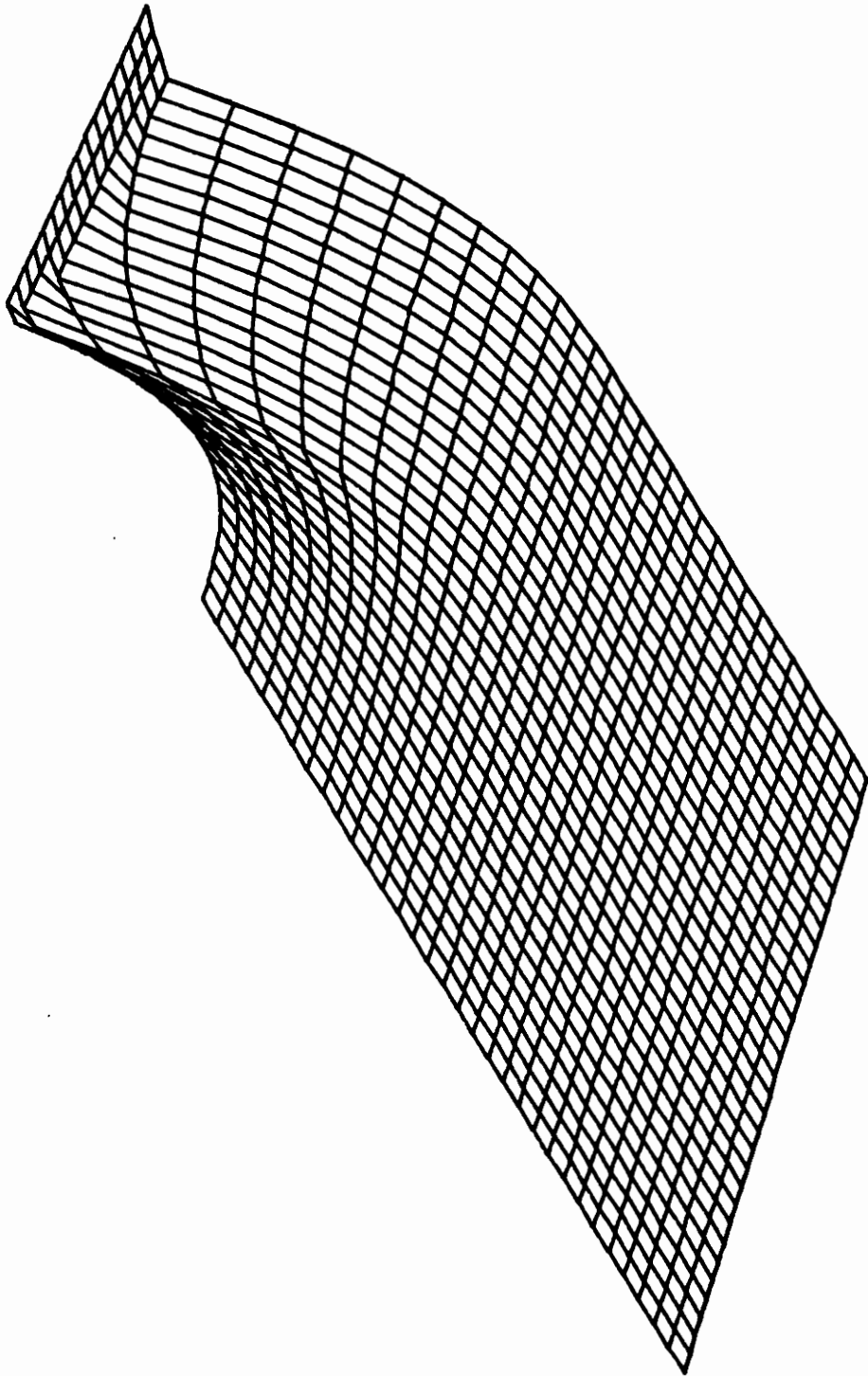


Figure 3.17

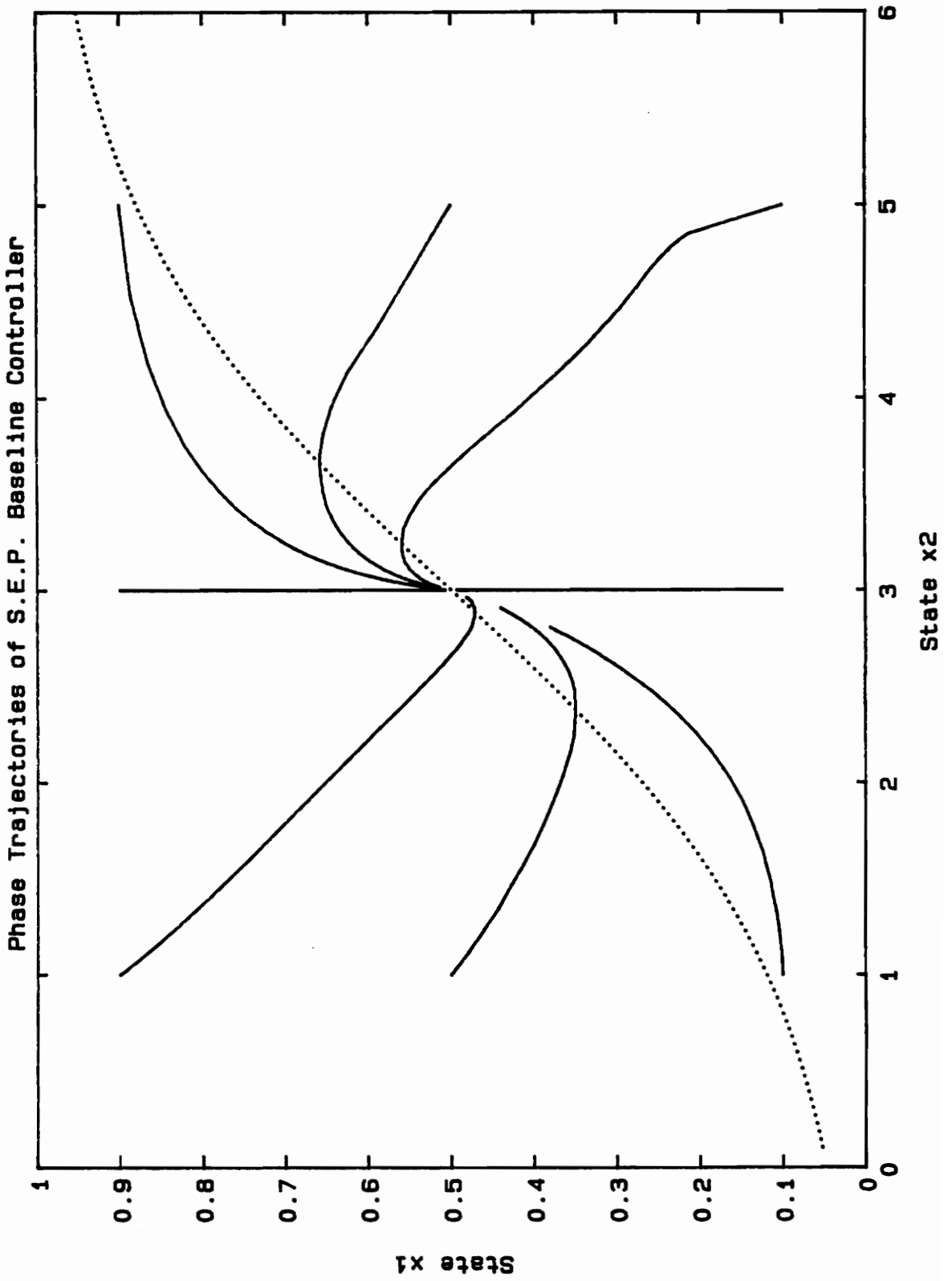


Figure 3.18

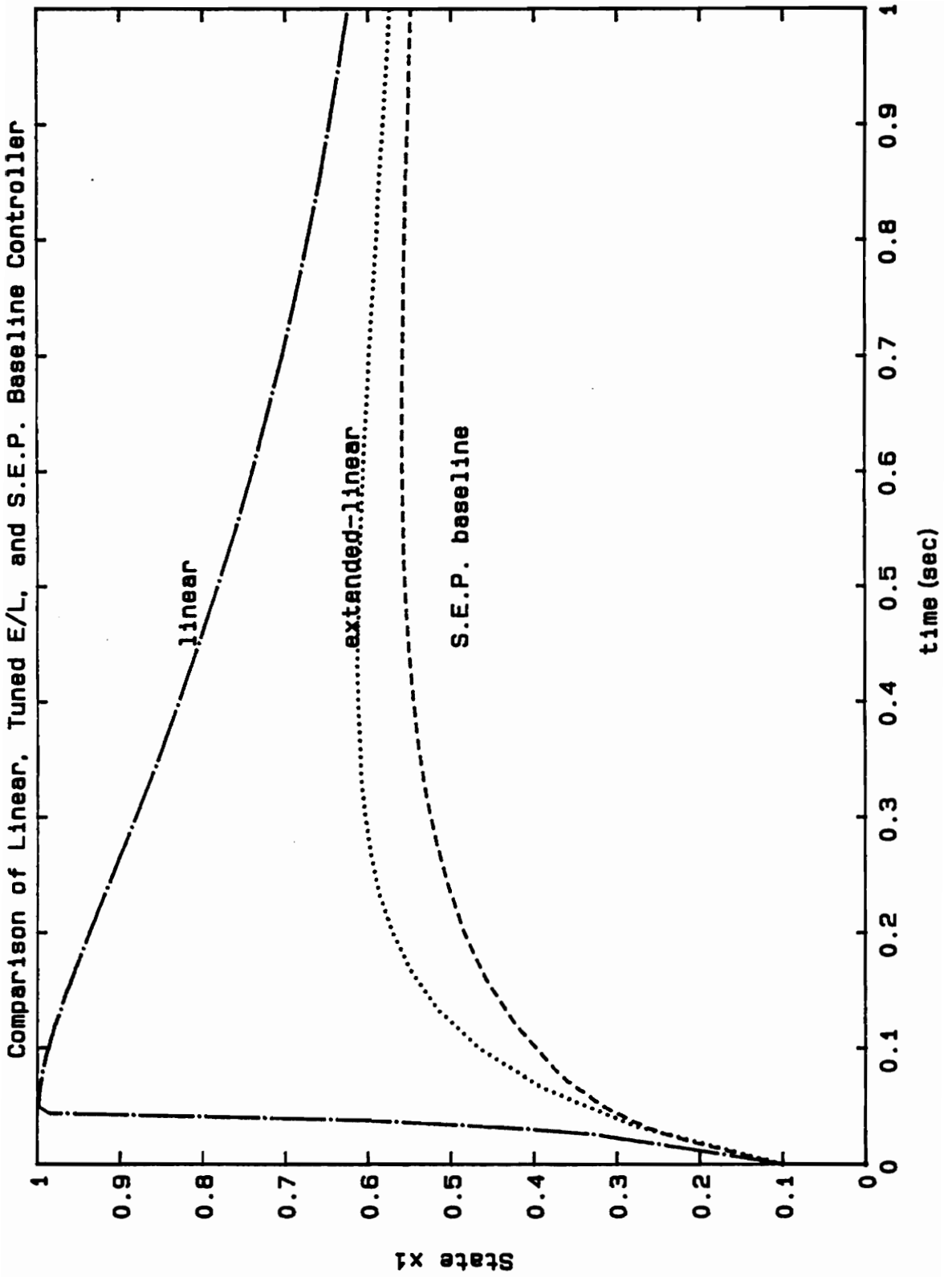
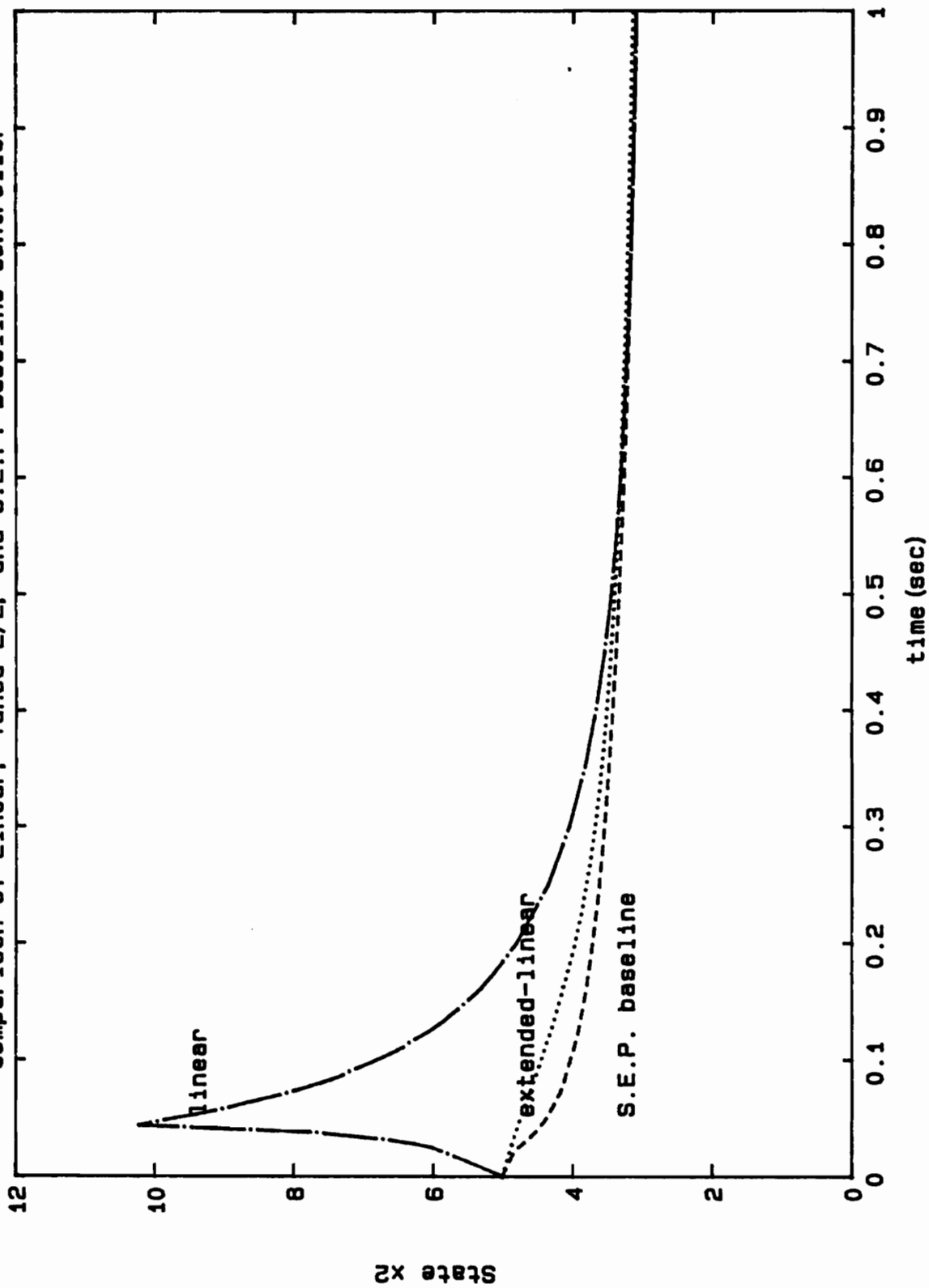


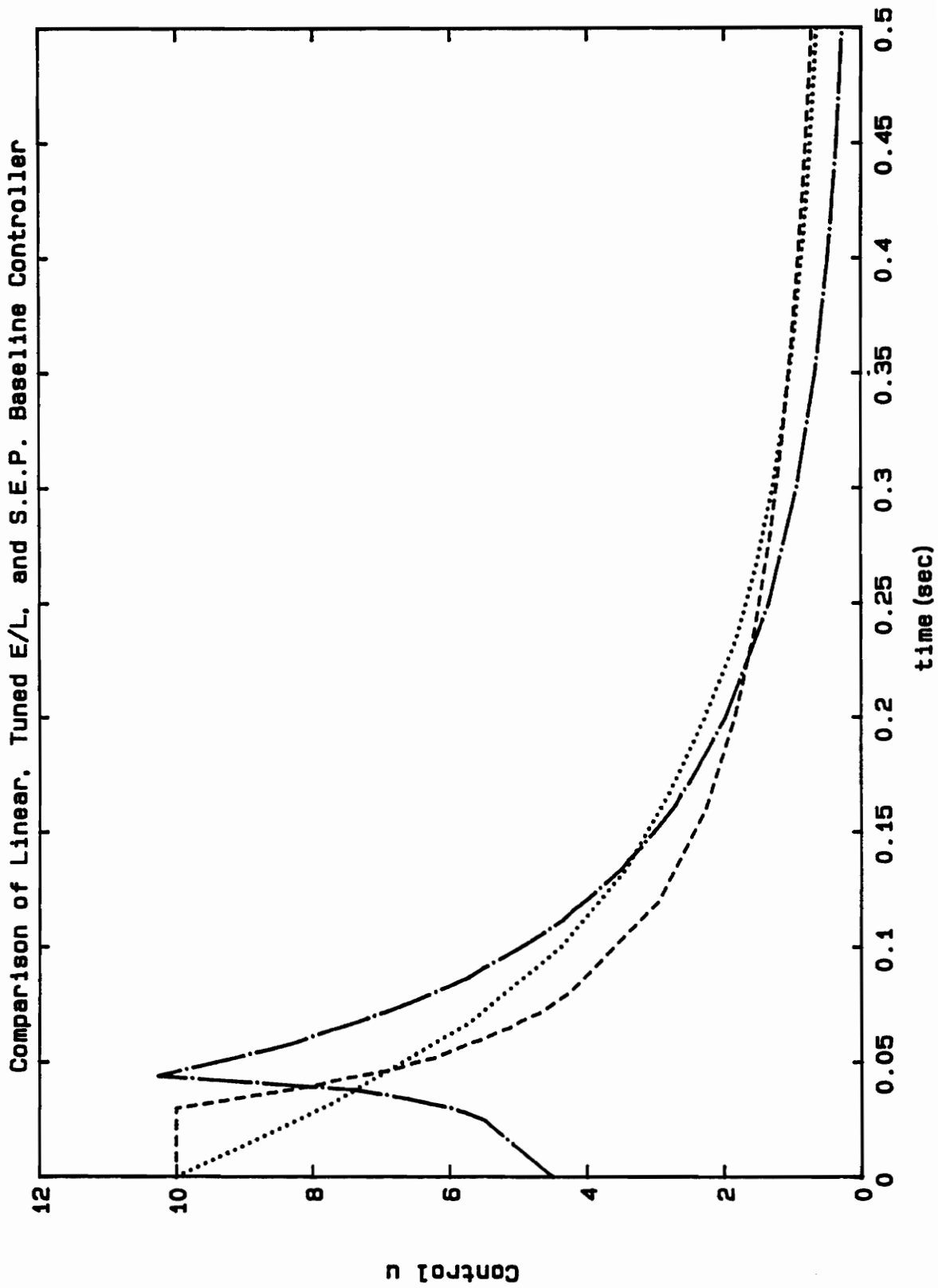
Figure 3.19

Comparison of Linear, Tuned E/L, and S.E.P. Baseline Controller



State x_2

Figure 3.20



Control u

Figure 3.21

toward the equilibrium point when the states are in the region around $\mathbf{X} = [0.99 \ 10.0]^T$. In contrast, the S.E.P. baseline controller uses the same amount of control in the beginning to stem the states' trajectories away from the $\mathbf{X} = [0.99 \ 10.0]^T$ region and toward the equilibrium point.

Unlike the first two prototypes, the S.E.P. baseline controller still obtains the desired local performance near the equilibrium grid point. Refer to figure 3.22 where the dashed line represents the linear controller and the solid line represents the S.E.P. baseline controller. Note that no difference exists between the trajectories when x_2 is less than three because the global information is zero in this region. However, as x_2 increases above three, the global specifications affect the trajectories more and more.

Next, the S.E.P. baseline controller will be applied to two more equilibrium points, $\mathbf{X}^o = [0.12 \ 1.0]^T$ and $\mathbf{X}^o = [0.88 \ 5.0]^T$. The first step is to design a linear controller for each specific equilibrium point. For the first equilibrium point, the linear controller, with eigenvalue placements of -2.0 , needs no global "fix-up" to have good global performance. Refer to figure 3.23. For the second equilibrium point, the phase trajectories of a linear controller, with eigenvalue placements of -2.0 , are shown in figure 3.24. The figure shows that not only is global control needed in the $\mathbf{X} = [0.1 \ 5.0]^T$ region, but that the linear controller has a spurious equilibrium point at $\mathbf{X}^o = [0.05 \ 0.2]^T$. To eliminate this unwanted equilibrium point, most of the left side of the region of interest needs to be pushed down. The point is eliminated if a control plane extends from a value of -10.0 along $x = 0.9$ to a value of -1.0 along $x = 0.1$, for $1 \leq y \leq 3$. This global information plus the global information from the grid points at $\mathbf{X} = [0.1 \ 5.0]^T$ and $\mathbf{X} = [0.5 \ 5.0]^T$ results in the control surface shown in figure 3.25. Conditions (3.57) to (3.62) were used to solve for the cubic splines. The resulting phase trajectories are shown in figure 3.26. The

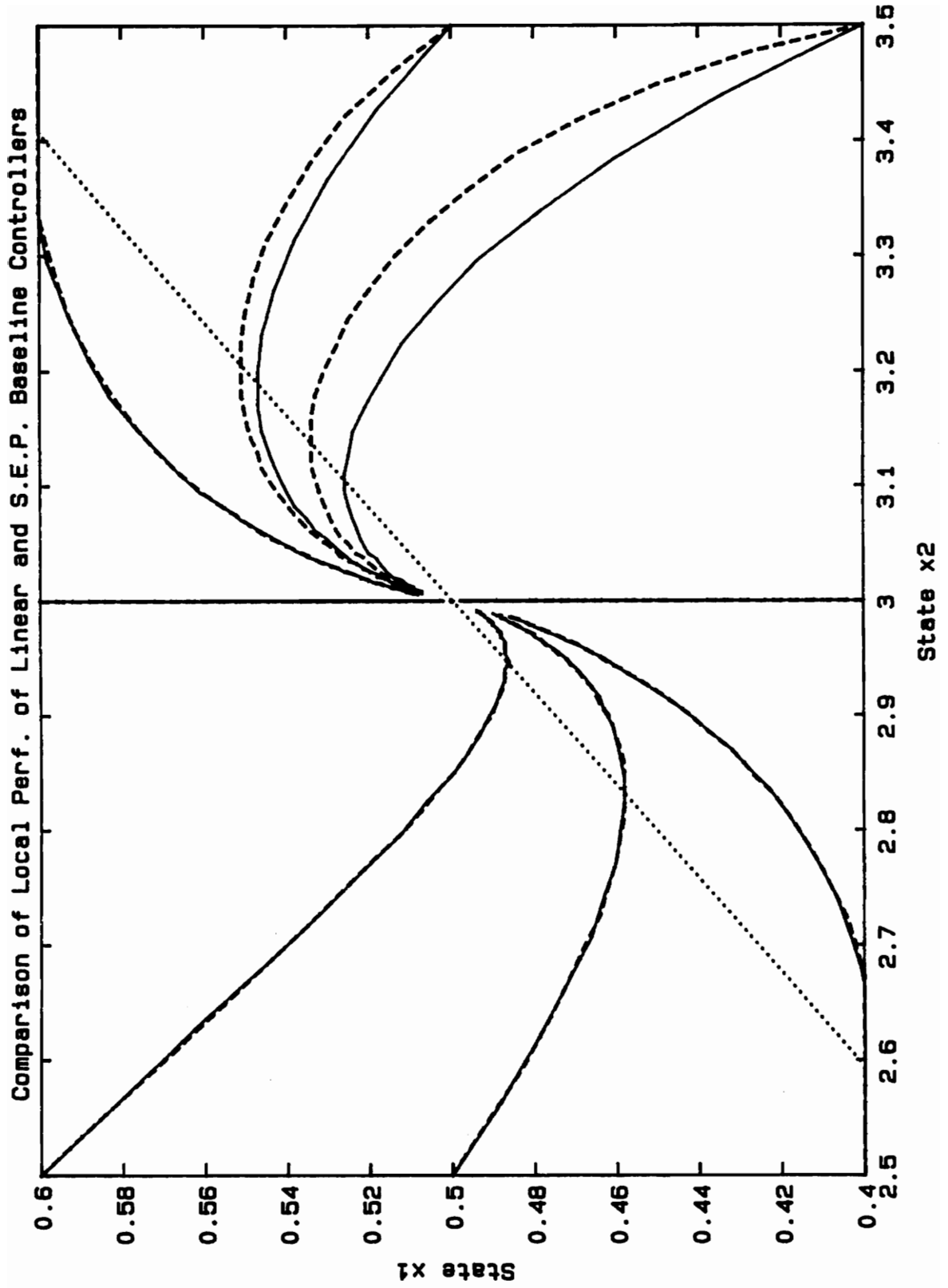


Figure 3.22

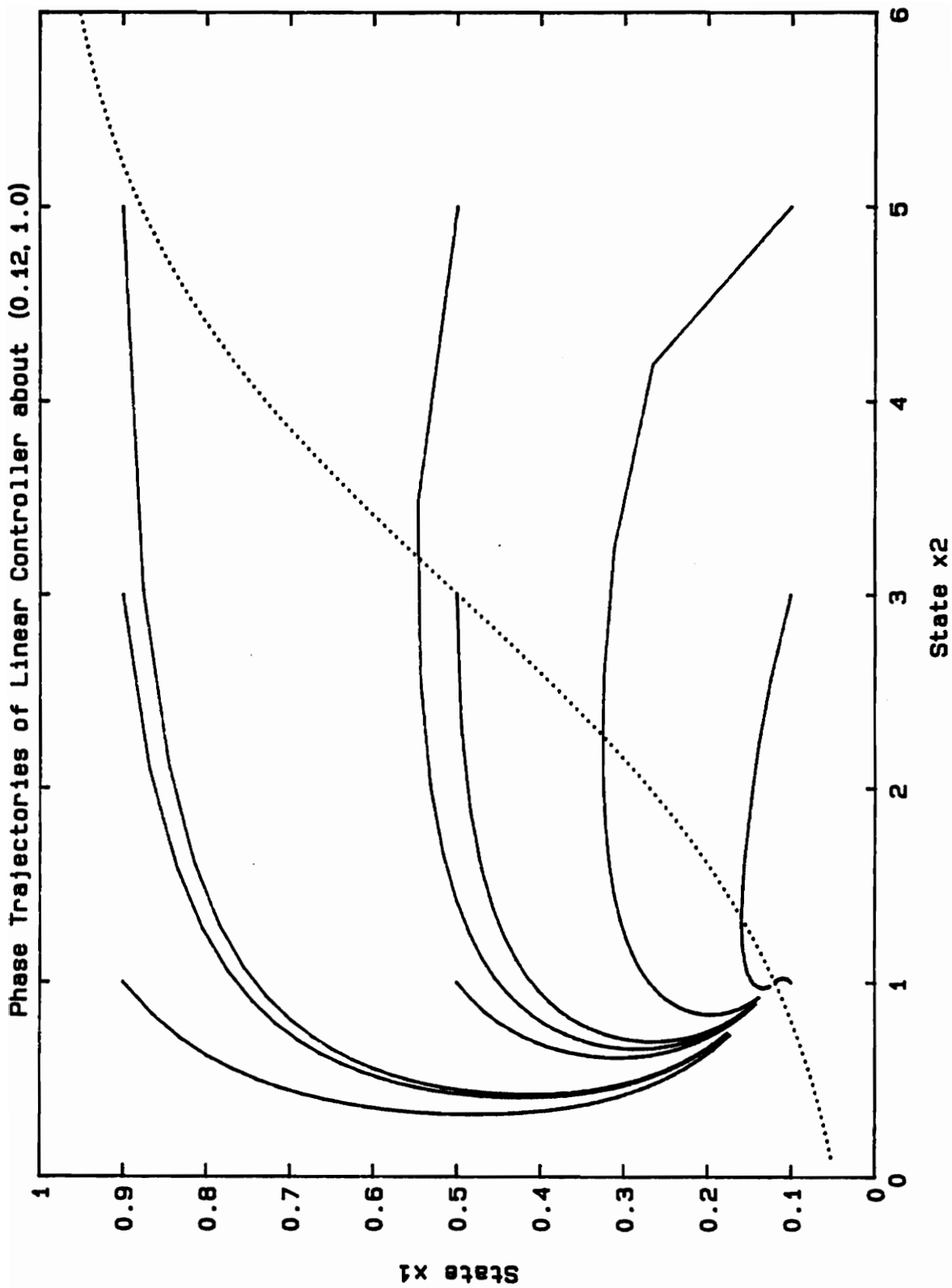


Figure 3.23

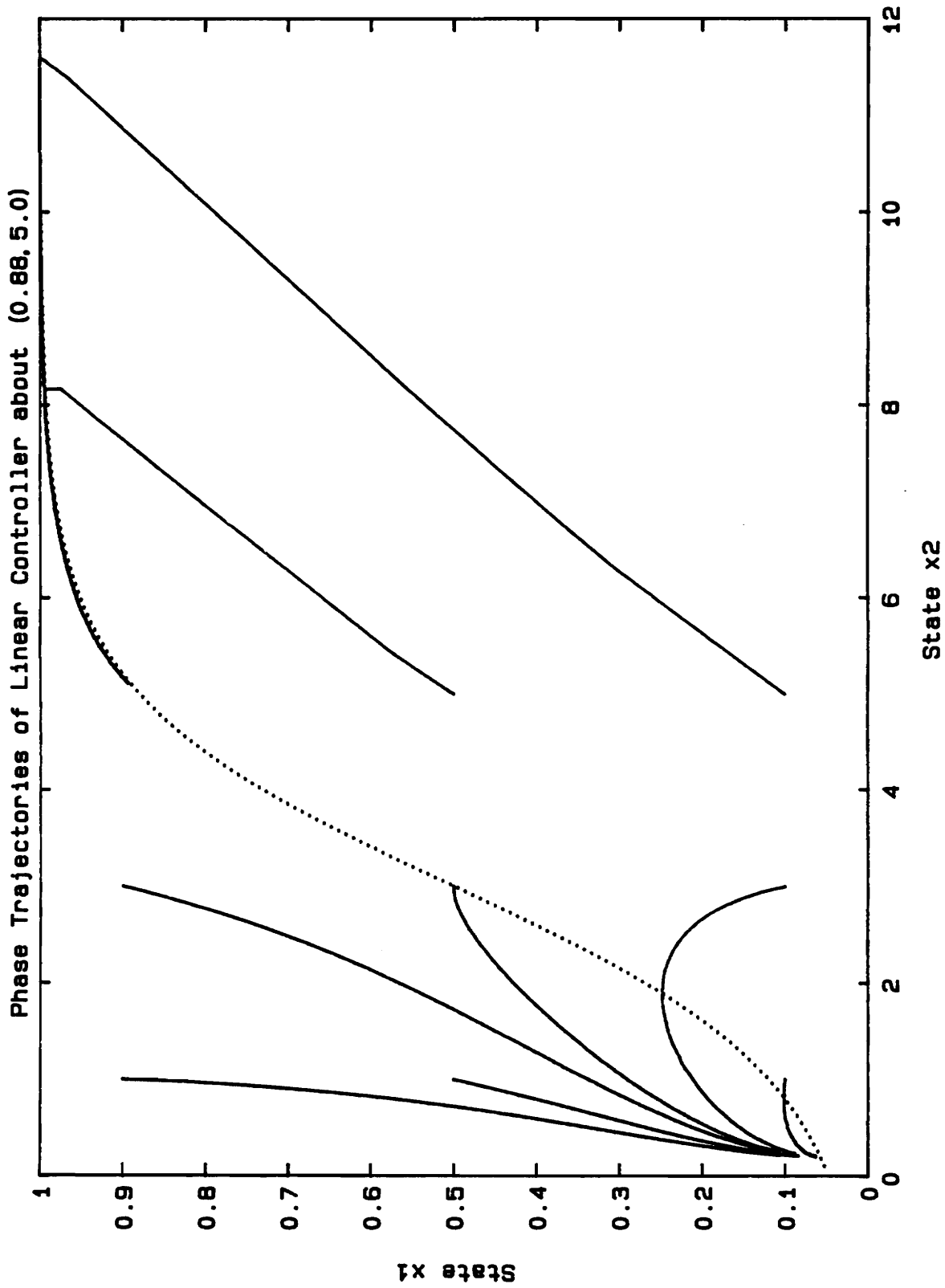


Figure 3.24

Control Surface of S.E.P. Baseline Controller about (0.88, 5.0)

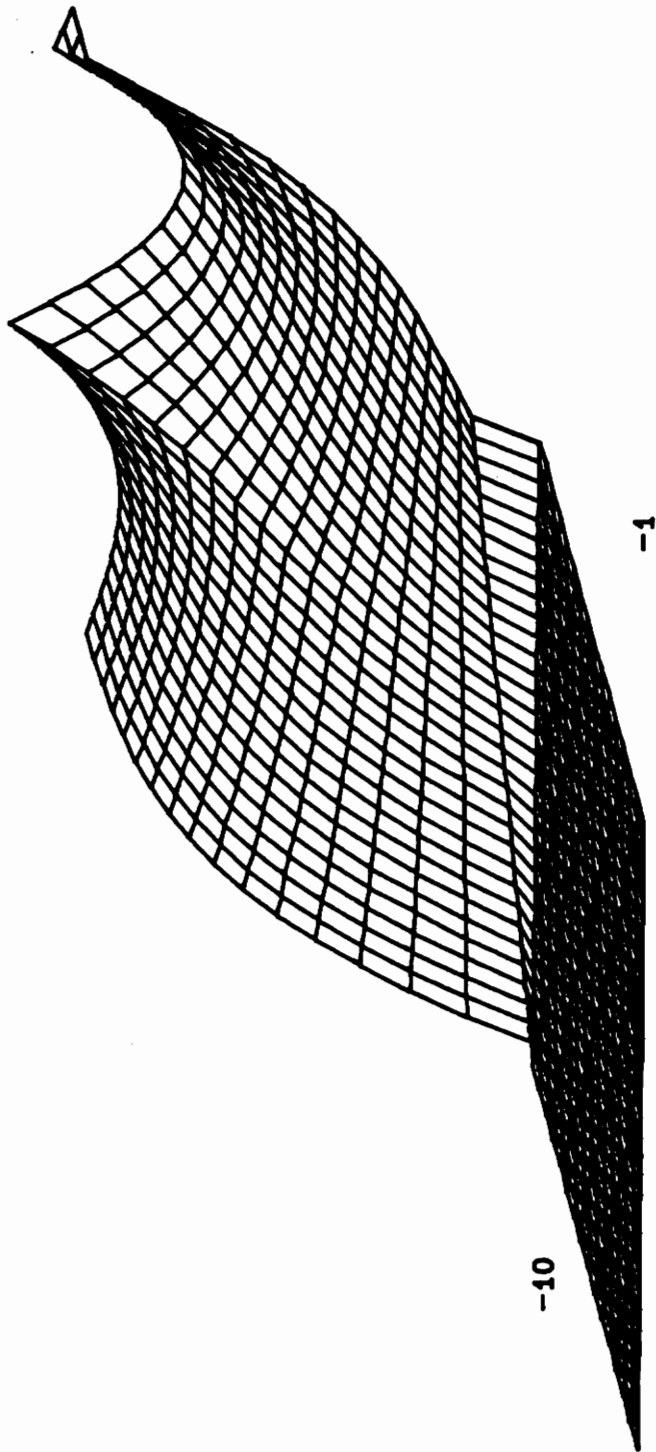


Figure 3.25

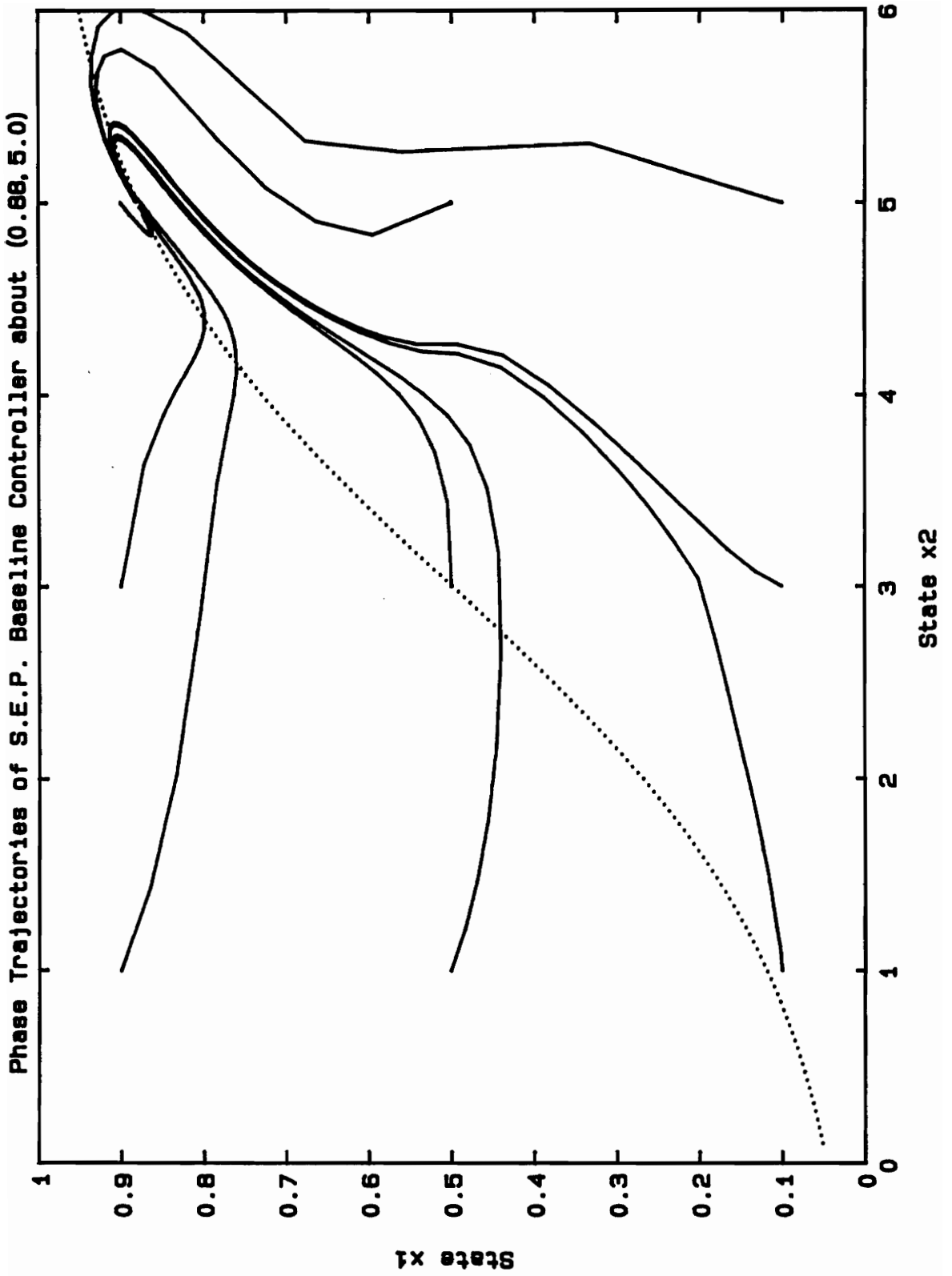


Figure 3.26

S.E.P. baseline controller not only eliminated the spurious equilibrium point, but it has obtain good global performance in the $X = [0.1 \ 5.0]^T$ region without sacrificing the local control performance about $X^o = [0.88 \ 5.0]^T$. The next chapter with try to extend this S.E.P. baseline approach to three dimensions where the third dimension will be the external input variable, w .

4.0 VARYING EXTERNAL INPUT GLOBAL CONTROLLERS

The previous chapter's goal was to obtain good global performance at a specified equilibrium point without sacrificing the local control performance. Chapter four will try to obtain the same goal but with a varying external input. As with chapter three, this chapter will deal only with two state variables, however, with the addition of the external input as a dependent variable, the state-control space is now three dimensions. First, this chapter will examine the S.E.P. baseline controller extended into three dimensions. Next, the V.E.P. baseline controller will be discussed.

4.1 Extended S.E.P. Controller

The S.E.P. baseline controller produces the desired local behavior and also obtains good global performance for a particular constant external input. Also for each constant external input, the S.E.P. baseline controller generates a different control surface that obtains a unique equilibrium point. For

the continuously stirred tank reactor, two different surfaces are shown in figures 3.17 and 3.25. The “extended S.E.P. baseline” controller tries to piece together these control surfaces using cardinal splines in the added dimension, i.e. in the external control direction. The form of the controller is

$$U(x_1, x_2, w) = \sum_{m=1}^M U_m(x_1, x_2) \cdot c_m^3(w) \quad (4.1)$$

where $U_m(x_1, x_2) \equiv$ control surfaces generated by S.E.P. baseline approach

and $c_m^3(w) \equiv$ cardinal cubic splines in direction of external input.

and M is the number of data points in the w -direction. Again the function $c_m^3(w)$ does not uniquely determine a control law.

To get a unique control law, the extended-linearization theory will be used. For each equilibrium point, there exist a set of feedback gains, G_1 and G_2 , that determine the local control objective at that point. The extended-linearization theory discussed in section 2.2 provides the following guidance. First, on the equilibrium curve, the partials of the control function with respect to the states must match the respective feedback gains, i.e.

$$\left. \frac{\partial U(x_1, x_2, w)}{\partial x_1} \right|_{\substack{\mathbf{x}=\mathbf{x}_o \\ w=w_o}} = G_1 \quad (4.2)$$

$$\left. \frac{\partial U(x_1, x_2, w)}{\partial x_2} \right|_{\substack{x=x^o \\ w=w}} = G_2 \quad (4.3)$$

Second, using equation (2.26b), G_1 and G_2 uniquely determine H , the gain associated with the external input. Finally, on the equilibrium curve, the partial of the control function with respect to the external input must equal H , i.e.

$$\left. \frac{\partial U(x_1, x_2, w)}{\partial w} \right|_{\substack{x=x^o \\ w=w}} = H \quad (4.4)$$

Therefore given G_1 and G_2 at an equilibrium point, the gain H is determined by equation (2.26b) which gives the partial of the control function in the third direction. The conditions (4.2) and (4.3) are already established by the control surfaces, $U_m(x_1, x_2)$. To establish condition (4.4), the unique piecewise cubic polynomial, $c_m^3(w)$, must satisfy

$$c_m^3(w_k) = \delta_{mk} \quad (4.5)$$

$$\frac{\partial c_m^3(w_k)}{\partial w} = \frac{H_m}{U_m(x_1^o, x_2^o)} \cdot \delta_{mk} \quad (4.6)$$

for $1 \leq k \leq M$. Note that H_m is a constant determined by the feedback gains, using equation (2.26b), of the particular control surface, and that $U_m(x_1^o, x_2^o)$ is evaluated at the equilibrium point for that particular surface.

Returning to the continuously stirred tank reactor example, the state-control space will have three grid points in the w -direction pertaining to

$$\mathbf{x}^o = [0.12 \quad 1.0]^T, \quad (4.7)$$

$$\mathbf{x}^o = [0.50 \quad 3.0]^T, \quad (4.8)$$

and

$$\mathbf{x}^o = [0.88 \quad 5.0]^T, \quad (4.9)$$

which are the equilibrium points discussed in the last chapter. For each of these equilibrium points, there will be a control surface, $U_m(x_1, x_2)$, and an equilibrium control value, $U_m(x_1^o, x_2^o)$, associated with it. The equilibrium control values for $m = 1, 2, \text{ and } 3$ are respectively $\{ -0.0432, 0.333, 0.408 \}$ which also equal w_k for $k = 1, 2, \text{ and } 3$.

For each subinterval in the w -direction, the form of the cardinal cubic spline is given by

$$c_m^3(w) = aw^3 + bw^2 + cw + d \quad (4.10)$$

The coefficients are determined by solving the following set of linear equations

for $m = 1$ and $-0.0432 \leq w < 0.333$:

$$c_1^3(w_1) = aw_1^3 + bw_1^2 + cw_1 + d = 1 \quad (4.11)$$

$$c_1^3(w_2) = aw_2^3 + bw_2^2 + cw_2 + d = 0 \quad (4.12)$$

$$\frac{dc_1^3(w_1)}{dw} = 3 \cdot aw_1^2 + 2 \cdot bw_1 + c = \frac{H_1}{U_1(0.12, 1.0)} \quad (4.13)$$

$$\frac{dc_1^3(w_2)}{dw} = 3 \cdot aw_2^2 + 2 \cdot bw_2 + c = 0 \quad (4.14)$$

for $m = 1$ and $0.333 \leq w \leq 0.408$: $c_1^3(w) = 0 \quad (4.15)$

for $m = 2$ and $-0.0432 \leq w < 0.333$:

$$c_2^3(w_1) = aw_1^3 + bw_1^2 + cw_1 + d = 0 \quad (4.16)$$

$$c_2^3(w_2) = aw_2^3 + bw_2^2 + cw_2 + d = 1 \quad (4.17)$$

$$\frac{dc_2^3(w_1)}{dw} = 3 \cdot aw_1^2 + 2 \cdot bw_1 + c = 0 \quad (4.18)$$

$$\frac{dc_2^3(w_2)}{dw} = 3 \cdot aw_2^2 + 2 \cdot bw_2 + c = \frac{H_2}{U_2(0.5, 3.0)} \quad (4.19)$$

for $m = 2$ and $0.333 \leq w \leq 0.408$:

$$c_2^3(w_2) = aw_2^3 + bw_2^2 + cw_2 + d = 1 \quad (4.20)$$

$$c_2^3(w_3) = aw_3^3 + bw_3^2 + cw_3 + d = 0 \quad (4.21)$$

$$\frac{dc_2^3(w_2)}{dw} = 3 \cdot aw_2^2 + 2 \cdot bw_2 + c = \frac{H_2}{U_2(0.5, 3.0)} \quad (4.22)$$

$$\frac{dc_2^3(w_3)}{dw} = 3 \cdot aw_3^2 + 2 \cdot bw_3 + c = 0 \quad (4.23)$$

for $m = 3$ and $-0.432 \leq w < 0.333$: $c_3^3(w) = 0 \quad (4.24)$

for $m = 3$ and $0.333 \leq w \leq 0.408$:

$$c_2^3(w_2) = aw_2^3 + bw_2^2 + cw_2 + d = 0 \quad (4.25)$$

$$c_2^3(w_3) = aw_3^3 + bw_3^2 + cw_3 + d = 1 \quad (4.26)$$

$$\frac{dc_2^3(w_2)}{dw} = 3 \cdot aw_2^2 + 2 \cdot bw_2 + c = 0 \quad (4.27)$$

$$\frac{dc_2^3(w_3)}{dw} = 3 \cdot aw_3^2 + 2 \cdot bw_3 + c = \frac{H_3}{U_3(0.88, 5.0)} \quad (4.28)$$

The value H_m equals $\{-3.52, -0.667, -0.096\}$ for $m = 1, 2,$ and $3,$ and the value $U_m(x_1^o, x_2^o)$ equals $\{-0.0432, 0.333, 0.408\}$ for $m = 1, 2,$ and $3.$ Note that a unique solution is not guaranteed for the above cubic splines since the values, $w_1, w_2,$ and $w_3,$ may not be monotonically increasing, and could equal each other. Further consideration suggest that the third direction could have been parametrized by

$$w = x_2^o \quad (4.29)$$

instead of

$$w = u^o \quad (4.30)$$

Any parametrization can be used in equation (4.1), and the parametrization using (4.29) would not suffer from the nonuniqueness of using (4.30). Further simulations using (4.29) were not tried, since the “extended S.E.P. baseline” controller failed to track the input properly.

If the system is commanded to go to any of the gridded equilibrium points, (4.7), (4.8), or (4.9), then the system behaves correctly, but if the system is commanded to go to an intermediate equilibrium point between the grid points, the system fails. For example, for a value of $w = 0.078,$ the phase trajectories should stop at the equilibrium point $X^o = [0.27 \ 2.0]^T,$ yet the phase trajectories end at $X^o = [0.13 \ 1.2]^T.$ Refer to figure 4.1 where the “+” marks where the phase trajectories should end.

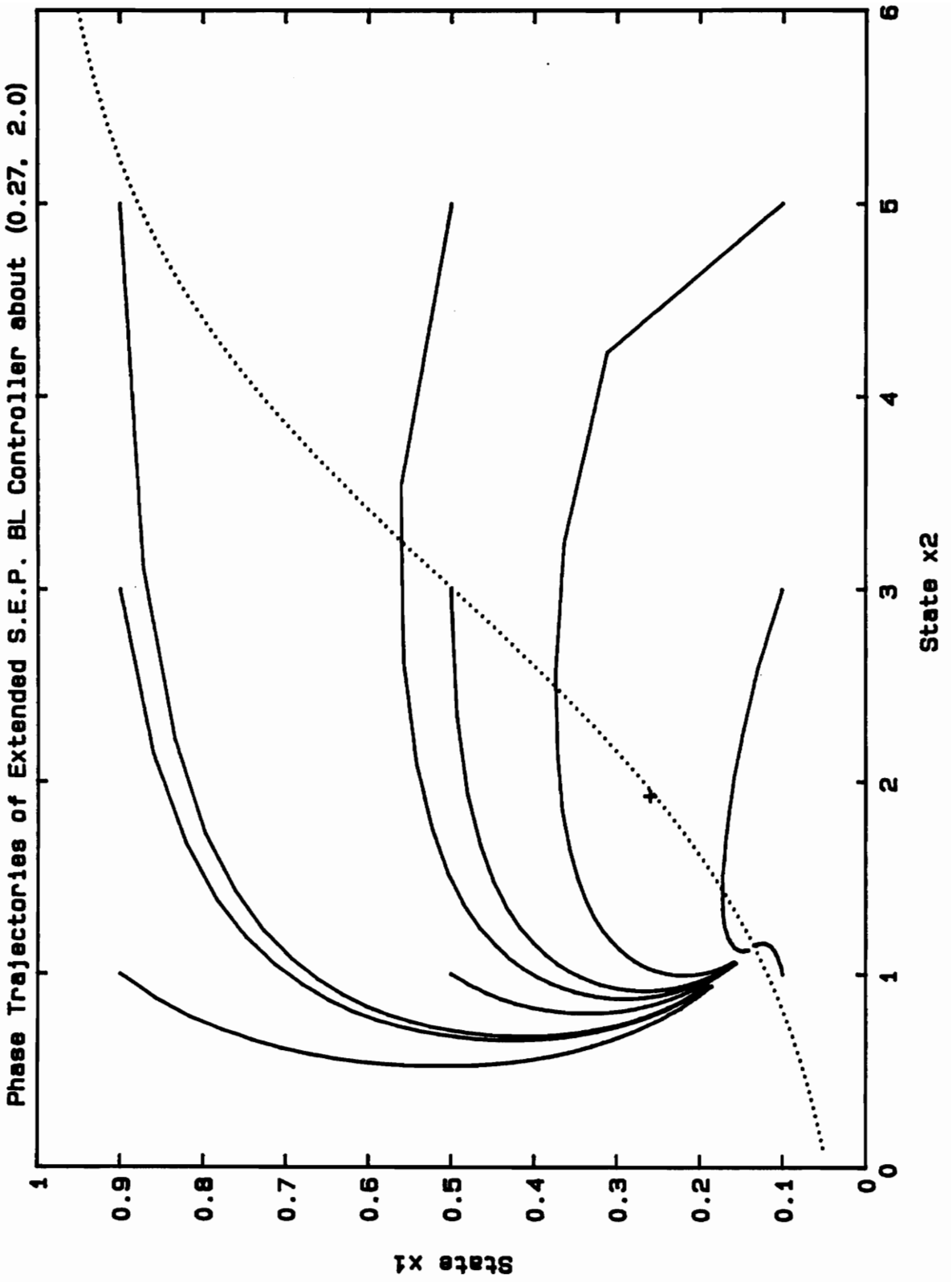


Figure 4.1

The reason for the failure is exactly the same reason why the first and second prototype in chapter three failed. Although the specification of condition (4.4) results in a region of good local performance around an equilibrium grid point, the region is so small that the impact the region has on the system's performance is negligible. The spline structure is unable to combine both the global information and the local information and get adequate results.

4.2 V.E.P. Baseline Controller

The next approach departs from the simple spline structure used by the extended S.E.P. baseline controller. The V.E.P. baseline controller first establishes good local performance by using the form of the extended-linear controller, i.e.

$$S(\mathbf{X}, \mathbf{W}) = U^o(\mathcal{F}(\mathbf{Z})) + \sum_{i=m+1}^{m+n} K_i(\mathcal{F}(\mathbf{Z})) \cdot (z_i - z_i |_{\substack{\mathbf{X}=\mathbf{X}^o(\mathcal{F}(\mathbf{Z})) \\ \mathbf{W}=\mathcal{F}(\mathbf{Z})}})} \quad (4.31)$$

The desired global information will be splined to the extended-linear controller to give the desired global performance without ruining the local performance. A key difference between the V.E.P. baseline controller and the extended S.E.P. baseline controller is that away from the grid points, the V.E.P. baseline controller has structure established by equation (4.31) which the simple spline approach lacks.

The extended-linearization theory specifies that the first partials of the control function must

match G_1 , G_2 , and H are on the equilibrium curve, i.e. (4.2), (4.3), and (4.4) must be satisfied. However, away from the equilibrium curve, conditions (4.2), (4.3), and (4.4) do not apply. Therefore, the global information can be incorporated into one of the gain terms to establish the desired global performance. For example, let us pick H as the term that will incorporate the global information. At any particular point in state space, the value of H will consist of two parts, the local value determined by equation (2.26b), denoted " ℓ ", and the global value determined by a global spline structure, denoted " g ". These two values will be added together to give the overall gain on the external input. The global spline structure should equal zero along the equilibrium curve so that condition (4.4) is still satisfied, i.e. $g = 0$ along the equilibrium curve.

Therefore at the equilibrium grid points, the values, $g_{i_e j_e}$, are zero. At the other grid points, the value $g_{i,j}$ is determined that will provide adequate global control, such as pointing the trajectories in certain directions. Then the values, $\ell_{i,j}$, determined from equation (2.26b), are calculated at each grid point and are subtracted from the global values, $g_{i,j}$. The resulting values, $h_{i,j} = g_{i,j} - \ell_{i,j}$, are then splined together. The spline structure is incorporated into the extended-linear controller, equation (4.31), to give an overall nonlinear controller. For the continuously stirred tank reactor, the spline structure is incorporated into equation (2.35) giving the following nonlinear controller

$$\begin{aligned} \mathbf{S}(\mathbf{X}^o, \mathbf{W}^o) = & u^o(x_2) - G_1(x_2) \cdot (x_1 - x_1^o(x_2)) + \\ & \left[D_2 \mathbf{S}(\mathbf{X}^o(x_2), x_2) + U_g(x_1, x_2) \right] \cdot (\mathbf{w} - x_2) \end{aligned} \quad (4.32)$$

where

$$U_g(x_1, x_2) = \left[\sum_{i=1}^3 \sum_{j=1}^3 h_{ij} \cdot c_{ij}^1(x_1) \cdot c_{ij}^2(x_2) \right] \quad (4.33)$$

Note that the global criterion for this example was to point the trajectory of the system toward the middle equilibrium grid point $\mathbf{X}^o = [0.5 \ 3.0]^T$. This requires the global control information to be a function of only x_1 and x_2 . For other systems, a more complicated global criterion might be required such that at any grid point, the trajectory of the system might point in different directions for different external input

values. Therefore, the global control information might also need the external input to be a dependent variable, i.e.

$$U_g(x_1, x_2, w) = \left[\sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K h_{ijk} \cdot c_{ij}^1(x_1) \cdot c_{ij}^2(x_2) \cdot c_{ij}^3(w) \right] \quad (4.34)$$

For equation (4.33), $c_{ij}^1(x_1)$ and $c_{ij}^2(x_2)$ do not uniquely determine a control law. To uniquely determine a control law, the cardinal cubic splines are determined using the same techniques as the S.E.P. baseline controller. This implies that the partial derivatives with respect to the state variables of the global control information, $\partial U_g(x_1, x_2)/\partial x_1$ and $\partial U_g(x_1, x_2)/\partial x_2$, must be zero at the equilibrium grid points. Also, the second partial derivatives are used to minimize the effect of a grid point on other grid points adjacent to them. Again assume the equilibrium point occurs at the grid point with index (i_e, j_e) , then $c_{ij}^1(x_1)$ is taken to be the unique piecewise cubic polynomial that satisfies

$$c_{ij}^1(x_1^k) = \delta_{ik} \quad (4.35)$$

$$\frac{\partial c_{ij}^1(x_1^k)}{\partial x_1} = 0 \quad \text{for } \{ i = i_e - 1 \text{ or } i_e + 1 \} \text{ and } \{ k = i_e \} \quad (4.36)$$

$$\frac{\partial^2 c_{ij}^1(x_1^k)}{\partial x_1^2} = 0 \quad \text{for } \{ i = i_e - 1 \text{ or } i_e + 1 \} \text{ and } \{ k = i_e \} \quad (4.37)$$

$$\frac{\partial c_{ij}^1(x_1^k)}{\partial x_1} = 0 \quad \text{for } \{ i = i_e \} \text{ and } \{ k = i_e - 1 \text{ or } i_e + 1 \} \quad (4.38)$$

$$\frac{\partial^2 c_{ij}^1(x_1^k)}{\partial x_1^2} = 0 \quad \text{for } \{ i = i_e \} \text{ and } \{ k = i_e - 1 \text{ or } i_e + 1 \} \quad (4.39)$$

$$c_{i_e j_e}^1(x_1^k) \triangleq \text{natural cubic splines for } \{ i \neq i_e - 1, i_e, \text{ or } i_e + 1 \} \quad (4.40)$$

for $1 \leq k \leq I$. The $c_{ij}^2(x_2)$ are defined in a similar fashion for $1 \leq k \leq J$. Note that since $g_{i_e j_e} = 0$, the spline functions, $c_{i_e j_e}^1(x_1)$ and $c_{i_e j_e}^2(x_2)$, have no influence on the spline structure and are not defined in the above specifications. Also note that in equation (4.33), $U_g(x_1, x_2)$ may not necessarily be zero on the equilibrium curve in between the equilibrium grid points. However, by using the cardinal spline conditions (4.35) to (4.40), the spline structure will generally be close to zero and have a

negligible affect. If the affect is not negligible, then additional equilibrium grid points could be chosen to force $U_g(x_1, x_2)$ to be negligible at the intermediate equilibrium points.

As with the S.E.P. baseline controller, the V.E.P. baseline controller need spline only the global information truly needed. The structure of the extended-linear controller extends throughout the state space, and only where the extended-linear controller fails does global information need to be specified. The values, h_{ij} , should be set to zero where the extended-linear controller has sufficient control to provide adequate global performance.

Returning to the continuously stirred tank reactor, the region of global “fix-up” is only needed at $\mathbf{X} = [0.1 \ 5.0]^T$, $\mathbf{X} = [0.5 \ 5.0]^T$, and $\mathbf{X} = [0.9 \ 5.0]^T$. In all other places, the extended-linear controller is sufficient to provide adequate global control. The form of the controller is given by equation (4.32) where the global information is contained in equation (4.33). However, cardinal splines are only needed for the three grid points already mentioned. Therefore the form of equation (4.33) is given by

$$U_g(x_1, x_2) = h_{13} \cdot c_{13}^1(x_1) \cdot c_{13}^2(x_2) + h_{23} \cdot c_{23}^1(x_1) \cdot c_{23}^2(x_2) + h_{33} \cdot c_{33}^1(x_1) \cdot c_{33}^2(x_2) \quad (4.41)$$

where $h_{13} = -6.73$, $h_{23} = -5.25$, and $h_{33} = -5.0$. The cardinal cubic splines $c_{13}^1(x_1)$, $c_{13}^2(x_2)$, $c_{23}^1(x_1)$, and $c_{23}^2(x_2)$ are the same functions determined by equations (3.70) through (3.92). Assuming that $x = x_1$ and $y = x_2$, then $c_{33}^1(x)$ and $c_{33}^2(y)$ are determined by solving the following set of linear equations

$$\text{for } 0.1 \leq x < 0.5: \quad c_{33}^1(x) = 0 \quad (4.42)$$

for $0.5 \leq x \leq 0.9$:

$$c_{33}^1(x_2) = ax_2^3 + bx_2^2 + cx_2 + d = 0 \quad (4.43)$$

$$c_{33}^1(x_3) = ax_3^3 + bx_3^2 + cx_3 + d = 1 \quad (4.44)$$

$$\frac{dc_{33}^1(x_2)}{dx} = 3 \cdot ax_2^2 + 2 \cdot bx_2 + c = 0 \quad (4.45)$$

$$\frac{d^2 c_{33}^1(x_2)}{dx^2} = 6 \cdot ax_2 + 2 \cdot b = 0 \quad (4.46)$$

where $x_1 = 0.1$, $x_2 = 0.5$ and $x_3 = 0.9$ in all the above equations, and

for $1.0 \leq y < 3.0$: $c_{33}^2(y) = 0 \quad (4.47)$

for $3.0 \leq y \leq 5.0$:

$$c_{33}^2(y_2) = ay_2^3 + by_2^2 + cy_2 + d = 0 \quad (4.48)$$

$$c_{33}^2(y_3) = ay_3^3 + by_3^2 + cy_3 + d = 1 \quad (4.49)$$

$$\frac{dc_{33}^2(y_2)}{dy} = 3 \cdot ay_2^2 + 2 \cdot by_2 + c = 0 \quad (4.50)$$

$$\frac{d^2 c_{33}^2(y_2)}{dy^2} = 6 \cdot ay_2 + 2 \cdot b = 0 \quad (4.51)$$

where $y_1 = 1.0$, $y_2 = 3.0$ and $y_3 = 5.0$ in the above equations.

For a value of $w = 0.078$, the same value used for the extended S.E.P. baseline controller, the phase trajectories are shown in figure 4.2. As seen in the figure, the system's state trajectories end at the correct equilibrium point, $\mathbf{X}^\circ = [0.27 \ 2.0]^\top$. Further simulations of the system verify that the system tracks the input exceptionally well in the region of interest of the state-control space and still maintains good global performance. Additionally, the desired local performance is maintained along the equilibrium curve from $\mathbf{X}^\circ = [0.12 \ 1.0]^\top$ to $\mathbf{X}^\circ = [0.5 \ 3.0]^\top$. As the equilibrium point progresses past $\mathbf{X}^\circ = [0.5 \ 3.0]^\top$, the local performance which does not extend to a sufficiently large area, is modified more and more due to the influence of the global specifications, i.e. the components of equation (4.33). Figure 4.3 illustrates that the trajectories of the system using the linear controller (the dashed line) match locally the trajectories of the system using the V.E.P. baseline controller (the solid line) near the equilibrium point $\mathbf{X}^\circ = [0.5 \ 3.0]^\top$. Although the global specifications cause a difference between the two trajectories, a difference also exists due to the extended-linear controller

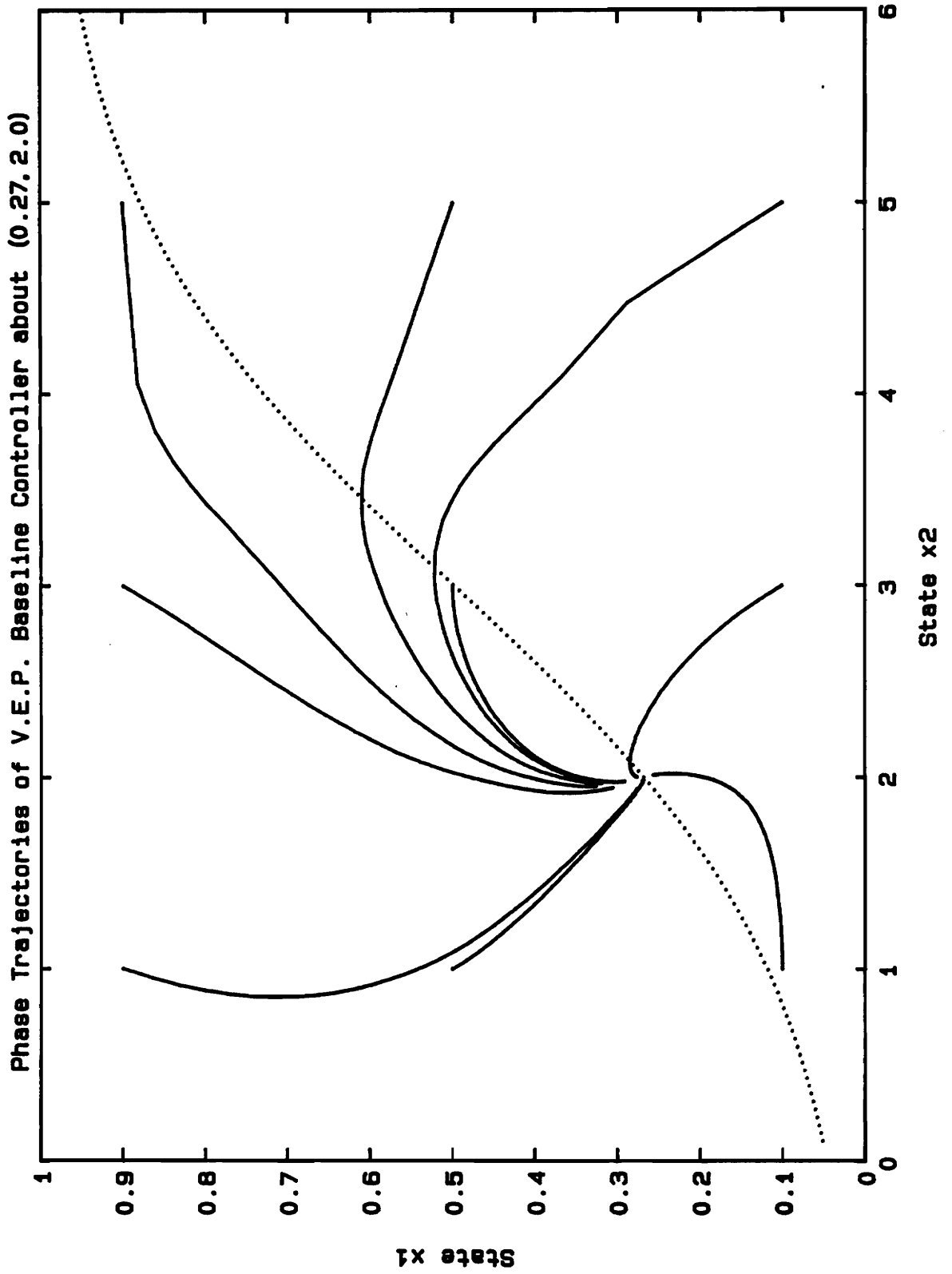


Figure 4.2

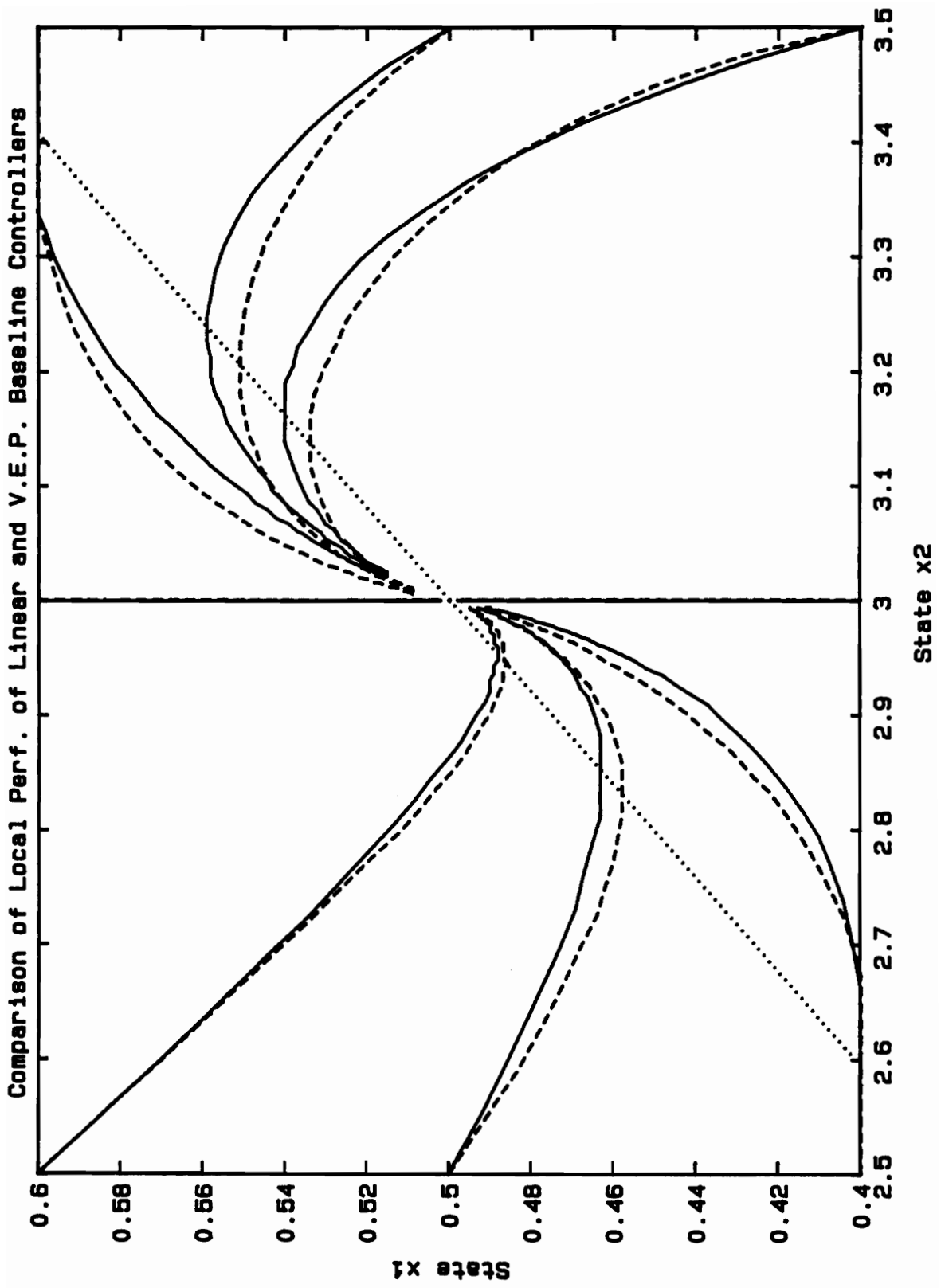


Figure 4.3

using linear splines to approximate some of the functional components of equation (4.32). Refer to section 2.2. This is especially noticeable in the left part of the figure, when $x_2 < 3$, where the global information is zero and has no effect on the local performance.

The desired local performance can easily be maintained at any point along the equilibrium curve if the region of interest is changed to accommodate it. For example, if a certain local performance is wanted at $\mathbf{X}^o = [0.73 \ 4.0]^T$, then the grid in figure 3.1 should be extended to the right from a value of $x_2 = 5.0$ to $x_2 = 6.0$.

5.0 CONCLUSION

The task to unify global and local performance into one generic controller is accomplished by using the extended-linearization controller as the framework and by using splines to incorporate global information into one or more of the gain terms. Although this thesis discussed only one system with two states and one input, this methodology can be extended to multiple input, multiple output systems. The disadvantage for this controller is that the spline structure adds a significant degree of complexity to an already complicated extended-linear controller. As the number of inputs and states increases, even a coarse spline structure becomes very cumbersome. A redeeming characteristic is that this controller allows the designer to spline global information only where truly needed in the state-control space. The rest of the spline structure equals zero.

The complexity of designing this type controller might be overcome if the methodology is programmed into a computer-aided-design package. No one “button” would solve for the best possible controller, but the CAD package would enable the designer to make educated decisions to tailor the

controller into the acceptable solution. This would require the designer to know important facts about the system such as the shape of the equilibrium curve or surface and where global control is needed. The designer could then choose, using a trial-and-error procedure, which cardinal splines would best link the global grid points. Once the solution has been developed, the controller would give the desired global and local performance unified into one nonlinear controller.

This research has also shown two major limitations of using spline techniques to incorporate global information. First, a “stand alone” interpolative spline structure using a coarse grid provides control that is too local to the grid points, and more structure is needed to provide the desired local performance. Second, this research has shown that cardinal splines, the natural extensions of one-dimensional splines to higher dimensions, are not capable of representing a simple linear controller.

List of References

- [1] Baumann, W. T., "Discrete-Time Control of Continuous-Time Nonlinear Systems," The International Journal of Control, to appear in 1990.
- [2] Baumann, W. T., "Feedback Control of Multi-Input Nonlinear Systems by Extended Linearizations," IEEE Transactions on Automatic Control, Vol. AC-33, No. 2, 1988, pp. 193-197.
- [3] Baumann, W. T., and Rugh, W. J., "Feedback Control of Nonlinear Systems by Extended-Linearization," IEEE Transactions on Automatic Control, Vol. AC-31, No. 1, 1986, pp. 40-46.
- [4] Johnson, L. W., Riess, R. D., Numerical Analysis, 2nd Ed., Reading, Massachusetts: Addison-Wesley Publishing Company, 1982, pp. 237-254.
- [5] Kuo, B. C., Automatic Control Systems, 4th Ed., Englewood-Cliffs, N.J.: Prentice-Hall, 1982, pp. 190-192.
- [6] Stewart, C. G.; Baumann, W. T., "Incorporating Global Information into Local Nonlinear Controllers," Proceedings of The IEEE International Conference on Systems Engineering, Pittsburgh, Pennsylvania, 1990, pp. 581-584.
- [7] Su, R., "On the Linear Equivalents of Nonlinear Systems," Systems & Control Letters, Vol. 2, 1982, pp. 48-52.
- [8] Hunt, L.R.; Su, R.; Meyer, G., "Design for Multi-input Nonlinear Systems," Differential Geometric Control Theory, Birkhauser, N.Y., *Proc. Conf. MTU*, 1982, pp. 268-298.
- [9] Hunt, L.R.; Su, R.; Meyer, G., "Global Transformations of Nonlinear Systems," IEEE Transactions on Automatic Control, Vol. AC-28, 1983, pp. 24-30.

- [10] Chen, D.; Tarn, T.; Isidori, A., "Global External Linearization on Nonlinear Systems Via Feedback," IEEE Transactions on Automatic Control, Vol. AC-30, 1985, pp. 808-811.
- [11] Claude, D.; Fliess, M.; Isidori, A., "Immersion, Directe et par Bouclage, d'un Systeme Non Lineaire dans un Lineaire," C.R. Acad. Sc. Paris, Serie I 296, 1983, pp. 237-240.
- [12] Porter, W. A., "Diagonalization and Inverses for Nonlinear Systems," International Journal of Control, Vol. 11, 1970, pp. 67-76.
- [13] Singh, S. N.; Rugh, W.J., "Decoupling in a Class of Nonlinear Systems by State Variable Feedback," Journal of Dynamic Systems, Measurement, and Control, 1972, pp. 323-329.
- [14] Singh, S. N.; Schy, A. A., "Nonlinear Decoupled Control Synthesis for Maneuvering Aircraft" Proceedings of the 1978 IEEE Conference on Decision and Control, San Diego, CA, 1978, pp. 360-370.
- [15] Grayson, L.P., "The Status of Synthesis Using Lyapunov's Method," Automatica, Vol. 3, 1965, pg. 65.
- [16] Mekel, R.; Peruo, P., "Design of Controllers for a Class of Nonlinear Control Systems," IEEE Transactions on Automatic Control, Vol. AC-17, 1972, pg. 206.
- [17] Grishin, S. A.; Krut'ko, P.D.; Popov, E. P., "Synthesis of Nonlinear Controls Using Lyapunov Functions," Soviet Physics - Doklady, Vol. 29, 1984, pg. 778.

APPENDIX

I. LINEAR CONTROLLER ABOUT (0.5, 3.0)

This program is a simulation using the linear controller. This program gives data for figure 2.2.

```
REAL X(2), PARAM(50)
EXTERNAL IVPRK, FCN
OPEN(9, FILE='D0005.DAT')
```

C

c+++Read in the number of points that should be printed.

```
WRITE(*,10002)
READ(*,*) Nsteps
STEPS = FLOAT (Nsteps)
```

c+++Vary initial conditions in region of interest.

```
DO J = 1, 3
DO I = 1, 3
```



```
X(1) = 0.1 + 0.4*FLOAT(J-1)
```

```
X(2) = 1.0 + 2.0*FLOAT(I-1)
```

```
c+++set up values for routine IVPRK.
```

```
NEQ = 2
```

```
T = 0.0
```

```
TOL = 0.0005
```

```
PARAM(10) = 1.0
```

```
PARAM(4) = 1000
```

```
IDO = 1
```

```
C+++Begin iterations to solve for the phase trajectories.
```

```
DO KSTEP = 1, NSTEPS+1
```

```
c----->CALCULATE CONTROLLER<-----
```

```
U = -2.9892 - 2.667*X(1) + 1.552*X(2)
```

```
WRITE(*, '(4F12.3)') T, X, U
```

```
WRITE(9, '(4F12.3)') T, X, U
```

```
c----->call IMSL routine to solve DE's<-----
```

```
DO ISTEP = 1, 2
```

```
TEND = FLOAT(KSTEP-1)/(STEPS/2.0)+ FLOAT(ISTEP)/STEPS
```

```
CALL IVPRK(IDO,NEQ,FCN,T,TEND,TOL,PARAM,X)
```

```
IF (X(2) .GT. 5.5) THEN
```

```
DO LSTEP = 1, 7
```

```
TEND = FLOAT(KSTEP-1)/(STEPS/2.0) + FLOAT(ISTEP)/STEPS +
```

```
& FLOAT(LSTEP)/(8*STEPS)
```

```
CALL IVPRK(IDO,NEQ,FCN,T,TEND,TOL,PARAM,X)
```

```

                END DO
            ENDIF
        END DO
    END DO
c——Release work space for new initial conditions.——
        IDO = 3
        CALL IVPRK(IDO,NEQ,FCN,T,TEND,TOL,PARAM,X)
    END DO
    END DO
    CLOSE (9)
10002  FORMAT(1X, 'HOW MANY POINTS SHOULD BE GRAPHED? '$)
    END
C*****
    SUBROUTINE FCN(NEQ,T,X,XPRIME)
    REAL X(NEQ), XPRIME(NEQ)
    T = T
    U = -2.9892 -2.667*X(1) + 1.552*X(2)
C
    TEMP1 = 1.0 + X(2)/2100.0
    TEMP2 = EXP(X(2)/ TEMP1)
    XPRIME(1) = -X(1) + 0.05 * (1.0 - X(1)) * TEMP2
    XPRIME(2) = -X(2) + 0.4 * (1.0 - X(1)) * TEMP2 - X(2)*U
    RETURN
    END

```

II. Approximated TUNED EXTENDED-LINEAR CONTROLLER

This program is a simulation using a approximate extended-linear controller. The program gives data for figure 2.5. The gains are tuned to give better global performance.

```
DIMENSION X(2), PARAM(50)
```

```
COMMON W, U
```

```
EXTERNAL IVPRK, FCN, INTERP
```

```
OPEN (9, FILE= 'd4041.dat')
```

```
c+++Read in the number of points that should be printed.
```

```
WRITE(*,10002)
```

```
READ(*,*) Nsteps
```

```
STEPS= FLOAT(NSTEPS)
```

```
c++++Vary initial conditions in region of interest.
```

```
DO I1 = 1, 3
```

```
DO J1 = 1, 3
```

```
IF (i1 .eq. 2 .AND. j1 .eq. 2) CYCLE
```

```
X(1) = 0.1 + 0.4*FLOAT(J1-1)
```

```
X(2) = 1.0 + 2.0*FLOAT(I1-1)
```

```
C+++Set up values for routine IVPRK
```

```
NEQ = 2
```

```
T = 0.0
```

```
TOL = 0.0005
```

```
PARAM(10) = 1.0
```

```
PARAM(4) = 200
```

```
IDO = 1
```

```
W = 3.0
```

C+++The following functions will be approximated by INTERP routine.

```
G1 = 0.0
```

```
G2 = 0.0
```

```
UsubO = 0.0
```

```
DusubO = 0.0
```

C+++Begin iterations to solve for the phase trajectories.

```
DO 1200 KSTEP = 1, NSTEPS+1
```

```
CALL INTERP (G1, G2, UsubO, DusubO, X(2))
```

c

```
TEMP1 = 1 + X(2)/2100.0
```

```
TEMP2 = EXP(X(2) / TEMP1)
```

```
TEMP3 = 1 + 0.05 * TEMP2
```

```
X1subO =(0.05*TEMP2) /TEMP3
```

```
Dx1sbO = (0.05*TEMP2) / (TEMP1**2 * TEMP3**2)
```

C

```
H = G1 * Dx1sbO + G2 + DusubO
```

C

```
U = UsubO - G1*(X(1) - X1subO) + H* (W - X(2))
```

c+++limit control value to less than 10.

```
IF (U .GT.10) THEN
```

```
U = 10.0
```

```
ENDIF
```

```

WRITE(*, '(4F12.3)') T, X, U
WRITE(9, '(4F12.3)') T, X, U

DO 1050 ISTEP = 1, 2
    TEND = FLOAT(KSTEP-1)/(STEPS/2.0)+ FLOAT(ISTEP)/STEPS
    CALL IVPRK(IDO,NEQ,FCN,T,TEND,TOL,PARAM,X)
IF (X(2) .GT. 5.5) THEN
    DO 1000 LSTEP = 1, 7
        TEND = FLOAT(KSTEP-1)/(STEPS/2.0) + FLOAT(ISTEP)/STEPS +
&          FLOAT(LSTEP)/(8*STEPS)
        CALL IVPRK(IDO,NEQ,FCN,T,TEND,TOL,PARAM,X)
1000    CONTINUE
    ENDIF
1050 CONTINUE
1200 CONTINUE

    IDO = 3
    CALL IVPRK(IDO,NEQ,FCN,T,TEND,TOL,PARAM,X)
END DO
END DO
CLOSE(9)
10002 FORMAT(1X, 'HOW MANY POINTS SHOULD BE GRAPHED? '$)
END

```

C*****

```
SUBROUTINE FCN(NEQ,T,X,XPRIME)
REAL X(NEQ), XPRIME(NEQ)
COMMON W, U
EXTERNAL INTERP
T = T
CALL INTERP (G1, G2, UsubO, DusubO, X(2))
```

```
TEMP1 = 1 + X(2)/2100.0
TEMP2 = EXP(X(2) / TEMP1)
TEMP3 = 1 + 0.05 * TEMP2
```

C

```
X1subO =(0.05*TEMP2) /TEMP3
Dx1sbO = (0.05*TEMP2) / (TEMP1**2 * TEMP3**2)
H = G1 * Dx1sbO + G2 + DusubO
U = UsubO - G1*(X(1) - X1subO) + H* (W - X(2))
```

c

```
IF (U .GT.10) THEN
    U = 10.0
ENDIF
```

C-----

```
TEMP1 = 1.0 + X(2)/2100.0
TEMP2 = EXP(X(2)/ TEMP1)
```

C

```
XPRIME(1) = -X(1) + 0.05 * (1.0 - X(1)) * TEMP2
XPRIME(2) = -X(2) + 0.4 * (1.0 - X(1)) * TEMP2 - X(2)*U
```

C

```
RETURN
END
```

C*****

```
SUBROUTINE INTERP(GAIN1, GAIN2, UsubO, DusubO, X2)
DIMENSION G1(5), G2(5), Uo(5), dUo(5)
```

c++++This subroutine uses linear splines to approximate the functions:

c G1(x2), G2(x2), UsubO(x2), DusubO(x2).

G1(1) = 5.16286

G1(2) = -0.73404

G1(3) = -2.66663

G1(4) = -5.38940

G1(5) = -10.48990

G2(1) = 2.86324

G2(2) = 1.85239

G2(3) = 1.55178

G2(4) = 2.16275

G2(5) = 1.85310

C

$$U_o(1) = -0.04318$$

$$U_o(2) = 0.07763$$

$$U_o(3) = 0.33333$$

$$U_o(4) = 0.46080$$

$$U_o(5) = 0.40799$$

$$dU_o(1) = -0.1152$$

$$dU_o(2) = 0.2470$$

$$dU_o(3) = 0.2203$$

$$dU_o(4) = 0.0271$$

$$dU_o(5) = -0.1134$$

C

IF (X2 .GT. 2.0) GOTO 10

I = 2

GOTO 50

10 IF (X2 .GT. 3.0) GOTO 20

I = 3

GOTO 50

20 IF (X2 .GT. 4.0) GOTO 30

I = 4

GOTO 50

30 I = 5

50 SLOPE1 = G1(I) - G1(I-1)

$$\text{SLOPE2} = G2(I) - G2(I-1)$$

$$\text{SLOPE3} = Uo(I) - Uo(I-1)$$

$$\text{SLOPE4} = dUo(I) - dUo(I-1)$$

$$\text{YAXIS1} = G1(I-1) - \text{SLOPE1} * \text{FLOAT}(I-1)$$

$$\text{YAXIS2} = G2(I-1) - \text{SLOPE2} * \text{FLOAT}(I-1)$$

$$\text{YAXIS3} = Uo(I-1) - \text{SLOPE3} * \text{FLOAT}(I-1)$$

$$\text{YAXIS4} = dUo(I-1) - \text{SLOPE4} * \text{FLOAT}(I-1)$$

$$100 \quad \text{GAIN1} = - (\text{SLOPE1} * X2 + \text{YAXIS1})$$

$$\text{GAIN2} = - (\text{SLOPE2} * X2 + \text{YAXIS2})$$

$$\text{UsubO} = (\text{SLOPE3} * X2 + \text{YAXIS3})$$

$$\text{DusubO} = (\text{SLOPE4} * X2 + \text{YAXIS4})$$

C

RETURN

END

III. SEP BASELINE CONTROLLER ABOUT (0.5, 3.0)

This is a matlab file that gives file "d0340.inp", i.e. the polynomial coefficients for the SEP baseline controller.

```
clear
```

```
f13 = 11.555;
```

```
f23 = 7.601;
```

```
%-----
```

```
y1 = 0.1;
```

```
y2 = 0.5;
```

```
y3 = 0.9;
```

```
x1 = 1.0;
```

```
x2 = 3.0;
```

```
x3 = 5.0;
```

```
X2 = [x2^3  x2^2 x2  1  
      x3^3  x3^2 x3  1  
      3*x2^2 2*x2 1  0  
      6*x2  2 0  0];
```

```
Y1 = [y1^3  y1^2 y1  1  
      y2^3  y2^2 y2  1  
      3*y1^2 2*y1 1  0  
      6*y1  2 0  0];
```

```
Y2 = [y2^3  y2^2 y2  1  
      y3^3  y3^2 y3  1  
      3*y3^2 2*y3 1  0
```

```

        6*y3  2 0  0];
Y3 = [y1^3  y1^2 y1  1
      y2^3  y2^2 y2  1
      3*y2^2 2*y2 1  0
      6*y2  2 0  0];
la = [0 0 0 0];
lb = (X2\[0 1 0 0])' ;
wa23 = (Y1\[0 1 0 0])' ;
wb23 = (Y2\[1 0 0 0])' ;
wa13 = (Y3\[1 0 0 0])' ;
wb13 = [0 0 0 0];

%-----
S1 = [0  0 0 0
      0  0 0 0
      0  0 0 1.552
      0  0-2.667 -2.9892];

%
S1 = f13*la'*wa13 + f23*la'*wa23 + S1 ;
S2 = f13*la'*wb13 + f32*la'*wb23 + S1 ;
S3 = f13*lb'*wa13 + f23*lb'*wa23 + S1 ;
S4 = f13*lb'*wb13 + f32*lb'*wb23 + S1 ;

save temp S1 S2 S3 S4 /ascii;
!copy temp d0340.inp

```

This program is a simulation using the SEP baseline controller. The program gives data for figure 3.18. The global information comes from data file "d0340.inp" which is generated from a matlab file.

```
DIMENSION X(2), PARAM(50)
COMMONS(4,4,4), U
EXTERNAL IVPRK, FCN
OPEN (8, FILE= 'd0340.inp')
OPEN (9, FILE= 'd0351.dat')
c+++Read in coefficients of polynomials. The linear controller is already
c    included in controller.
      DO 25 K = 1, 4
      DO 20 I = 1, 4
          READ(8,*) (S(K,I,J), J = 1,4)
20    CONTINUE
25    CONTINUE
      CLOSE(8)
c+++Read in the number of points that should be printed.
      WRITE(*,10002)
      READ(*,*) Nsteps
      STEPS= FLOAT(NSTEPS)
c+++Vary initial conditions in region of interest.
      DO 5000 I1 = 1,3
      DO 4000 I2 = 1,3
```

```
IF (I1 .EQ. 3 .AND. I2 .EQ. 1) CYCLE
```

```
IF (I1 .EQ. 2 .AND. I2 .EQ. 2) CYCLE
```

```
X(2) = 1.0 + FLOAT(I1-1)*2.0
```

```
X(1) = 0.1 + FLOAT(I2-1)*0.4
```

```
C+++Set up values for routine IVPRK
```

```
NEQ = 2
```

```
T = 0.0
```

```
TOL = 0.0005
```

```
PARAM(10) = 1.0
```

```
IDO = 1
```

```
C+++Begin iterations to solve for the phase trajectories
```

```
DO 120 KSTEP = 1, NSTEPS+1
```

```
C----->CALCULATE CONTROLLER<-----
```

```
c+++Find out which quadrant (x,y) is located
```

```
IF (X(2) .LT. 3.0) THEN
```

```
IF (X(1) .LT. 0.5) THEN
```

```
    K = 1
```

```
ELSE
```

```
    K = 2
```

```
ENDIF
```

```
ELSE
```

```
IF (X(1) .LT. 0.5) THEN
```

```
    K = 3
```

```
ELSE
```

```

        K = 4
    ENDIF
ENDIF
C+++Calculate polynomial solution
    U = 0.0
    DO 60 I = 1, 4
    DO 55 J = 1, 4
        TEMP = S(K,I,J) * X(2)**(4-I) * X(1)**(4-J)
        U = U + TEMP
55    CONTINUE
60    CONTINUE
c++++limit control value to less than 10.
    IF (U.GT.10.0) THEN
        U = 10.0
    ENDIF
    WRITE(*, '(4F12.3)') T, X, U
    WRITE(9, '(4F12.3)') T, X, U
C-----call imsl routine to solve DE's-----
    DO 100 ISTEP = 1, 2
        TEND = FLOAT(KSTEP-1)/(STEPS/2.0)+ FLOAT(ISTEP)/STEPS
        CALL IVPRK(IDO,NEQ,FCN,T,TEND,TOL,PARAM,X)
100    CONTINUE
120    CONTINUE
C-----

```

c+++clear workspace for another initial condition.

```
      IDO = 3
```

```
      CALL IVPRK(IDO,NEQ,FCN,T,TEND,TOL,PARAM,X)
```

```
4000  CONTINUE
```

```
5000  CONTINUE
```

```
      CLOSE(9)
```

```
10002 FORMAT(1X, 'HOW MANY POINTS SHOULD BE GRAPHED? '$)
```

```
      END
```

```
C*****
```

```
      SUBROUTINE FCN(NEQ,T,X,XPRIME)
```

```
      REAL X(NEQ), XPRIME(NEQ)
```

```
      COMMON S(4,4,4), U
```

```
      T = T
```

c+++Find out which quadrant (x,y) is located

```
      IF (X(2) .LT. 3.0) THEN
```

```
      IF (X(1) .LT. 0.5) THEN
```

```
          K = 1
```

```
      ELSE
```

```
          K = 2
```

```
      ENDIF
```

```
      ELSE
```

```
      IF (X(1) .LT. 0.5) THEN
```

```
          K = 3
```

```
      ELSE
```

```

        K = 4
    ENDIF
ENDIF
C+++Calculate polynomial solution
    U = 0.0
    DO 60 I = 1, 4
    DO 55 J = 1, 4
        TEMP = S(K,I,J) * X(2)**(4-I) * X(1)**(4-J)
        U = U + TEMP
55    CONTINUE
60    CONTINUE
    IF (U.GT.10.0) THEN
        U = 10.0
    ENDIF
C
    TEMP1 = 1.0 + X(2)/2100.0
    TEMP2 = EXP(X(2)/ TEMP1)
C
    XPRIME(1) = -X(1) + 0.05 * (1.0 - X(1)) * TEMP2
    XPRIME(2) = -X(2) + 0.4 * (1.0 - X(1)) * TEMP2 - X(2)*U
C
    RETURN
END

```


IV. VEP BASELINE CONTROLLER

This is a matlab file that gives file "d4030.inp", i.e. the polynomial coefficients for the global H value.

```
clear
```

```
f13 = -6.73;
```

```
f23 = -5.25;
```

```
f33 = -5.0;
```

```
%-----
```

```
y1 = 0.1;
```

```
y2 = 0.5;
```

```
y3 = 0.9;
```

```
x1 = 1.0;
```

```
x2 = 3.0;
```

```
x3 = 5.0;
```

```
X2 = [x2^3 x2^2 x2 1
```

```
      x3^3 x3^2 x3 1
```

```
      3*x2^2 2*x2 1 0
```

```
      6*x2 2 0 0];
```

```
Y1 = [y1^3 y1^2 y1 1
```

```
      y2^3 y2^2 y2 1
```

```
      3*y1^2 2*y1 1 0
```

```
      6*y1 2 0 0];
```

```
Y2 = [y2^3 y2^2 y2 1
```

```
      y3^3 y3^2 y3 1
```

```
      3*y3^2 2*y3 1 0
```

```

        6*y3  2 0  0];
Y3 = [y1^3  y1^2 y1  1
      y2^3  y2^2 y2  1
      3*y1^2 2*y1 1  0
      6*y2  2 0  0];
la = [0 0 0 0];
lb = (X2\[0 1 0 0])';
wa23 = (Y1\[0 1 0 0])';
wb23 = (Y2\[1 0 0 0])';
wa13 = (Y3\[1 0 0 0])';
wb13 = [0 0 0 0];
wa33 = [0 0 0 0];
wb33 = (Y2\[0 1 0 0])';
%-----
%
S1 = f13*la'*wa13 + f23*la'*wa23 + f33*la'*wa33 ;
S2 = f13*la'*wb13 + f23*la'*wb23 + f33*la'*wb33 ;
S3 = f13*lb'*wa13 + f23*lb'*wa23 + f33*lb'*wa33 ;
S4 = f13*lb'*wb13 + f23*lb'*wb23 + f33*lb'*wb33 ;
%
save temp S1 S2 S3 S4 /ascii
!copy temp d4030.inp

```

This program is a simulation using the V.E.P. baseline controller. The program gives data for figure 4.2. The global information comes from data file "d4030.inp" which will be incorporated into the gain term "H".

```
DIMENSION X(2), PARAM(50)

COMMONS(4,4,4), W, U

EXTERNAL IVPRK, FCN, INTERP
```

c

```
OPEN (8, FILE= 'd4030.inp')

OPEN (9, FILE= 'd4040.dat')
```

c+++Read in coefficients of polynomial spline structure for Global H.

```
DO 25 K = 1, 4

DO 20 I = 1, 4

READ(8,*) (S(K,I,J), J = 1,4)

20 CONTINUE

25 CONTINUE

CLOSE(8)
```

c+++Read in the number of points that should be printed.

```
WRITE(*,10002)

READ(*,*) Nsteps

STEPS= FLOAT(NSTEPS)

WRITE(*,10005)

READ(*,*) W
```

c+++Vary initial conditions for simulations.

```
DO J1 = 1, 3
```

```
DO I1 = 1, 3
```

```
X(1) = 0.1 + FLOAT(J1-1)*0.4
```

```
X(2) = 1.0 + FLOAT(I1-1)*2.0
```

```
C+++Set up values for routine IVPRK
```

```
NEQ = 2
```

```
T = 0.0
```

```
TOL = 0.0005
```

```
PARAM(10) = 1.0
```

```
PARAM(4) = 500
```

```
IDO = 1
```

```
G1 = 0.0
```

```
G2 = 0.0
```

```
UsubO = 0.0
```

```
DsubO = 0.0
```

```
C+++Begin iterations to solve for the phase trajectories
```

```
DO 1200 KSTEP = 1, NSTEPS+1
```

```
C----->CALCULATE CONTROLLER<-----
```

```
c+++Find out which quadrant (x,y) is located
```

```
IF (X(2) .LT. 3.0) THEN
```

```
IF (X(1) .LT. 0.5) THEN
```

```
    K = 1
```

```
ELSE
```

```
    K = 2
```

```
ENDIF
```

```

ELSE
IF (X(1) .LT. 0.5) THEN
    K = 3
ELSE
    K = 4
ENDIF
ENDIF

```

C+++Calculate polynomial solution for global H

```

H = 0.0
DO 60 I = 1, 4
DO 55 J = 1, 4
    TEMP = S(K,I,J) * X(2)**(4-I) * X(1)**(4-J)
    H = H + TEMP

```

55 CONTINUE

60 CONTINUE

c--> Call subroutine to interpolate functions for approximate E/L controller<-----

```

CALL INTERP (G1, G2, UsubO, DsubO, X(2))

```

C

```

TEMP1 = 1 + X(2)/2100.0
TEMP2 = EXP(X(2) / TEMP1)
TEMP3 = 1 + 0.05 * TEMP2

```

C

```

X1subO =(0.05*TEMP2) /TEMP3
UsubO = -1.0 + 8.0 * X1subO / X(2)

```

C

```
Dx1sbO = (0.05*TEMP2) / (TEMP1**2 * TEMP3**2)
```

```
DusubO= 8.0*Dx1sbO/X(2) - 8.0*X1subO/(X(2)**2)
```

C

++++This term adds global H to local H.

```
H = H + G1 * Dx1sbO + G2 + DusubO
```

C

```
U = UsubO - G1*(X(1) - X1subO) + H* (W - X(2))
```

+++limit control to a value less than 10.

```
IF (U .GT.10) THEN
```

```
    U = 10.0
```

```
ENDIF
```

```
WRITE(*, '(4F12.3)') T, X, U
```

```
WRITE(9, '(4F12.3)') T, X, U
```

C-----call imsl routine to solve DE's-----

```
DO 1050 ISTEP = 1, 2
```

```
TEND = FLOAT(KSTEP-1)/(STEPS/2.0)+ FLOAT(ISTEP)/STEPS
```

```
CALL IVPRK(IDO,NEQ,FCN,T,TEND,TOL,PARAM,X)
```

```
IF (X(2) .GT. 5.5) THEN
```

```
    DO 1000 LSTEP = 1, 7
```

```
    TEND = FLOAT(KSTEP-1)/(STEPS/2.0) + FLOAT(ISTEP)/STEPS +
```

```
&    FLOAT(LSTEP)/(8*STEPS)
```

```
    CALL IVPRK(IDO,NEQ,FCN,T,TEND,TOL,PARAM,X)
```

```
1000    CONTINUE
```

```

                ENDIF
1050  CONTINUE
1200  CONTINUE
C-----Release work space for new initial conditions-----

                IDO = 3

                CALL IVPRK(IDO,NEQ,FCN,T,TEND,TOL,PARAM,X)

            END DO

        END DO

        CLOSE(9)
10002  FORMAT(1X, 'HOW MANY POINTS SHOULD BE GRAPHED? '$)
10005  FORMAT(1X, 'WHAT IS THE EXTERNAL CONTROL W? '$)

        END

C*****
        SUBROUTINE FCN(NEQ,T,X,XPRIME)

        REAL X(NEQ), XPRIME(NEQ)

        COMMONS(4,4,4), W, U

        EXTERNAL INTERP

        T = T

C----->CALCULATE CONTROLLER<-----

c+++Find out which quadrant (x,y) is located

        IF (X(2) .LT. 3.0) THEN

            IF (X(1) .LT. 0.5) THEN

                K = 1

            ELSE

```

```

        K = 2
ENDIF
ELSE
IF (X(1) .LT. 0.5) THEN
        K = 3
ELSE
        K = 4
ENDIF
ENDIF

```

C+++Calculate polynomial solution

```

        H = 0.0
        DO 60 I = 1, 4
        DO 55 J = 1, 4
        TEMP = S(K,I,J) * X(2)**(4-I) * X(1)**(4-J)
        H = H + TEMP
55      CONTINUE
60      CONTINUE

```

c----->Calculate Controller<-----

```

        CALL INTERP (G1, G2, UsubO, DsubO, X(2))

```

C

```

        TEMP1 = 1 + X(2)/2100.0
        TEMP2 = EXP(X(2) / TEMP1)
        TEMP3 = 1 + 0.05 * TEMP2

```

C


```
X1subO =(0.05*TEMP2) /TEMP3
```

```
Dx1sbO = (0.05*TEMP2) / (TEMP1**2 * TEMP3**2)
```

```
H = H + G1 * Dx1sbO + G2 + DusubO
```

```
U = UsubO - G1*(X(1) - X1subO) + H* (W - X(2))
```

```
c
```

```
IF (U .GT.10) THEN
```

```
U = 10.0
```

```
ENDIF
```

```
C -----
```

```
TEMP1 = 1.0 + X(2)/2100.0
```

```
TEMP2 = EXP(X(2)/ TEMP1)
```

```
C
```

```
XPRIME(1) = -X(1) + 0.05 * (1.0 - X(1)) * TEMP2
```

```
XPRIME(2) = -X(2) + 0.4 * (1.0 - X(1)) * TEMP2 - X(2)*U
```

```
C
```

```
RETURN
```

```
END
```

```
C*****
```

```
SUBROUTINE INTERP(GAIN1, GAIN2, UsubO, DusubO, X2)
```

```
DIMENSION G1(5), G2(5), Uo(5), dUo(5)
```

```
G1(1) = 5.16286
```

```
G1(2) = -0.73404
```

```
G1(3) = -2.66663
```

$$G1(4) = -4.41469$$

$$G1(5) = -2.57403$$

$$G2(1) = 2.86324$$

$$G2(2) = 1.85239$$

$$G2(3) = 1.55178$$

$$G2(4) = 1.16275$$

$$G2(5) = 0.25310$$

C

$$Uo(1) = -0.04318$$

$$Uo(2) = 0.07763$$

$$Uo(3) = 0.33333$$

$$Uo(4) = 0.46080$$

$$Uo(5) = 0.40799$$

$$dUo(1) = -0.1152$$

$$dUo(2) = 0.2470$$

$$dUo(3) = 0.2203$$

$$dUo(4) = 0.0271$$

$$dUo(5) = -0.1134$$

C

IF (X2 .GT. 2.0) GOTO 10

I = 2

GOTO 50

```

10    IF (X2 .GT. 3.0) GOTO 20
      I = 3
      GOTO 50
20    IF (X2 .GT. 4.0) GOTO 30
      I = 4
      GOTO 50
30    I = 5
50    SLOPE1 = G1(I) - G1(I-1)
      SLOPE2 = G2(I) - G2(I-1)
      SLOPE3 = Uo(I) - Uo(I-1)
      SLOPE4 = dUo(I) - dUo(I-1)

      YAXIS1 = G1(I-1) - SLOPE1*FLOAT(I-1)
      YAXIS2 = G2(I-1) - SLOPE2*FLOAT(I-1)
      YAXIS3 = Uo(I-1) - SLOPE3*FLOAT(I-1)
      YAXIS4 = dUo(I-1) - SLOPE4*FLOAT(I-1)

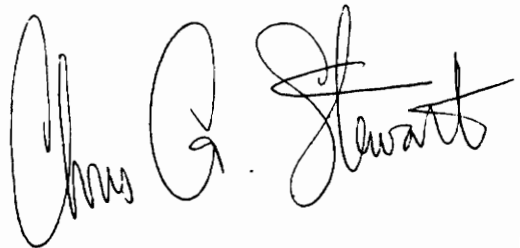
100   GAIN1 = - (SLOPE1 * X2 + YAXIS1)
      GAIN2 = - (SLOPE2 * X2 + YAXIS2)
      UsubO = (SLOPE3 * X2 + YAXIS3)
      DusubO= (SLOPE4 * X2 + YAXIS4)

C
      RETURN
      END

```

Vita

Mr. Stewart was born in July 1963. He lived in Virginia Beach, Virginia for eighteen years before attending Virginia Polytechnic Institute and State University where he received a Bachelor of Science degree in Electrical Engineering in June 1985. In September 1985, he accepted employment as a Security Engineering Officer for the Foreign Service. In March 1986, he relocated to Abidjan, Ivory Coast, West Africa where he worked for two years. While there, he traveled sixty percent of the time to nineteen other West African nations, performing security services for the Department of State at various embassies, consulates, and other American facilities. In April of 1988, he returned to the United States where he enrolled in the Master's program at Virginia Polytechnic Institute and State University.

A handwritten signature in black ink that reads "Chris G. Stewart". The signature is written in a cursive style with a large, stylized "C" and "S".