## INFORMATION TO USERS

This reproduction was made from a copy of a manuscript sent to us for publication and microfilming. While the most advanced technology has been used to photograph and reproduce this manuscript, the quality of the reproduction is heavily dependent upon the quality of the material submitted. Pages in any manuscript may have indistinct print. In all cases the best available copy has been filmed.

The following explanation of techniques is provided to help clarify notations which may appear on this reproduction.

1. Manuscripts may not always be complete. When it is not possible to obtain missing pages, a note appears to indicate this.

2. When copyrighted materials are removed from the manuscript, a note appears to indicate this.

3. Oversize materials (maps, drawings, and charts) are photographed by sectioning the original, beginning at the upper left hand corner and continuing from left to right in equal sections with small overlaps. Each oversize page is also filmed as one exposure and is available, for an additional charge, as a standard 35mm slide or in black and white paper format.*

4. Most photographs reproduce acceptably on positive microfilm or microfiche but lack clarity on xerographic copies made from the microfilm. For an additional charge, all photographs are available in black and white standard 35mm slide format.*

***For more information about black and white slides or enlarged paper reproductions, please contact the Dissertations Customer Services Department.**

**Boadu, Herbert Odame**

CONTINUOUS-TIME OPTIMAL CONTROL OF A SIMULATED BOILING WATER NUCLEAR (BWR) POWER PLANT

*The University of Arizona* PH.D. 1985

# CONTINUOUS-TIME OPTIMAL CONTROL OF A SIMULATED BOILING WATER NUCLEAR (BWR) POWER PLANT

by

Herbert Odame Boadu

---

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF NUCLEAR AND ENERGY ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY
WITH A MAJOR IN NUCLEAR ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

1 9 8 5

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Final Examination Committee, we certify that we have read the dissertation prepared by Herbert Odame Boadu

entitled Continuous-Time Optimal Control of a Simulated Boiling Water Nuclear Power Plant

and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

| | |
|---|---|
| David L. Hetrick | 27 November 1985 |
| | Date |
| Robert K. Leale | 27 November 1985 |
| | Date |
| W. Morin Fan | 27 November, 1985 |
| | Date |
| François E. Cellier | November 27, 1985 |
| | Date |
| Øyvind A. Paulsrud | November 27, 1985 |
| | Date |

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copy of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

| | |
|---|---|
| David L Hetrick | 27 November 1985 |
| Dissertation Director | Date |

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Herbert Odame Boadu

ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. D. L. Hetrick, my research advisor, for his constant and continued support. I am very grateful to Dr. Francois E. Cellier of the Department of Electrical and Computer Engineering for his outstanding interest and invaluable guidance throughout this work.

Instituto de Investigaciones Electricas, Cuernavaca, Mexico, is commended for providing data for the Laguna Verde Nuclear Power Plant Model.

Finally, the author is deeply grateful to Dr. Robert Seale, Chairman of the Department of Nuclear and Energy Engineering, and all the faculty and staff for their support through teaching assistantships and computer time. Above all, I thank God for making this work possible.

TABLE OF CONTENTS

TABLE OF CONTENTS--Continued

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS--Continued

LIST OF TABLES

## LIST OF VARIABLES

| Symbol | Parameter | Units |
|---|---|---|
| A, A(t) | Continuous time linear model system matrix | |
| $A_B$ | Total available bypass valve area | $ft^2$ |
| $A_c$ | Effective area of core at steam separators | $ft^2$ |
| $A_t$ | Total available turbine valve area | $ft^2$ |
| $a_B$ | Bypass valve opening | |
| $a_{BSP}$ | Bypass valve setpoint | |
| $a_t$ | Turbine valve opening | |
| $a_{tsp}$ | Turbine valve setpoint | |
| B, B(t) | Continuous time linear model control matrix | |
| $C_T$ | Turbine flow coefficient | |
| $h_c$ | Core inlet enthalpy | BTU/lbm |
| $h_f$ | Water saturation enthalpy | BTU/lbm |
| $h_F$ | Feedwater enthalpy | BTU/lbm |
| $h_g$ | Steam saturation enthalpy | BTU/lbm |
| $h_{fg}$ | $h_g - h_f$, water vaporization enthalpy | BTU/lbm |
| $h_{sub}$ | $h_f - h_c$, core inlet subcooling | BTU/lbm |
| $H_c$ | Water level in the pressure vessel | in. |
| I | Unit matrix, order apparent from text | |
| J | Mechanical equivalent of heat | |

| Symbol | Parameter | Units |
|---|---|---|
| K | Feedback gain matrix | |
| $m_f$ | Mass of saturated water in vessel | lbm |
| $m_{fc}$ | Mass of saturated water in core | lbm |
| $m_g$ | Mass of steam in vessel | lbm |
| $m_{gc}$ | Mass of steam in core | lbm |
| $m_{fp}$ | Mass of water in core exit plenum | lbm |
| $m_{gp}$ | Mass of steam in core exit plenum | lbm |
| $P_B$ | Pressure at inlet to HP-turbine | psi |
| $P_T$ | Pressure at end of steam line | psi |
| $P_V$ | Pressure in reactor vessel | psi |
| $q_f$ | Core average heat flux from fuel rod to moderator | $\frac{Btu}{hr\text{-}ft^2}$ |
| $q_T$ | Total power output from turbines as measured by sensors, relative to rated value | |
| Q | State weighting matrix, positive semi-definite | |
| $Q_f$ | Convective heat transfer rate from fuel rod to moderator | Btu/sec |
| $Q_{fR}$ | Rated value of $Q_f$ | BTU/sec |
| $Q_{sat}$ | Rate of heat transfer to saturated part of core | BTU/sec |
| $Q_{sub}$ | Rate of heat transfer to subcooled part of core | BTU/sec |
| R | Control weighting matrix, positive definite | |
| $T_f$ | Core average fuel temperature | $^oF$ |
| $u_f$ | Specific internal energy for water | BTU/lbm |

| Symbol | Parameter | Units |
|---|---|---|
| $u_g$ | Specific internal energy for steam | BTU/lbm |
| $v_f$ | Saturated water specific volume at vessel pressure | $ft^3$/lbm |
| $v_g$ | Saturated steam specific volume at vessel pressure | $ft^3$/lbm |
| $v_g(P)$ | Saturated steam specific volume at pressure p | $ft^3$/lbm |
| $v_{sub}$ | Specific volume of vessel subcooled water | $ft^3$/lbm |
| $V_c$ | Volume of saturated mass in core | $ft^3$ |
| $V_{cT}$ | Total active core volume | $ft^3$ |
| $V_d$ | Volume of vessel dome available for saturated mass | $ft^3$ |
| $w_{BT}$ | Steam flow through the bypass valve | lbm/sec |
| $w_c$ | Core mass flow | lbm/sec |
| $w_{ce}$ | Core exit mass flow | lbm/sec |
| $W_{CT}$ | Steam flow through the turbine valve | lbm/sec |
| $W_d$ | Mass flow through one recirculation loop | lbm/sec |
| $W_{dw}$ | Downcomer mass flow | lbm/sec |
| $W_F$ | Feedwater flow entering the vessel | lbm/sec |

ABSTRACT

A suboptimal controller has been developed for a Boiling Water Reactor Nuclear Power Plant, using the DARE P Continuous Simulation language which was developed in the Electrical Engineering Department at the University of Arizona. A set of 48 nonlinear first-order differential equations and a large number of algebraic equations has been linearized about the equilibrium state. Using partitioning, the linearized equations were transformed into a block triangular form.

The concept of optimal control and a square performance index reflecting the desired plant behavior have been applied on the slow subsystem to develope a suboptimal controller. The obtained feedback law is shown by simulation to be able to compensate for a variety of plant disturbances. A large variety of responses can be obtained by changing the weighting matrices.

The control is basically a regulator approach to speed up response during load demand changes. Several simulations are included to demonstrate the control performance. The variables to be controlled have mainly been the average neutron density and the average coolant temperature.

Simplifications have been suggested, thus obtaining considerable savings in the computations and ease in design.

# CHAPTER 1

# INTRODUCTION

## 1.1 General Review

Present generation nuclear plants, which typically are designed to produce about 1000 Mwe per unit, are among the most complex and expensive power systems that are built today. The design and operation of these plants is dependent upon a highly sophisticated technology. There will probably be a continuous effort for the next one or two decades to look for improvements in performance and flexibility of these plants. However, the issues of safety and standardization will, by their very nature, introduce conservatism in adopting new technology.

Closely connected to a technology is the theory that is to guide the engineer in how to make the best use of the device he has available. The theory most used in control of industrial processes today was developed largely during the 1940s and 1950s. The Ziegler-Nichols (see, e.g., Takahashi et al., 1970), empirical approach to the optimum settings of the feedback controllers was developed during World War II, and is still used by many practicing engineers, particularly in the process control industry. Laplace domain and frequency response methods were developed during the 1940s

and early 1950s, and this strengthened the scientific study of feedback control problems. Soon after this, root-locus methods were developed to study the poles of the feedback system as a function of a parameter, e.g., feedback gain. During the years, a variety of methods has been developed based upon these approaches, and these are now often labelled as "classical control theory" (Gille et al., 1959).

Classical control theory has played a most important role in solution of most feedback control problems in the last three decades, and it has been readily implemented through analog circuitry and servomechanisms that also have seen a tremendous development in the last decades.

Sampled data theory, which is an outgrowth of the Laplace and frequency domain approach, was developed during the late 1950s and early 1960s, and it was envisioned to fill the need for theory when using digital computers as the control device.

The desire and need to generalize the theory led in the 1950s and early 1960s to the state-space approach. Originally developed in the field of classical and statistical mechanics some fifty to eighty years earlier, the state-space description was found to have great utility in the analysis and design of control systems. This new viewpoint acquired the label "modern control theory."

The number of theoretical problems that could be specified and studied now grew immensely; and the mathematical level of sophistication in many contributions has reached a stage that is causing a "gap" between the theoretician and practicing engineer. Theoreticians on the intermediate level with background in engineering are often needed to find the appropriate theory and adapt it for specific applications. Many of the algorithms resulting from this modern control are so involved that they can only be implemented by a digital computer, and thus the application of the theory depends upon the technology of digital computation.

## 1.2 Advanced Control Studies in Areas Other than Nuclear Engineering

In this section we briefly mention a few studies on advanced control in areas other than nuclear engineering.

Conventional fossil-fired power plants have some resemblance to Light Water Reactors (LWR), since they both produce steam under high pressure, which then drives a turbine and a generator. They are both, in many cases, designed to be load following, and thus it might be possible that a similar control strategy could be used.

McDonald and Kwatny (1973) used optimal linear regulator theory on a simulated 200 Mwe conventional plant, and compared modern and classical control approaches. They

controlled the simulated plant with a $14^{th}$ order linear model and developed algorithms that were intended to be implemented through conventional analog circuitry. On the basis of their simulations, they reported a projected improvement in plant transient performance compared to the performance with existing control. The importance of a good and meaningful linear model was, however, stressed.

The aircraft industry, military, and space programs are probably the groups that have contributed to and profited the most from the advances in automatic control. The knowledge in the use of digital computers obtained from these programs should be an important source of experience when implementing computers in nuclear power plants.

## 1.3 Applications of Computers in Nuclear Power Plants

Lunde (1973) conducted a review of current trends in applications of process control computers to nuclear power plants. In Norway, modern control theory has been implemented on the Halden Reactor Project. For detailed description of the plant, the control system approach and a collection of papers relating to this program, the reader is referred to Bjorlo et al. (1971).

Canadian nuclear power stations like Gentilly and Pickering are completely dependent upon computer control. The Pickering plant is reported to have plant unavailability

due to computer malfunctions of less than 0.5%. The control functions include total power and power distribution, pressure and turbine control, in addition to passive alarm and monitoring functions. The control theory used is relatively conventional.

In the U.S., on-line computers are employed on most nuclear power plants, but they are only assigned to passive tasks such as data acquisition, alarm reporting, post incident reviews, power distribution, and thermohydraulic calculations. Reactor vendors in the U.S. have apparently so far been reluctant to go much further into advanced applications with possible feedback control through the computer. The reasons seem to be licensing problems, fear of unreliability affecting plant performance, and doubtful economic advantage.

The study by Rasmussen et al. (1974) and the complexity of the Three Mile Island incident indicate that operator errors strongly contribute to risk probabilities for nuclear reactors. This is expected to stimulate an increased interest in automatic control of LWR's in the U.S.

## 1.4 Review and Objective of the Thesis

In this thesis a DARE P software package (Korn and Wait, 1978) will be used in developing a suboptimal controller for a non-linear simulation model of a Boiling Water Reactor Nuclear Power Plant (the Laguna Verde Nuclear Power

Plant in Mexico). The plant was originally modelled by the Instituto de Investigaciones Electricas (IIE) in Cuernavaca. A set of 48 nonlinear first-order differential equations and a large number of algebraic equations will be linearized about the equilibrium state using the subroutine EIGEN in the DARE P package. We will use partitioning to transform the linearized equations into a block triangular subsystem.

The concept of optimal control theory and a square performance index reflecting the desired plant behavior will be applied to the slow subsystem to develop a suboptimal feedback controller. The decision to use Linear-Quadratic (LQ) theory is mainly based upon the simplicity that can be achieved in the final performance. The LQ-approach has also, during the last decade, achieved considerable theoretical maturity, and has now received serious attention among practicing control engineers.

The literature in modern control theory contains a multitude of methods in addition to the LQ-approach. Most of these are more interesting from a theoretical point of view, although if they could be implemented without regard to limited computer capacity, they have the potential of giving better control performance. The theory of optimal control using Pontryagin's maximum principle, dynamic programming or nonlinear programming (Luenberger, 1969) enables one to solve a variety of nonlinear problems, and

virtually any analytical expression can be used as the performance index.

# CHAPTER 2

# LINEAR OPTIMAL CONTROL THEORY

## 2.1 Introduction

So many possibilities and permutations of compensation are possible in a complex system, that an efficient design may never be found unless a systematic approach is used. Wiener (1947) was one of the first to suggest that the search for an efficient system should be formulated as an optimization problem. The idea behind optimal control theory is that the designer devotes considerable attention to developing a mathematical model of the processes and then gathers together the competing control objectives into a cost function.

The choice of assumptions and level of mathematical sophistication has to be a trade-off between what control performance can be achieved and efforts (cost) needed to implement it. The Linear-Quadratic (LQ) Optimal Control Theory, as it will be presented in this chapter, results in surprisingly simple schemes when the right assumptions are made. However, several assumptions made about the system have to be liberally interpreted, but with a good insight into the theory and the processes, the LQ-theory offers the

designer a flexible and fairly simple approach to obtain a desired system behavior. The theory to be presented in this chapter is available in various forms in the literature, e.g., Kwakernaak and Sivan (1972). To keep the presentation simple, the equations to be presented will not be made more general than what is needed in this work.

## 2.2 Physical Problem, Its Analytical Description and Reference Trajectory

We are dealing with the interconnected entity of a physical plant connected to actuators and sensors, as indicated in Figure 2.1.

1. Actuators: The actuators are physical devices that translate command inputs, $\underline{u}(t)$, into actual plant inputs (reactivity, feedwater flow, etc.). This translation is not exact, as indicated by actuator uncertainties in Figure 2.1.
2. Plant: The plant is a physical device, in our case a BWR power plant, that responds to the actuator inputs, as well as other disturbances (power changes, condenser temperature changes, etc.). The condition of the plant is specified by the plant state variables. The description of the time evolution of the state vector, $\underline{x}(t)$, is assumed to be of the form

$$\dot{\underline{x}}(t) = f(\underline{x}, \underline{u}, t) \qquad (2.1)$$

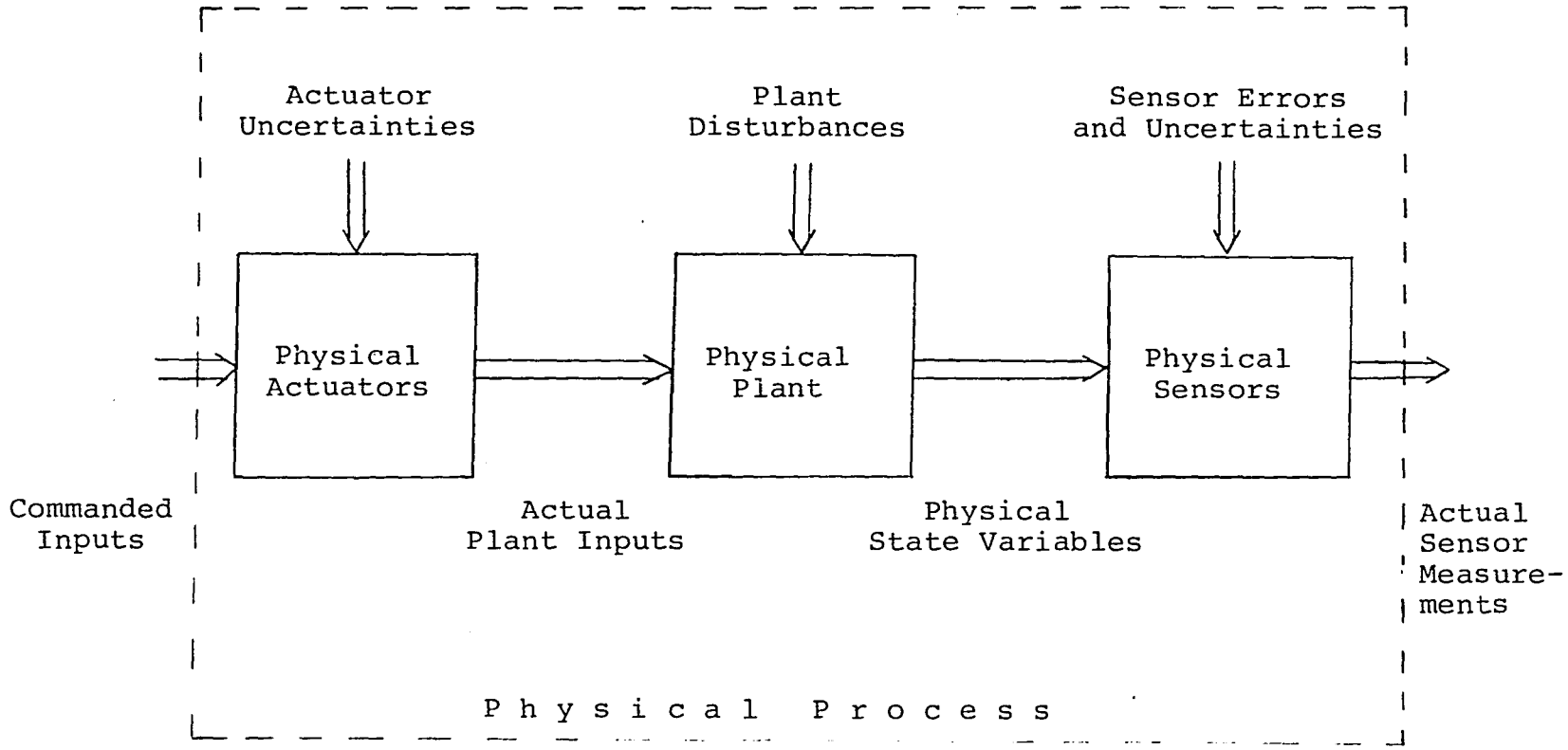



Figure 2.1. Schematic representation of physical processes.

3. Sensors: It is either impossible or not desirable, for engineering reasons, to measure all the state variables. We designate by the term physical sensors those devices that measure physical variables (state variables). However, the actual sensor measurements differ from the true values, because any sensor has a limited accuracy and many introduce errors by themselves. Denote the measurement variables as $\underline{y}(t)$, and assume their relation to the state vector is given by

$$\underline{y}(t) = h(\underline{x}) \tag{2.2}$$

The actuators and sensors may, in addition to uncertainties, be limited in the speed of response, i.e., have physical response characteristics of their own independent of those of the plant. If such is the case, one may describe this by adding more variables to the plant state equation above.

The objective of a control system, loosely stated, is to find a control sequence, $\underline{u}(t)$, such that the response of the plant is "satisfactory" for the considered application. The key quantities that enter into such a formulation are components of the whole state vector rather than only the measured variables. To synthesize this "state variable feedback," we need a physical device working

according to specified rules (algorithms), whose task will be to translate the actual sensor measurements into command inputs to the physical process. Since the device we eventually have in mind is a digital computer, we will call the collection of rules (algorithms) a "compensator." The algorithm to estimate the plant state from the measurements is called a "state estimator," and the algorithm to calculate the amount of feedback control is called a "controller."

When an analytic description of the processes is available, one may be able to specify a desired (reference) response for the plant in the form

$$\dot{\underline{x}}^*(t) = f(\underline{x}^*, \underline{u}^*) \tag{2.3}$$

One may arrive at such a reference response in at least two ways:

1. through deterministic optimal control theory, calculate $\underline{u}^*(t)$ as a function of $\underline{x}^*(t)$ and initial conditions, or
2. by engineering design, specify the desired control $\underline{u}^*(t)$ as a function of specific conditions.

We will, in our case, choose the reference control such that we obtain the desired steady state conditions in the case of no disturbances, i.e., we are looking for the solution of

$$\underline{f}(\underline{x}^*, \underline{u}^*) = \underline{o} \tag{2.4}$$

Generally Equation 2.4 is nonlinear with nonunique solution. The question that now arises is, what happens to the plant if we apply $\underline{u}^*$ as the control. The general answer will be that

$$\underline{f}(\underline{x}^*, \underline{u}^*) \neq 0 \tag{2.5}$$

for at least two reasons: (1) the process is not exactly modeled, and (2) we ignored $\underline{x}^*(t)$ in Equation 2.4 as compared to the exact equation (2.3).

If we agree that our design objective is to keep the actual plant state $\underline{x}(t)$ close to its desired value $\underline{x}^*(t)$ for all time of interest, then it is clear that the actual plant input must be different from the computed reference input $\underline{u}^*(t)$. One therefore has to use feedback control to take care of errors that primarily are associated with errors in modeling.

## 2.3 Linearized Model and Some Linear Control Theory Theorems

If we apply the reference control $\underline{u}^*$, the actual state is suspected to be quite close to the reference value. This suggests that we may linearize the model given by Equation 2.1 by considering small perturbations about $\underline{x}^*(t)$ and $\underline{u}^*(t)$. By retaining only first-order terms in a Taylor expansion, one obtains

$$\frac{d}{dt}(\underline{x} - \underline{x}^*) = \frac{\partial \underline{f}}{\partial \underline{x}}\Big|_o (\underline{x} - \underline{x}^*) + \frac{\partial \underline{f}}{\partial \underline{u}}\Big|_o (\underline{u} - \underline{u}^*)$$

where the partial derivations are evaluated at $\underline{x}^*$ and $\underline{u}^*$.

Define

$$\underline{\delta x} = \underline{x} - \underline{x}^*$$

$$\underline{\delta u} = \underline{u} - \underline{u}^*$$

We obtain

$$\underline{\dot{\delta x}} = A(t)\underline{\delta x} + B(t)\underline{\delta u} \tag{2.6}$$

Similarly, the measurement Equation 2.2 can be expanded to obtain

$$\underline{\delta y} = C(t)\underline{\delta x} \tag{2.6a}$$

where

$$A = \begin{vmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{vmatrix} \tag{2.7a}$$

$$B = \begin{vmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{vmatrix} \qquad (2.7b)$$

The matrices A, B and C above were designated as time variant; however, in this work these matrices are considered time invariant with their elements evaluated at some arbitrarily specified power level. The matrix A is also denoted the Jacobian of the nonlinear system.

The linear equations (2.6, 2.6a) are the basis for the feedback control that is to be described in the next sections. Before going into this theory, we will quote a few theorems (Kwakernaak and Sivan, 1972), that are well known in linear systems theory.

Theorem 2.1 Stability: A system described by the time invariant version of Equation 2.6 is asymptotically stable if and only if all the eigenvalues of A have negative real parts.

Theorem 2.2 Controllability: A system described by Equation 2.6, where A is a n x n matrix and B is a n x m matrix, is completely controllable if and only if:

$$\text{Rank } [B,\ AB,\ A^2B,\ \ldots,\ A^{n-1}B] = n \qquad (2.7)$$

We will, in short notation, say that the system described by Equation 2.6 is completely controllable if and only if (A,B) is a completely controllable pair.

Theorem 2.3 Observability: A system described by Equations 2.6 and 2.6a where A is a n x n matrix and C is a $\ell$ x n matrix, is completely observable if and only if:

$$\text{Rank } [C^T,\ A^TC^T,\ (A^T)^2C^T,\ \ldots,\ (A^T)^{n-1}C^T] = n \qquad (2.8)$$

Theorem 2.4: (A,B) is a completely controllable pair if and only if $(B^T,\ A^T)$ is a completely observable pair, i.e., the conditions are dual.

## 2.4 Optimal Feedback Control

A major development in control theory occurred during the late 1950s and 1960s, which was fostered by high-speed computers and an advanced technology space program. Now known under various names (quadratic synthesis, linear-quadratic Gaussian synthesis, modern optimal control theory, linear quadratic regulator theory), this development has been the subject of intensive research. Theoretically, it is known that optimal control has great potential since

it is inherently a multi-input/multi-output linear system control design method. Much of the demonstration of the versatility and the method, however, has been limited to simulation studies.

Some general characteristics of quadratic optimal control are:

1. Quadratic synthesis techniques have a different design objective from classical techniques. Instead of attempting to obtain a specific transient response (as do classical methods), LQ methods minimize the control energy required to keep the mean square error response of the system as small as possible. The design parameters are weightings or penalties on deviations of states and controls.
2. Quadratic controls can produce simple control systems if the design process is conducted with an integrated understanding of the physical system.

The optimal control system (Kalman, 1958) is designed to a performance index that defines the optimality condition. The LQ approach is based on a particular form of the performance index. The most common choice for the cost function or performance index is a quadratic form:

$$J = 1/2 \int_{t_o}^{t_f} [\underline{x}^T(t)Q\underline{x}(t) + \underline{u}^T(t)R\underline{u}(t)]dt \tag{2.9}$$

where Q is a positive semi-definite weighting matrix on state errors and R is the positive definite weighting matrix on the control effort. For the linear dynamic system described by Equation 2.6, the control law that minimizes the performance index J has the simple structure:

$$\underline{u}(t) = -K(t)\underline{x}(t) \tag{2.10}$$

where the optimal feedback gain matrix is given by

$$K(t) = R^{-1}B^{T}S(t) \tag{2.11}$$

and S(t) is the solution of the associated matrix differential Riccati equation (Laub, 1978)

$$\dot{S}(t) = SA + A^{T}S - SBR^{-1}B^{T}S + Q \tag{2.12}$$

The steady-state quadratic optimal regulator gain for continuous, linear, time-invariant system is obtained when $\dot{S}(t)$ is set to zero and $t_f \rightarrow \infty$ in Equation 2.12 and the solution substituted in Equation 2.11. Under these conditions, minimization of the cost function results in the constant feedback law, Equation 2.10.

The resulting closed-loop system

$$\dot{X}_{CL}(t) = [A - BK]\underline{x}(t)$$

is always stable; that is, the eigenvalues of the closed-loop system have negative real parts. The control law is optimal in the sense that it minimizes the control effort required to keep the mean-square deviation of the states from the reference value as small as possible.

## 2.5 Order Reduction Based on the L-Transformation

Mathematical problems of large-scale systems are usually too large and too complicated to handle. Analysis and synthesis problems concerned with such systems are, consequently, very difficult to solve. Also, in many cases the exact solution of the problem is not necessary. Instead, a good approximate solution is sufficient. For this reason, great efforts have been made to develop methods to obtain reduced order models that preserve the main properties of the higher order one. These properties are usually taken to be stability and steady-state response for some standard input function.

Several methods have been developed, based on different ideas, to reach this goal. Davison (1966) developed a modal reduction method which evoked a series of correspondence between him and Chidambara (1967). They ended up with two similar methods based on the dominant mode concept. These methods are all approximate in the sense that the normal modes calculated from a reduced model differ

from the corresponding normal modes of the original system. The method to be described here, based on similar work by O'Malley and Anderson (1979) produces subsystems which exactly preserve the modes of the original system.

Consider a dynamic system described by Equation 2.1, which is well represented by the linearized Equation 2.6 with the matrices A and B being constant. Let us assume that we can separate the eigenvalues (A) of A into two sets W and S of order $n_1$ and $n_2$, respectively (where $n_1 + n_2 = n$).

Equation 2.6 can be writen as:

$$\begin{vmatrix} \dot{\underline{x}}_1 \\ \dot{\underline{x}}_2 \end{vmatrix} = \begin{vmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{vmatrix} \begin{vmatrix} \underline{x}_1 \\ \underline{x}_2 \end{vmatrix} + \begin{vmatrix} B_1 \\ B_2 \end{vmatrix} \underline{u} \tag{2.13}$$

where $\underline{x}_1$ and $\underline{x}_2$ are $n_1$ and $n_2$ dimensional vectors and the $A_{ij}$ are submatrices, each having proper dimensions. The next step is to exhibit A in its Jordan Canonical Form:

$$A = MJQ = \begin{vmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{vmatrix} \begin{vmatrix} J_1 & 0 \\ 0 & J_2 \end{vmatrix} \begin{vmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{vmatrix} \tag{2.14}$$

where M consists of column eigenvectors, $Q = M^{-1}$ and $J_1$ and $J_2$ are the Jordan blocks. Before proceeding further, we

require that the eigenvalues of Jordan block $J_1$ be the fast ones, i.e., $(J_1) = W$. If this is not already the case, then a single permutation transformation will make it so.

A canonical transformation of variables can be employed to reduce Equation 2.13 to a block-triangular form. Specifically, define the transformation matrix T as

$$T = \begin{vmatrix} I_1 & 0 \\ L & I_2 \end{vmatrix} \tag{2.15}$$

where

$$L = -M_{21}M_{11}^{-1} . \tag{2.16}$$

L is a real $n_2$ x $n_1$ matrix and $I_1$, $I_2$ are identity matrices of order $n_1$ and $n_2$, respectively. Applying the above transformation as follows,

$$\begin{vmatrix} \underline{z}_1 \\ \underline{z}_2 \end{vmatrix} = T \begin{vmatrix} \underline{x}_1 \\ \underline{x}_2 \end{vmatrix} \tag{2.17}$$

Equation 2.13 becomes

$$\begin{vmatrix} \dot{\underline{z}}_1 \\ \dot{\underline{z}}_2 \end{vmatrix} = \begin{vmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{vmatrix} \begin{vmatrix} \underline{z}_1 \\ \underline{z}_2 \end{vmatrix} + \begin{vmatrix} B_1 \\ LB_1 + B_2 \end{vmatrix} \underline{u}$$

where

$$\tilde{A}_{11} = A_{11} - A_{12}L \tag{2.19}$$

$$\tilde{A}_{12} = A_{12} \tag{2.20}$$

and

$$\tilde{A}_{22} = A_{22} + LA_{12} \tag{2.21}$$

This last result is due to Anderson and O'Malley (1979, 1980) and may be thought of as a partial diagonalization of A. Also, it can be shown (Anderson and Hallauer, 1980) that L has to satisfy the nonsymmetric algebraic Riccati equation (NARE)

$$LA_{11} + A_{21} - LA_{12}L - A_{22}L = 0 \tag{2.22}$$

The transformed A-matrix, i.e.,

$$\tilde{A} = \begin{vmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{vmatrix} \tag{2.23}$$

has the desired property that the fast eigenvalues are contained in $\tilde{A}_{11}$ and the slow eigenvalues are contained in $\tilde{A}_{22}$ (the matrix $\tilde{A}_{12}$ represents the coupling between these two subsystems, Figure 2.2). The two subsystems are therefore decoupled. The new system has two features of practical importance. Firstly, observe that to obtain L, only $M_{11}$ and $M_{21}$ and not the full modal matrix is required. This is an advantage since relatively little computational effort is expended to calculate the eigenvectors. The other point to note is that the transformation matrix T has a particularly simple inverse, namely

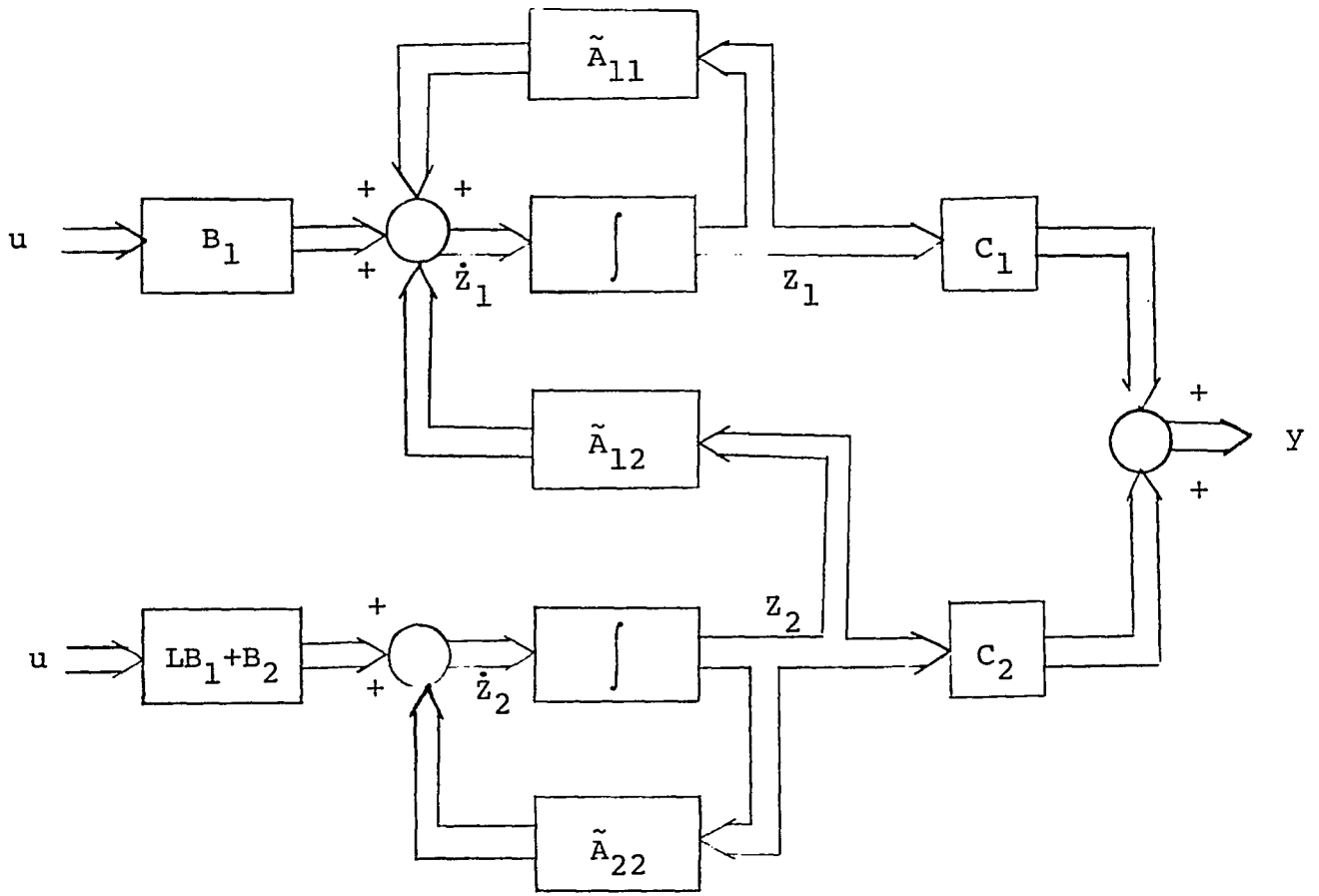



Figure 2.2. Block diagram of the transformed linear system.

$$T^{-1} = \begin{vmatrix} I_1 & 0 \\ -L & I_2 \end{vmatrix} \tag{2.24}$$

Thus no matrix inversions are required when transforming back to the original state variables.

Another issue of importance concerns the calculation of the matrix L (Hetrick, 1982). In the numerical calculation of eigenvalues and eigenvectors, machine capability limitations will result in inaccurate calculation of the matrices $M_{11}$ and $M_{21}$. This may yield inaccuracies in L as given by Equation 2.16. Instead, Equation 2.22 may be written as

$$L = (A_{22}L - A_{21})(A_{11} - A_{12}L)^{-1} \tag{2.25}$$

and the estimate of L may be improved by iteration using

$$L_{i+1} = (A_{22}L_i - A_{21})(A_{11} - A_{12}L_i)^{-1} \tag{2.26}$$

Experience indicates that this procedure converges very rapidly, even with a poor initial estimate for the matrix L.

## 2.6 Suboptimal Control

This method of system partitioning has several applications that could yield reduced computer cost without loss of accuracy.

1. The efficiency of explicit numerical integration methods may be increased by using small time steps for only a few of the variables.
2. An implicit method may be used for integrating a few variables, while many slower variables are integrated by an explicit method (Keepin, 1982).
3. The partitioning scheme may be used for order reduction, identifying the $n_1$ variables in a fast system that can safely be treated quasistatically (Hetrick and Palusinski, 1977).

Suboptimization can be considered to be applicable to such systems which have eigenvalues of nonpositive real parts in the fast subset. Consider the eigenvalues which are far from the origin. They either have large real part or imaginary part or both. The contribution of the mode corresponding to such an eigenvalue dies out faster. Indeed the modes corresponding to eigenvalues close to the origin roughly determines the type of the response which the whole system will have. Based on these considerations, the decision in practice is to partition the system such that the eigenvalues of smaller magnitude very close to the origin are placed in the slow subset for control purposes (Kleinman, 1968).

The concept of suboptimal control (if the subsystems are weakly coupled) is therefore to find the optimal gain

for the slow subset. When the closed loop is coupled with the fast subset, a general solution with the appropriate initial conditions can be found for the whole transformed system which can be transformed back to the original state variables with the transformation matrix.

Optimal systems may become unstable if subjected to structural perturbations. Intuitively, the higher the degree of cooperation among different parts of the subsystems, the higher the efficiency of the overall system. Increased cooperation, however, means increased interdependence among the subsystems, which in turn may jeopardize the functioning of the overall system when some of the subsystems cease to participate. Since for a proper operation of large-scale systems, it is essential that structural changes do not cause a breakdown, a trade-off between optimality and dynamic reliability should be established. Therefore, the control schemes for large systems should be designed to ensure dynamic reliability despite a possible deterioration of the optimal performance.

Siljak and Sundareshan (1976) outlined a multilevel control scheme which inherently incorporates the desired trade-off. It is assumed that a large system is decomposed into a number of subsystems which are optimized by local feedback controllers with respect to a local performance index when ignoring the interactions among the subsystems.

That is, each subsystem optimizes its performance as if it were decoupled from the rest of the system. This strategy is shown to result in a reliable system when the interconnections are suitably limited (Kokotovic' and Singh, 1971).

When the locally optimal subsystems are interconnected, the interconnections act as perturbations, causing a degradation of the system performance and thus a suboptimal performance of the overall system. For this reason, a suboptimality index is defined which measures the performance deterioration and represents the trade-off between optimality and reliability in large-scale dynamic systems.

Unless the subsystems are weakly coupled (Kokotovic', Perkins, Cruz, D'Ans, 1969; Kokotovic' and Singh, 1971), the local optimizations may be in conflict with each other, producing a poor or even unstable overall system.

# CHAPTER 3

# MODELING

## 3.1 System Description

The system being studied in this work is a General Electric Company boiling water reactor, the Laguna Verde Nuclear Power Plant in Mexico (Schwarzblat et al., 1982). Relevant technical and safety descriptions of the BWR plant can be found in the Preliminary Safety Analysis Report (PSAR) or Final Safety Analysis Report (FSAR) for the Laguna Verde Nuclear Power Plant in Mexico. For a more qualitative description, reference BWR/6 (1973) is a good source, For an extensive analytical description of BWR dynamics for current plant designs, the reader is referred to Linford (1973).

In this section, the system will be very briefly described. Table 3.1 lists the main design data at 100% power level, i.e., rated conditions, as assumed in this work. Figure 3.1 shows a very simplified diagram of a direct cycle BWR. Feedwater comes from the condenser through the feedwater pumps. The feedwater is, in some plants, heated by extracting steam from the turbine. Feedwater, together with core-recirculated water, goes down the downcomer, where about one-third is forced through the

Table 3.1. Assumed designed data for BWR plant (Laguna Verde, 1979).

| | |
|---|---|
| Vessel pressure | 1032.25 psia |
| Core average void fraction | 38% |
| Core exit quality | 13.5% |
| Core inlet enthalpy | 527.1 BTU/lbm |
| Core inlet subcooling | 21.5 BTU/lbm |
| Core mass flow | $1.708 \times 10^4$ lbm/sec |
| Mass flow through each recirculation loop | $2.9305 \times 10^3$ lbm/sec |
| Steam flow | $2.3064 \times 10^3$ lbm/sec |
| Active core height | 12.5 ft |
| Boiling length in core | 8.4 ft |
| Average fuel temperature | 1075 $^oF$ |
| Doppler coefficient, $[\frac{\partial \rho}{\partial T_f}]$ | $-9.8 \times 10^{-4}$ \$/$^oF$ |
| Void-reactivity coefficient, $[\frac{\partial \rho}{\partial \alpha}]$ | -26.6 \$ |
| Total power generated in core | 1931 MW(t) |
| Pressure in front of turbine valve | 973.8 psia |
| Pressure at inlet to HP-turbine | 948.27 psia |

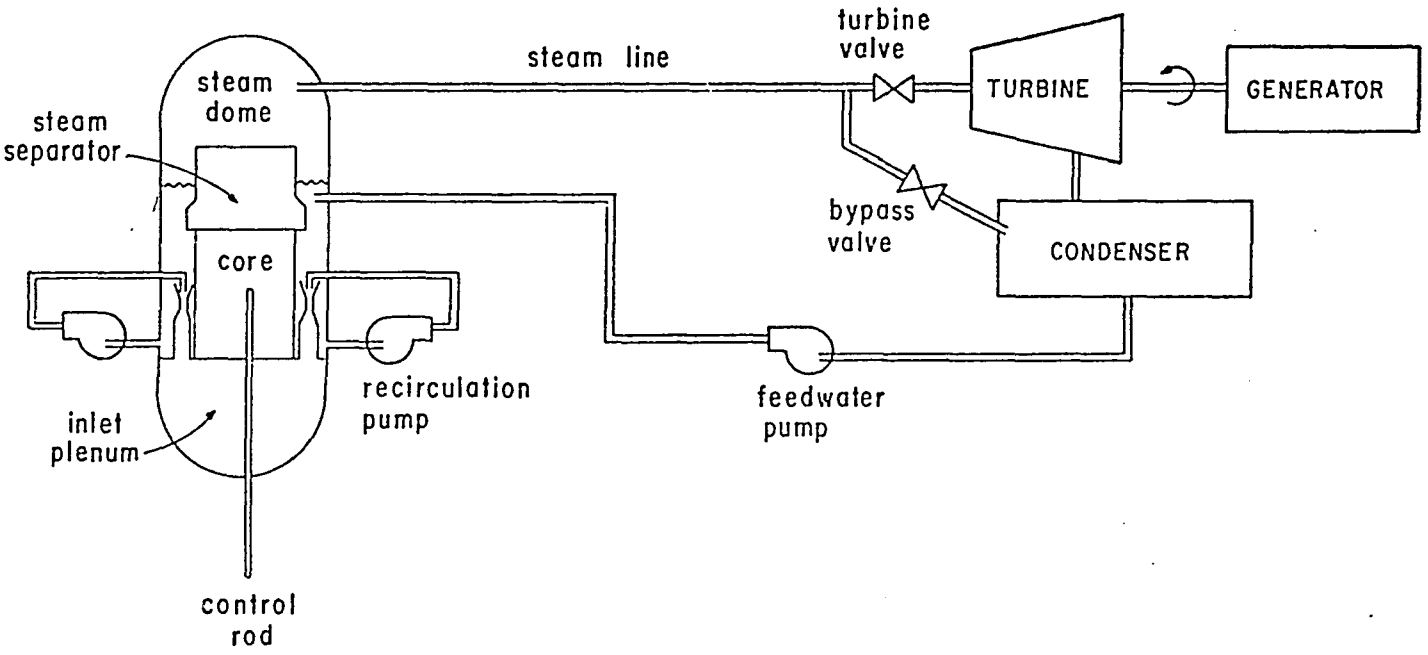



Figure 3.1. Simplified direct cycle BWR system (BWR/6, 1973).

recirculation pumps providing driving flow to the jet pumps. The jet pumps thus enhance circulation through the core where liquid is heated and partially evaporated. Two-phase mixture exits the core at about 10 to 15% quality. The steam goes further through the steam separators and steam dryers before it reaches the vessel steam dome.

Steam enters the steam line and goes through the turbine valve, where it enters the high pressure (HP) turbine and finally enters the low-pressure (LP) turbine. Moisture separators, reheaters and steam extraction as part of the turbine are found on most designs, but are not explicitly shown on the figure because they do not play any part in the model.

Generally, BWR plants have a strong negative void-reactivity coefficient; i.e., when the average void fraction (volume of steam/volume of the core) increases, the effectiveness of the coolant to the thermalize neutrons decreases, less neutrons undergo fission and thus less heat is produced by the governing fission process. This phenomenon overwhelmingly determines the transient behavior of the reactor vessel.

If the core mass flow is increased, it will soon cause a decrease of the core void fraction, which increases the neutron density. Manipulation of the power input to the recirculation pumps is the principal way of changing the

generated power in the nuclear system, particularly during load following. Manipulating the control rods can also change the core power output, but these manoeuvers are kept to a minimum.

If the vessel pressure is changed, e.g., decreased by opening the turbine valve, it will cause an increase in the void fraction, which then decreases the core power output. Similarly, if the pressure increases it will contribute to an increase in core power output. This behavior is opposite to what could be desired, and constitutes a potential for instability. A BWR for this reason is essentially operated at constant vessel pressure, i.e., one important objective of the control system is to keep the pressure constant, while utilizing small changes of the pressure to improve load-following response. If the load demand increases, a temporary lowering of vessel pressure will result in an extra amount of steam being generated by flashing of the liquid in the vessel. If the load decreases, an intentional increase in vessel pressure will condense some of the superfluous steam and thus store the energy temporarily in the vessel rather than dump it all through the bypass valve into the condenser.

## 3.2 Modeling Approach

The LQ-optimal control theory, as described in Chapter 2, demands a linear model of the processes. A good

model does not necessarily mean a high-order model. The computational requirements increase rapidly with the size of the state vector, and it is, generally speaking, more difficult to assign values to the weighting matrices for a high-order model. A priori requirements on the model are that it should be of as low an order as possible while still accounting for the main physical events taking place.

Developing a good linear model for a complex plant like a BWR is usually a major effort. A common way of developing linear models is to first develop a set of nonlinear differential and algebraic equations for the system and linearize these equations, as indicated in Section 2.3. This linearized model often contains more state variables than desired, and some criterion is then needed to reduce the order of the model. Davison (1966) has described a computationally attractive approach to linear model reduction, that essentially eliminates the fast modes of the model. The use of such an approach is questionable because it would most likely suggest the omission of important variables or even omitting the dynamics of complete subsystems.

Another approach is to first simplify the extensive nonlinear model and then linearize this simplified set of equations. One difficulty with this approach is the problem of making the approximations, and certain aspects of the

processes may have to be ignored altogether in the simplified model.

In this work two approaches were tried. The first was to use the sensitivity function (Haberman, 1977) approach, with a single-time-constant response to couple together the desired state variables. Detailed effects that are not of prime interest may thus be lumped into the time constant. This approach, however, did not work. The response of the linearized model did not represent its non-linear version. The second method, which was successively implemented, was the direct conservation approach as indicated in Section 2.3.

## 3.3 Reactor Kinetics

Most of the neutrons produced in nuclear fissions are of high energies and are emitted instantaneously (prompt neutrons). However, a fraction of the fission neutrons are emitted, as a result of the formation and decays of radionuclei, after a time delay which ranges from a fraction of a second up to minutes. These delayed neutrons are represented by six groups. The behavior of the core power is not affected by the time delay during steady power operation. In transients, the power is affected by the delayed neutrons. The neutron distribution (flux) in a reactor core is space-, energy-, and time-dependent. The energy dependence of the neutron flux is eliminated by

dividing neutrons into energy groups and expressing the neutron flux of each energy group by a differential equation, called the neutron diffusion equations (Henry, 1975), yielding a set of equations which is very difficult to solve. For simplicity, the neutron flux is expressed by the one group diffusion equation to which further approximations are applied by assuming that the flux is separable in space and time. This leads to the derivation of a simple model, called the point kinetic model, describing the time-dependence of the neutron power (Duderstadt and Hamilton, 1976). The basic assumptions for the point kinetic model which will be used in this work, therefore, are

1. The entire neutron population responds as a single energy group.
2. The neutron flux spatial and temporal variations are separable for all considered transients with system response due to only the fundamental flux mode.
3. For the transient of interest, the reactivity feedbacks can be calculated for an average channel to represent total core response.

The neutron balance equation can be expressed by the point-kinetic model (Hetrick, 1971)

$$\frac{dn}{dt} = \frac{\rho-\beta}{\ell} n + \sum_{i=1}^{6} \lambda_i C_i \qquad (3.1)$$

$$\frac{dC_i}{dt} = \frac{\beta_i}{\ell} n - \lambda_i C_i \qquad (3.2)$$

where

$n(t)$ = neutron density at time t

$\beta$ = effective delayed neutron fraction

$$\beta = \sum_{i=1}^{6} \beta_i$$

$c_i(t)$ = $i^{th}$ delayed neutron precursor density

$\lambda_i$ = $i^{th}$ delayed neutron precursor decay constant

$\ell$ = mean neutron generation time

$\rho$ = total reactivity

$= \rho_{cR} + \rho_f + \rho_v$

$\rho_{cR}$ = control rod reactivity

$\rho_f$ = reactivity due to the fuel temperature changes

$\rho_f = \alpha_{T,f}(T_F - T_{Fo})$

$\alpha_{T,f}$ = fuel temperature coefficient of reactivity

$T_f$ = fuel temperature at time t

$T_{Fo}$ = fuel temperature at steady state

$\rho_v$ = void reactivity feedback

$\rho_v = 0.455\ (\alpha_c - \alpha_{co}) + 20.232(\alpha_c - \alpha_{co})^2$

$\alpha_c$ = core void fraction at time t

$\alpha_{co}$ = core void fraction at steady state

Typical values used for the delay group fractions $\beta_i$, and decay constants $\lambda_i$ are given in Table 1.1 (Hetrick, 1971).

Using two groups of delayed neutron precursors, and defining

$$T_o = \ell/\beta \tag{3.3}$$

$$\rho\$ = \rho/\beta \tag{3.4}$$

$$y_i = T_o C_i \tag{3.5}$$

results in the following, when substituted into Equations 3.1 and 3.2:

$$\frac{dn}{dt} = \frac{\rho\$-1}{T_o} n + \frac{\lambda_1 y_1}{T_o} + \frac{\lambda_2 y_2}{T_o} \tag{3.6}$$

$$\frac{dy_1}{dt} = \frac{\beta_1}{\beta} n - \lambda_1 y_1 \tag{3.7}$$

$$\frac{dy_2}{dt} = \frac{\beta_2}{\beta} n - \lambda_2 y_2 \tag{3.8}$$

## 3.4 Fuel Heat Balance

The fuel surface temperature gives rise to a convective heat transfer rate to the moderator given by

$$Q_f = A_f h_{fm}(T_{f3} - T_{sat}) \tag{3.9}$$

where

$T_{sat}$ = coolant saturation temperature at vessel pressure

$T_{fi}$ = fuel temperature in fuel node $\neq$ i (i = 3 is the surface node (Linford))

$A_f$ = total amount of heat transfer area

$h_{fm}$ = $h_{fm}(T_3)$
= heat transfer coefficient for fuel/moderator

The coolant heat transfer coefficient ($h_{fm}$) is a function of the thermodynamic properties of the coolant and its Reynolds number as expressed by the following Dittus-Boelter correlation for a single phase flow (Rust, 1979)

$$Nu = 0.023(Re)^{0.8}(Pr)^{0.4} \tag{3.10}$$

where

Pr = fluid Prandtl number

Nu = Nusselt number = $\frac{h_{fm}D_e}{K}$

Re = coolant Reynolds number = $\rho v De/\mu$

De = flow channel equivalent diameter

v = fluid velocity

$\rho$ = flow density

u = fluid dynamic viscosity

For detailed calculation of the temperature difference between the clad surface and the coolant bulk temperature in the two-phase flow region, appropriate correlation should be used to predict the boiling heat transfer coefficient. However, this model assumes wet steam at low quality and, therefore, approximates the temperature difference by Jens and Lottes correlation (El-Wakil, 1971).

$$\Delta T = T_c - T_{sat}$$

$$= \frac{60(q''/10^6)^{1/4}}{e^{P/900}}$$

where

$T_c$ = clad outer surface temperature, $^oF$

$T_{sat}$ = saturation temperature at system pressure, $^oF$

p = coolant pressure, psi

q" = heat flux, Btu/hr-ft$^2$

In addition to the heat flux leaving the fuel rod surface, there is prompt heating of the moderator due to neutron slowing down and attenuation of radiation energy. The prompt heating accounts for approximately 4% of reactor power. The total rate of heat transfer is therefore

$$Q_t = Q_f + 0.04 \frac{Q_{tR} n(t)}{n_R(t)} \qquad (3.11)$$

where

$Q_{tR}$ = rated value of total heat generation

$n_R(t)$ = rated neutron flux

Consider the power distribution as known and remaining constant. If we know the amount $m_{sub}$ of the subcooled water in the core (boiling boundary), one may find the fraction of heat going to the subcooled and saturated part of the core, respectively. Then for a uniform heat addition (El - Wakil, 1971)

$$Q_{sub} = f_q Q_t \tag{3.12}$$

$$Q_{sat} = (1 - f_q) Q_t \tag{3.13}$$

where

$f_q = \frac{H_o}{H}$

$H_o$ = nonboiling height

$H$ = active length of the coolant channel

Generally, $f_q$ is a function of $m_{sub}$ and it can be determined by a curve fitted function of the $m_{sub}$. The fuel is divided into 3 nodes, with a heat balance of the form

$$\rho_f C_f V_{fi} \dot{T}_{fi} = (\text{heat flow in})_i - (\text{heat flow out})_i + (\text{heat generated})_i \tag{3.14}$$

where

$V_{fi}$ = volume of fuel node #i

$C_f$ = specific heat of the fuel

$\rho_f$ = density of fuel

Assuming a flat power distribution in the fuel pins and a cylindrical fuel geometry (Linford, 1973), the fuel temperature for the nodes can then be written in the form

$$\dot{T}_{f1} = \frac{1}{\rho_f C_f V_{f1}} \left[a_4 k_{f1}(T_{f2} - T_{f1}) + \frac{Q_{fR}}{n_R} n\right] \tag{3.15}$$

$$\dot{T}_{f2} = \frac{1}{\rho_f C_f V_{f2}} \left[a_4 k_{f1}(T_{f1} - T_{f2}) - a_5 k_{f2}(T_{f2} - T_{f3}) + \frac{Q_{fR}}{n_R} n\right] \tag{3.16}$$

$$\dot{T}_{f3} = \frac{1}{\rho_f C_f V_{f3}} \left[a_5 k_{f2}(T_{f2} - T_{f3}) - Q_f + \frac{Q_{fR}}{n_R} n\right] \tag{3.17}$$

where $a_4$ and $a_5$ are functions of the node radii, and $k_{f1}$ and $k_{f2}$ are average values for the fuel conductivity as a function of fuel temperatures. More details on the coefficients $a_4$, $a_5$, $k_f$, and $k_{f2}$ are provided in Appendix B. The product $\rho_f C_f$ is treated as constant. The average fuel temperature is then obtained by power and volume weighting on $T_1$, $T_2$, and $T_3$. Thus

$$\bar{T}_f = \frac{Q_1 V_{f1} T_{f1} + Q_2 V_{f2} T_{f2} + Q_3 V_{f3} T_{f3}}{Q_1 V_{f1} + Q_2 V_{f2} + Q_3 V_{f3}} \tag{3.18}$$

which simplifies if $Q_i$ is constant to

$$\bar{T}_f = \frac{T_{f1}V_{f1} + T_{f2}V_{f2} + T_{f3}V_{f3}}{V_{f1} + V_{f2} + V_{f3}} \tag{3.19}$$

## 3.5 Thermodynamic Model

To describe the thermodynamic model for the saturated regions of the core, reactor vessel and steam lines, a lumped parameter approach (Lewis, 1978) has been used.

The core is divided into two hypothetical sections, one part containing only subcooled liquid, and another part containing saturated liquid and steam. It is assumed that the saturated liquid and vapor in the core and plenum remain in thermodynamic equilibrium for any transient studied.

### 3.5.1 Core Exit Steam Quality

The average core exit steam quality will be defined by

$$X_e = \frac{\text{Core exit steam flow}}{\text{Core exit flow rate of mixture}} \tag{3.20}$$

The average exit quality at steady state is given by

$$X_e = \frac{Q_{sat}}{h_{fg} W_c} \tag{3.21}$$

where

$Q_{sat}$ = rate of heat transfer to saturated part of core

$h_{fg} = h_g - h_f$

= vaporization enthalpy

$w_c$ = core mass flow rate

A change in pressure will cause some water to flash or some steam to condense, depending on pressure decrease or increase, respectively, thus giving additional effect to $x_e$. This effect will occur during load following, and it is not accounted for in Equation 3.21. Below is given a derivation of $x_e$ where effort is made to account for the flashing effect.

Consider an energy and mass balance for the total volume of the core, Figure 3.2, which has a saturated part of volume $V_c$, and a subcooled part of mass $m_{sub}$:

energy balance:

$$Q_{sat} + Q_{sub} + w_c h_c - w_{fe} h_f - w_{ge} h_g = \frac{d}{dt}(m_f u_f + m_g u_g) \tag{3.22}$$

mass balance:

$$w_c - w_{fe} - w_{ge} = m_f + m_g + m_{sub} \tag{3.23}$$

saturated volume:

$$\frac{d}{dt}(m_f v_f) + \frac{d}{dt}(m_g v_g) = \dot{V}_c \tag{3.24}$$

where

$m_f$ = mass of saturated liquid in the core

$m_g$ = mass of steam in the core

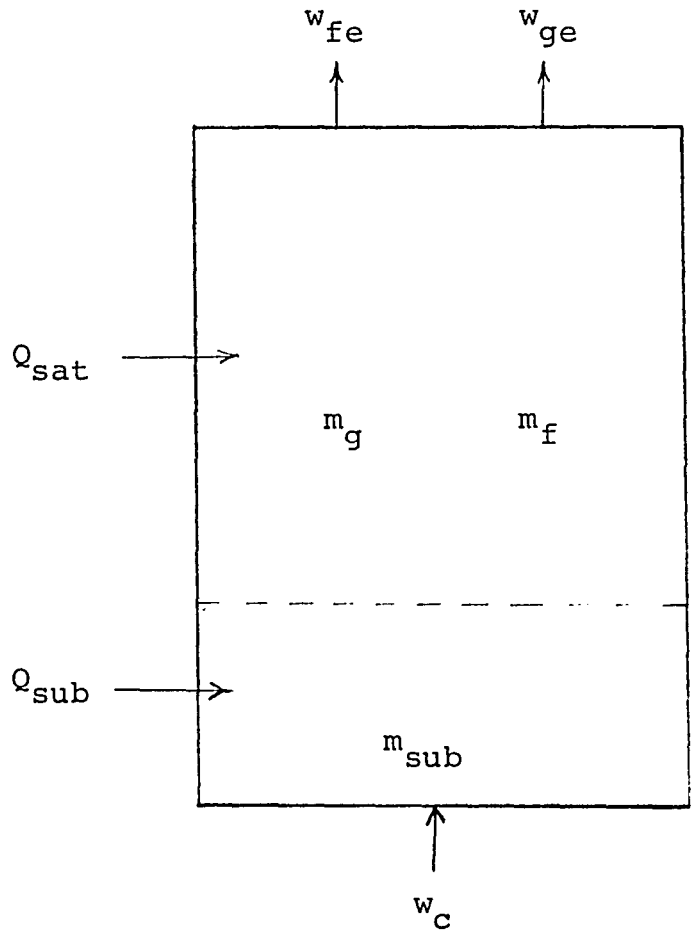



Figure 3.2. Thermodynamics model schematic.

$v_f$ = specific volume of saturated liquid

$v_g$ = specific volume of saturated steam

$u_f$ = specific internal energy of saturated liquid

$u_g$ = specific internal energy of saturated steam

$m_{sub}$ = mass of subcooled liquid in the core

$Q_{sub}$ = rate of heat transfer to subcooled part of core

$w_{fe}$ = liquid mass flow leaving the core

$w_{ge}$ = steam mass flow leaving the core

$V_c$ = volume of saturated part of core

The time derivative of the saturated volume is related to subcooled volume by differentiating the total core volume, which is constant.

$$V_{Tc} = V_c + m_{sub}v_{sub} \tag{3.25}$$

which gives

$$\dot{V}_c \simeq -\dot{m}_{sub}v_{sub} \tag{3.26}$$

where

$v_{sub}$ = specific volume of the subcooled liquid.

Introduce the approximation

$$\dot{m}_{sub}(v_f - v_{sub}) \simeq 0 \tag{3.27}$$

Notice

$$Q_{sub} = w_c(h_f - h_c) \tag{3.28}$$

i.e.,

$$Q_{sub} + w_c h_c = w_c h_f \tag{3.29}$$

If now the energy equation is expressed in terms of enthalpy, i.e.,

$$u_f = h_f - \frac{pv_f}{J} \tag{3.30}$$

$$u_g = h_g - \frac{pv_g}{J} \tag{3.31}$$

where

$h_f$ = enthalpy of saturated liquid

$h_g$ = enthalpy of steam

P = pressure

J = mechanical equivalent of heat

and when these are substituted into the energy equation (3.22), we get

$$Q_{sat} + w_c h_f - w_{fe} h_f - w_{ge} h_g = \frac{d}{dt}[m_f(h_f - \frac{pv_f}{J}) + m_g(h_g - \frac{pv_g}{J}] \tag{3.40}$$

By eliminating $\dot{m}_f$ from Equations 3.23 and 3.24 and rearranging, the following is obtained

$$\dot{m}_g = - \frac{v_f(w_c - w_{fe} - w_{ge}) + m_f\dot{v}_f + m_g\dot{v}_g}{v_g - v_f} \tag{3.41}$$

If $m_f$ from Equation 3.23 and $m_g$ from Equation 3.41 are substituted in Equation 3.40, the result is:

$$\frac{v_{fg}}{h_{fg}}(Q_{sat} - a_2\dot{P}) = v_{fg}w_{ge} - v_f(w_c - w_{fe} - w_{ge}) \tag{3.42}$$

Using a quasi-static approximation to Equation 3.23, we can write

$$w_c - w_{fe} - w_{ge} \simeq 0 \tag{3.43}$$

and therefore

$$w_c \simeq w_{ge} + w_{fe} \tag{3.44}$$

which, when substituted into Equation 3.42, yields

$$w_{ge} = \frac{Q_{sat} - a_2\dot{P}}{h_{fg}} \tag{3.45}$$

where

$$a_2 = m_f[\frac{dh_f}{dp}] + m_g[\frac{dh_g}{dp}] - \frac{V_c}{J} - \frac{h_{fg}}{v_{fg}}\{m_f[\frac{dv_f}{dp}] + m_g[\frac{dv_g}{dp}]\}$$

We can then obtain the core exit quality as

$$x_e = \frac{w_{ge}}{w_{ge} + w_{fe}} \simeq \frac{w_{ge}}{w_c} = \frac{Q_{sat} - a_2\dot{P}}{h_{fg}W_c} \tag{3.46}$$

### 3.5.2 Boiling Boundary

Determination of the boiling boundary in a heated channel is taken as synonymous with calculating the amount

of subcooled water in the channel. From a mass and energy balance on the subcooled mass in the core, the time derivative of the subcooled mass in the core is given by

$$\dot{m}_{sub} = \frac{w_c h_{sub} - Q_{sub} + m_{sub}(\dot{\bar{h}}_s - \frac{v_{sub}\dot{P}}{J})}{h_f - \bar{h}_{sub}} \tag{3.47}$$

The increase of liquid enthalpy from core inlet enthalpy ($h_c$), to saturated ($h_f$), at the boiling boundary is assumed to follow the axial pattern of the integral of power from the inlet to the boiling boundary. Since the total enthalpy inrease ($h_f - h_c$) is simply the inlet subcooling ($h_{sub}$), the average increase (as a function of $h_{sub}$) can be determined for any boiling boundary by averaging the enthalpy distribution up to the boundary. Thus, the average enthalpy (Rust, 1979) can be expressed as

$$\bar{h}_{sub} = F_s h_{sub} + h_c \tag{3.48}$$

where

$$F_s = \frac{1}{\frac{b}{L} Q_{sub}} \int_o^{b/L} Q''_T P_h dz$$

$P_h$ = channel heated perimeter

$Q''_T$ = wall heat flux

The normalized nonboiling height in the averege channel is obtained from

$$\frac{b}{L} = \frac{m_{sub} v_{sub}}{V_{cT}} \tag{3.49}$$

If the rate of change of $F_s$ is neglected in Equation 3.48, then the time derivative of $h_{sub}$ can be written as

$$\dot{h}_{sub} = F_s \frac{dh_f}{dp} \dot{P} + \dot{h}_c(1 - F_s) \qquad (3.50)$$

3.5.3 Core Average Void Fraction

There are mainly two types of phenomena that are important in describing the transient behavior of the core void fraction. First there is a "sweep" type behavior of the steam bubbles. This sweeping of voids through the core is due to the transport of energy or flow variations through the channel by the core flow. Hence, the variation of inlet subcooling, core flow, or heat flux affects not only the local void fraction but also the downstream void fraction along the channel until the coolant is discharged into the exit plenum. The second important phenomenon in the core void variation is the instantaneous effect of system pressure.

One technique for handling the sweep-type effects on core void fraction used in this model was developed by Akcasu (1960). By his approach, the core void $\alpha_c$ is derived as follows:

Influence from quality and subcooling:

$$\alpha_l = \alpha_l(x_e, h_{sub}) \qquad (3.51)$$

Influence from core flow (Linford, 1973):

$$\alpha_2 = \alpha_2(w_c, \alpha_1) \tag{3.52}$$

Influence from pressure:

$$\alpha_3 = \alpha_2 \frac{v_g}{v_{gR}} \tag{3.53}$$

$v_{gR}$ is the specific volume of steam at the design reference pressure.

By assuming a second-order sweep model with Laplace transform:

$$G(s) = \frac{\alpha_{3(s)}}{\alpha_{c(s)}} = (1 + 2\zeta\tau_{sc}s + \tau_{sc}^2 s^2) \tag{3.54}$$

the dynamic transients of the average void fraction can be incorporated through the second-order void sweep model in the form

$$\ddot{\alpha}_c = \frac{1}{\tau_{sc}^2}(\alpha_3 - \alpha_c - 2\zeta\tau_{sc}\dot{\alpha}_c) \tag{3.55}$$

where

$\zeta$ = damping coefficient

$\tau_{sc} = t_b\bar{\tau}(x_e)$

$\bar{\tau}(x_e)$ = curve fitted function of core exit quality

$$t_b = \frac{\text{boiling length}}{\text{inlet velocity}} = \frac{(1-m_{sub}v_{sub}/V_{cT})h_{core}}{w_c v_{sub}/A_{core}}$$

$h_{core}$ = active core length

$A_{core}$ = active core flow area

### 3.5.4 Saturated Mass in the Vessel

The saturated mass and its associated volume in the different parts of the vessel are given as follows:

Volume of saturated mass in core:

$$V_c = V_{cT} - m_{sub} v_{sub} \tag{3.56}$$

where

$V_{cT}$ = total active core volume

Mass of steam in core:

$$m_{gc} = \frac{\alpha_c V_{cT}}{v_g} \tag{3.57}$$

Saturated water in core:

$$m_{fc} = \frac{V_c - \alpha_c V_{cT}}{v_f} \tag{3.58}$$

Mass of steam in exit plenum:

$$m_{gp} = \frac{\alpha_p V_p}{v_g} \tag{3.59}$$

where

$\alpha_p$ = exit plenum average void fraction

and

$V_p$ = volume of reactor core outlet plenum

Saturated water in exit plenum:

$$m_{fp} = \frac{(1-\alpha_p) V_p}{v_f} \tag{3.60}$$

Mass of steam in vessel dome:

$$m_{gd} = \frac{V_d - m_{fd}v_f}{v_g} \tag{3.61}$$

where

$V_d$ = volume of vessel steam dome available for saturated steam and water

and

$m_{fd}$ = amount of saturated water in volume $V_d$

Total amount of saturated mass in vessel:

$$m_g = m_{gd} + m_{gc} + m_{gp} \tag{3.62}$$

$$m_f = m_{fd} + m_{fc} + m_{fp} \tag{3.63}$$

Total saturated volume in the pressure vessel:

$$V_V = V_c + V_p + V_d \tag{3.64}$$

From a mass balance on the core (see Appendix B), one gets the core exit mass as:

$$w_{ce} = w_c + \frac{v_{fg}}{v_f v_g}\dot{\alpha}_c V_{cT} - (1 - \frac{v_{sub}}{v_f})\dot{m}_{sub} + [m_f[\frac{dv_f}{dp}]\frac{1}{v_f} + m_g[\frac{dv_g}{dp}]\frac{1}{v_g}]\dot{P} \tag{3.65}$$

The core exit plenum void fraction is obtained from

$$\dot{\alpha}_p = \frac{1}{\tau_{ps}}(\alpha_{ec} - \alpha_p) \tag{3.66}$$

where

$$\tau_{ps} = \frac{m_{gp} + m_{fp}}{w_{ce}} \tag{3.67}$$

and from El Wakil (1971)

$$\alpha_{ec} = \frac{1}{1 + \frac{v_f}{v_g}(1 - x_e)\frac{1}{x_e}} \tag{3.68}$$

The mass flow through the core exit plenum is obtained from a mass balance (Linford, 1973)

$$w_{pe} = w_{ce} + \frac{v_{fg}}{v_f v_g}\dot{\alpha}_p V_p + \{\frac{\alpha_p}{v_g^2}[\frac{dv_g}{dp}] + \frac{1-\alpha_p}{v_f^2}[\frac{dv_f}{dp}]\}\dot{V}_p \tag{3.69}$$

### 3.5.5 Downcomer and Core Inlet Enthalpy

Some (approximately one-third) of the water going down the downcomer enters the jet pump loop, and this introduces a transportation delay and thermal mixing effect for the enthalpy perturbations in core inlet enthalpy following changes in the downcomer enthalpy $h_{dw}$. This effect is modeled by the following equations:

$$w_{dw} = w_c - w_f \tag{3.70}$$

Enthalpy of water entering the downcomer

$$h_{dw} = \frac{w_{dw}}{w_c} h_f + \frac{w_F}{w_c} h_F \tag{3.71}$$

Consider the enthalpy at the jet pump suction, $h_{suc}$, to be a pure delay of the downcomer enthalpy, such that in Laplace domain

$$h_{suc} = h_{dw} e^{-s\tau_{dw}} \tag{3.72}$$

where

$\tau_{dw}$ = transport delay in downcomer

$$= \frac{V_{dw}}{w_c v_{sub}}$$

$v_{sub}$ = specific volume of the subcooled water

$V_{dw}$ = volume of vessel downcomer

Transformation of Equation 3.72 to an ordinary differential equation by use of the approximation

$$e^{-s\tau} = \frac{e^{-1/2s\tau}}{e^{1/2s\tau}} \simeq \frac{1 - 1/2s\tau}{1 + 1/2s\tau}$$

one gets in the time domain

$$\dot{h}_{dp} = \frac{2}{\tau_{dw}}(h_{dw} - h_{suc}) \tag{3.73}$$

where

$h_{dp}$ = enthalpy at the lower plenum

$= h_{suc} + h_{dw}$

A fraction, $\frac{2w_d}{w_c}$, of the core flow is taken from the vessel through the two recirculation loops. The enthalpy at the jet nozzle, $h_{noz}$, is in the Laplace domain assumed to be given by

$$h_{noz} = h_{suc} e^{-s\tau_{pl}} \tag{3.74}$$

where

$\tau_{pl}$ = transport delay in recirculation loop
$= \frac{V_{pl}}{w_d v_{sub}}$

$w_d$ = mass flow through the recirculation loop

and

$V_{pl}$ = volume of one recirculation loop

As the approximation for the time derivative of the recirculation flow enthalpy is obtained as:

$$\dot{h}_{pl} = \frac{2}{\tau_{pl}}(h_{suc} - h_{noz}) \tag{3.75}$$

where

$$h_{pl} = h_{noz} + h_{suc}$$

Thus one obtains the jet pump diffuser enthalpy from

$$h_{diff} = (1 - \frac{2w_d}{w_c})h_{suc} + \frac{2w_d}{w_c} h_{noz} \tag{3.76}$$

The core inlet enthalpy is now obtained from

$$\dot{h}_c = \frac{1}{\tau_{ip}}(h_{diff} - h_c) \tag{3.77}$$

where

$$\tau_{ip} = \frac{V_{ip}}{w_c v_{sub}}$$

and

$V_{ip}$ = volume of vessel inlet plenum.

### 3.5.6 Vessel Pressure Rate of Change

Changes in vessel pressure are considered to occur uniformly throughout the vessel. A change in vessel pressure, P, is caused by an imbalance between steam produced, steam flow leaving the vessel through the steam line, and mass flow entering the vessel as feedwater. As already mentioned, steam is produced by two mechanisms:

1. by adding heat to the moderator, or
2. by flashing caused by pressure changes.

By following a derivation similar to that made for the core exit steam quality in Section 3.5.1, one gets the vessel pressure rate

$$\dot{P} = \frac{1}{a_3}[Q_{sat} + \frac{h_{fg}}{v_{fg}}(w_F v_F - w_l v_g)] \tag{3.78}$$

where

$$a_3 = m_f[\frac{dh_f}{dp}] + m_g[\frac{dh_g}{dp}] - \frac{V_v}{J} - \frac{h_{fg}}{v_{fg}}\{m_f[\frac{dv_f}{dp}] + m_g[\frac{dv_g}{dp}]\}$$

$w_l$ = steam flow leaving the vessel

$w_F$ = feedwater flow entering the vessel

### 3.5.7 Feedwater and Water Level

The feedwater flow entering the vessel is modeled as

$$\dot{w}_F = \frac{1}{\tau_F}(w_{FSP} - w_F) \tag{3.79}$$

The feedwater is limited by

$$w_F \leq 1.15 w_{FR}$$

where

$w_{FR}$ = rated value of feedwater flow

From a mass balance on the volume $V_d$, in Equation 3.61, one gets

$$\dot{m}_{fd} = \frac{m_{fp}}{m_{fp} + m_{gp}} w_{pe} - w_{dw} \tag{3.80}$$

Water level in the pressure vessel is given by

$$H_c = \frac{m_{fd} v_f}{A_c} - H_{co} \tag{3.81}$$

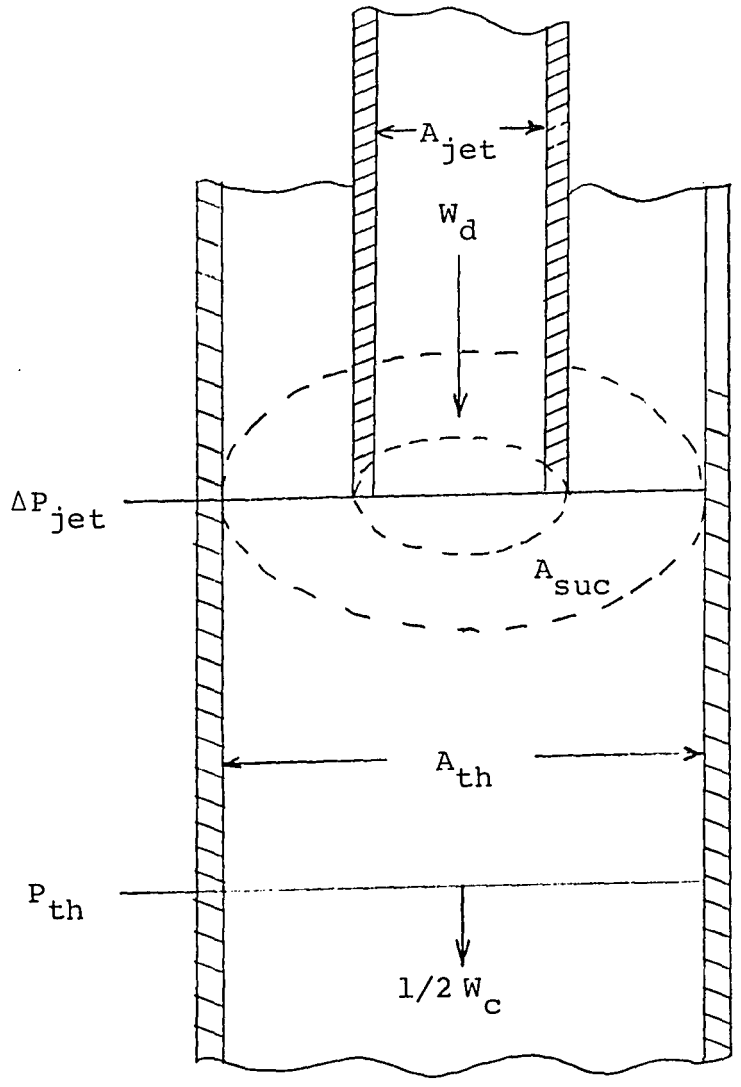
where

$H_{co}$ = some suitable reference value

3.5.8 Jet Pump and Core Inlet Mass Flow

The plant is assumed to have two identical recirculation loops. The idealized geometry of the nozzle, suction channel and throat regions is shown in Figure 3.3. A mass flow of $W_d$ enters each loop, and by writing momentum balance for the jet pump and the recirculation loop, the pressure drop across the jet pump mixing region is found for each loop as:

$$\Delta p_j = \frac{v_{sub}}{32.2 A_{th}} \left[ \frac{(1/2 w_c - w_d)^2}{A_{suc}} + \frac{w_d^2}{A_{jet}} - \frac{w_c^2}{4 A_{th}} \right] \tag{3.82}$$




$$\Delta P_{jet} = P_{th} - P_{jet}$$

Figure 3.3. Jet pump throat diagram.

where

$A_{th}$ = jet-pump throat area

$A_{suc}$ = jet-pump suction area

$A_{jet}$ = jet-pump nozzle area

The core mass flow is found from a balance of the pressure acting on the fluid in the core

$$\dot{w}_c = \frac{32.2}{a_{11}}[\Delta p_j + \frac{\Delta Z}{v_{sub}} - K_{suc} v_{sub}(1/2\ w_c - w_d)^2 - (1/4)K_{dif} v_{sub} w_c^2 - 144\Delta p_c] \quad (3.83)$$

where

$a_{11}$ = reactor recirculation loop length/area ratio

$\Delta Z$ = reactor recirculation static head

$K_{suc}$ = jet-pump suction loss coefficient

$K_{dif}$ = jet-pump diffuser loss coefficient

$\Delta P_c$ = core pressure drop

For a two-phase flow, the total pressure drop is given by the sum of the three components (El-Wakil, 1971).

$$\Delta p_{total} = \Delta p_{friction} + \Delta p_{acceleration} + \Delta p_{elevation} \quad (3.84)$$

The frictional pressure drop, $\Delta p_{friction}$, in the core is calculated using Darcy-Weisbach formula for the single-phase flow and the two-phase friction multiplier given by Lottes-Flinn correlation.

The jet pump mass flow is modeled as

$$\dot{w}_d = \frac{1}{\tau_{j1}}(w_{dsp} - w_d) \tag{3.85}$$

where

$\tau_{j1}$ = time constant for jet pump response

The jet flow is limited by

$$W_d \leq 1.2\ w_{dR}$$

where

$w_{dR}$ = rated value of jet flow

An upper limit on $|\dot{w}d|$ is introduced as

$$|\dot{w}d| \leq \dot{w}_{drmax}$$

where

$\dot{w}_{drmax}$ = maximum rate of change in jet flow

3.5.9 Turbine Power Output

The turbine is modeled to respond with a single time constant corresponding to changes in the HP turbine inlet pressure. The detailed dynamics of the HP and LP turbines, the moisture separators, the generator, etc., are thus all lumped into this one time constant. Such a simple model precludes turbine frequency control, which could be included if properly modeled. The model here simply considers the turbine to "follow" the steam supply, and thus mainly controls the nuclear system such that it meets the steam

demand. However, it is not the steam flow rate that is of most importance, it is its delayed influence on the turbine power.

The dynamic of the turbine is, therefore, modeled as follows: the power production in the LP-turbine relative to its rated value is

$$\dot{q}_t = \frac{1}{\tau_{Lp}}\left(\frac{P_B}{P_{BR}} - q_t\right) \tag{3.86}$$

where

$P_B$ = pressure at inlet to HP-turbine

$P_{BR}$ = pressure at inlet of HP-turbine at rated power

and

$\tau_{Lp}$ = time constant for power response of LP-turbine

The time constant $\tau_{Lp}$ mainly accounts for the time needed to change the pressure in the volume between HP and LP-turbine, such as the moisture separator volume.

The total power as measured by sensors, relative to the rated value, is now given by

$$\dot{q}_T = \frac{1}{\tau_{pw}}(q_{it} - q_T) \tag{3.87}$$

where

$$q_{it} = f_{Lp}q_t + (1 - f_{Lp})\frac{P_B}{P_{BR}}$$

$\tau_{pw}$ = time constant for generator power sensors

$f_{Lp}$ = fraction of rated load produced in LP-turbine

3.5.10 Steam Line Steam Flow

A lumped pressure node approach is taken to calculate the pressure rate in the steam line. If steam inertia and kinetic energy changes are ignored, the steam flow out of the vessel is of the form

$$w_l = K_{SL} \sqrt{\frac{|P_v - P_{sL}|}{v_g}} \; \text{sign}(P_v - P_{SL}) \qquad (3.88)$$

where

$K_{SL}$ = steam line conductance from vessel to half way to turbine

and

$P_{SL}$ = pressure in steam line half way to turbine

Similarly, for the steam flow through the second half of the steam line

$$w_{SL} = K_{SL} \sqrt{\frac{|P_{sL} - P_T|}{v_g(P_T)}} \; \text{sign}(P_{SL} - P_T) \qquad (3.89)$$

The expression for the steam flow through the turbine valve considers the possibility of reverse flow if $P_B > P_T$, and considers the flow to be choked if $P_T > 0.6\ P_B$.

For $P_T > P_B$, the turbine valve flow rate is given by (Linford, 1973):

$$W_{CT} = C_{TV} a_t \sqrt{\frac{P_T^{a10}}{v_g(P_T)}} \tag{3.90}$$

$$P_{rt} = \frac{P_B}{P_T}$$

If $P_T < P_B$,

$$W_{CT} = -C_{TV} a_t \sqrt{\frac{P_B^{a10}}{v_g(P_B)}} \tag{3.91}$$

$$P_{rt} = \frac{P_T}{P_B}$$

For both equations above

$$a_{10} = P_{rt}^{a7}(1 - P_{rt}^{a6})/a_6 \tag{3.92}$$

and

$$C_{TV} = a_8 + a_9 P_{rt}$$

The turbine flow coefficients $a_6$, $a_7$, $a_8$, and $a_9$ are taken from Baumister et al. (1981). Flow through the bypass valve is obtained from

$$W_{BT} = .7628 a_B \sqrt{\frac{P_T}{v_g(P_T)}} \tag{3.93}$$

In Equations 3.89 through 3.93, the pressure $P_T$ has been assumed known. However, there is no explicit equation for $P_T$; instead, we consider $P_{SL}$ and $P_B$ as known and obtain $P_T$ by iteration on Equations 3.89 through 3.93 until

$$W_{SL} = W_{CT} + W_{BT}$$

is approximately satisfied.

Pressure rate equations for $P_B$ and $P_{SL}$ are obtained from a mass balance assuming no condensation or vaporization in the steam line. Then, from the continuity equation, we can write

$$\dot{P}_{SL} = \frac{(w_1 - w_{SL}) v_g^2 (P_{SL})}{-[\frac{dv_g}{dp}] V_{SL}} \tag{3.94}$$

where

$V_{SL}$ = total steam line volume from reactor vessel to turbine

Similarly,

$$\dot{P}_B = \frac{(w_{CT} - w_T) v_g^2 (P_B)}{-[\frac{dv_g}{dp}] V_B} \tag{3.95}$$

where

$W_T$ = steam flow through the turbine

$$= C_T \sqrt{\frac{P_B}{v_g(P_B)}}$$

and

$V_B$ = turbine bowl volume

### 3.5.11 Turbine and Bypass Valve Dynamics

The response of the turbine valve is modeled as:

$$\dot{a}_t = \frac{1}{\tau_{TV}} (a_{tsp} - a_t) \tag{3.96}$$

where

$\tau_{TV}$ = turbine valve time constant

The turbine valve is subjected to the conditions

$$0 \leq a_t \leq A_t$$

$$|\dot{a}_t| \leq V_{Tmax}$$

where

$V_{Tmax}$ = maximum rate of change of turbine valve

The response of the bypass valve is modeled as

$$\dot{a}_B = \frac{1}{\tau_{BV}}(a_{BSP} - a_B) \tag{3.97}$$

where

$\tau_{BV}$ = bypass valve time constant

The bypass valve is subjected to the conditions

$$0 \leq a_B \leq A_B$$

$$|\dot{a}_B| \leq V_{Bmax}$$

where

$V_{Bmax}$ = maximum rate of change of bypass valve

Saturation enthalpies and specific volumes as a function of pressure are obtained from a linear relation of the form

$$V = V_R + (\frac{dV}{dp})(P - P_R) \tag{3.98}$$

$$h = h_R + (\frac{dh}{dp})(P - P_R) \tag{3.99}$$

where $V_R$ and $h_R$ are specific volume and specific enthalpy, respectively, at some reference pressure $P_R$.

Similarly, for the water saturation temperature in the vessel

$$T_{sat} = T_{satR} + [\frac{dT_{sat}}{dp}](P_V - P_R) \tag{3.100}$$

where

$T_{satR}$ = saturation temperature at pressure $P_R$

The set of equations developed in this chapter is of the form

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

They consist of a large number of algebraic equations and a set of 48 first-order differential equations. Though smaller in size compared with a typical BWR power plant (BWR/6, 1973), it is adequate to perform transients for small changes in the operating point.

# CHAPTER 4

# RESULTS AND DISCUSSION

As presented in an earlier chapter, the BWR model consists of 48 nonlinear first-order differential equations and a large number of algebraic equations. Most of the steam properties were represented by approximate linear relations where possible; otherwise, table lookups were used. In all cases, the systems of equations were simulated in the DARE P environment (Korn and Wait, 1978), using the EPISODE (Hindmarsh and Byrne, 1975) integration routine.

Results for selected perturbations are presented in graphs in this chapter. The first set of these graphs, is the transient of the open loop system for the original nonlinear system. The second set of transients will be the comparison of the nonlinear model and its linear version. This will be followed by the transient of the closed-loop system with the suboptimal controller.

## 4.1 Plant Transient Without Compensator

The response for the open loop system of some of the variables for a 5 percent step increase in the recirculation flow is shown in Figures 4.1-4.4. The response appears to

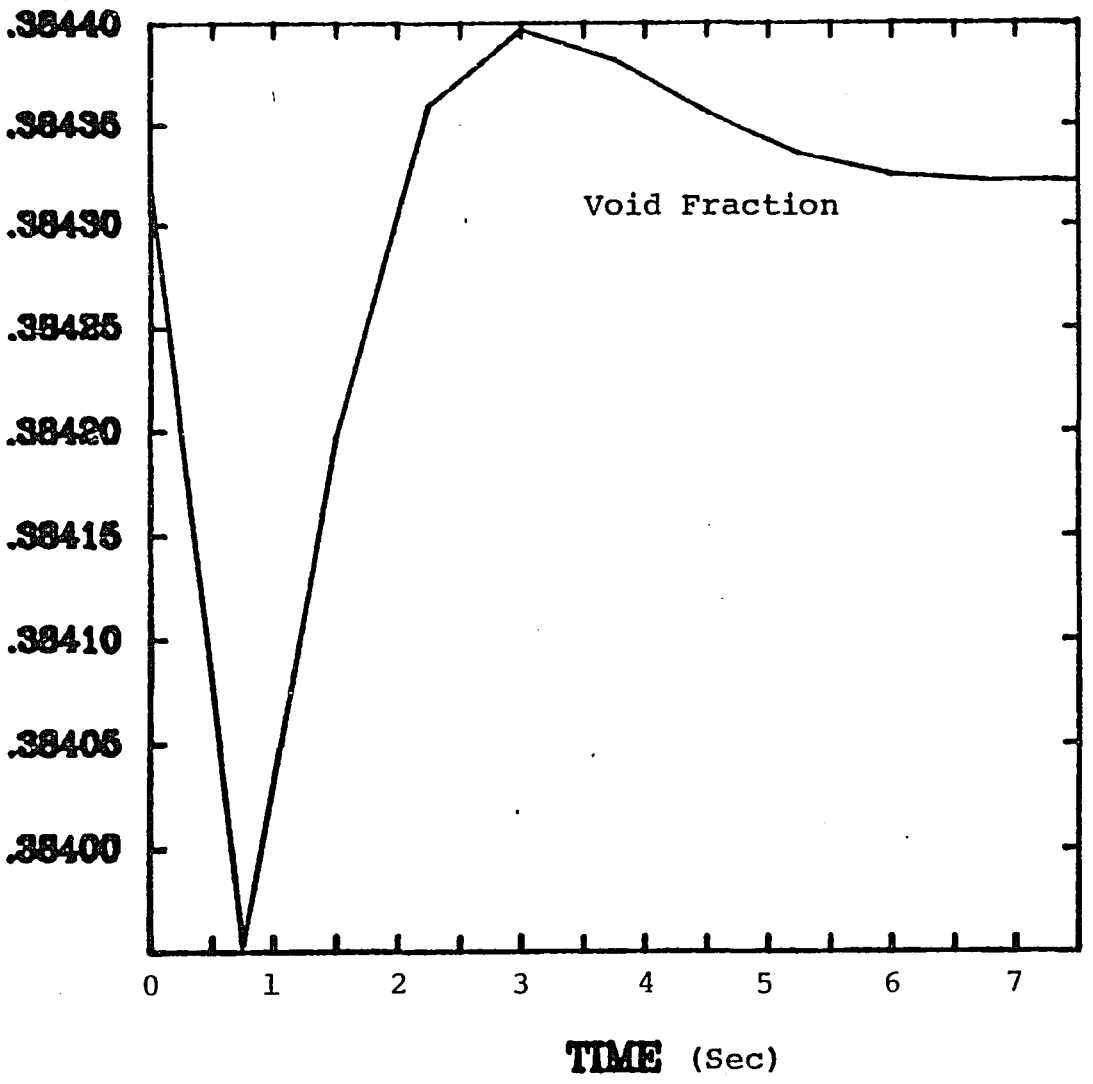



Figure 4.1. Void fraction transient, following a 5% step increase in the recirculation flow.

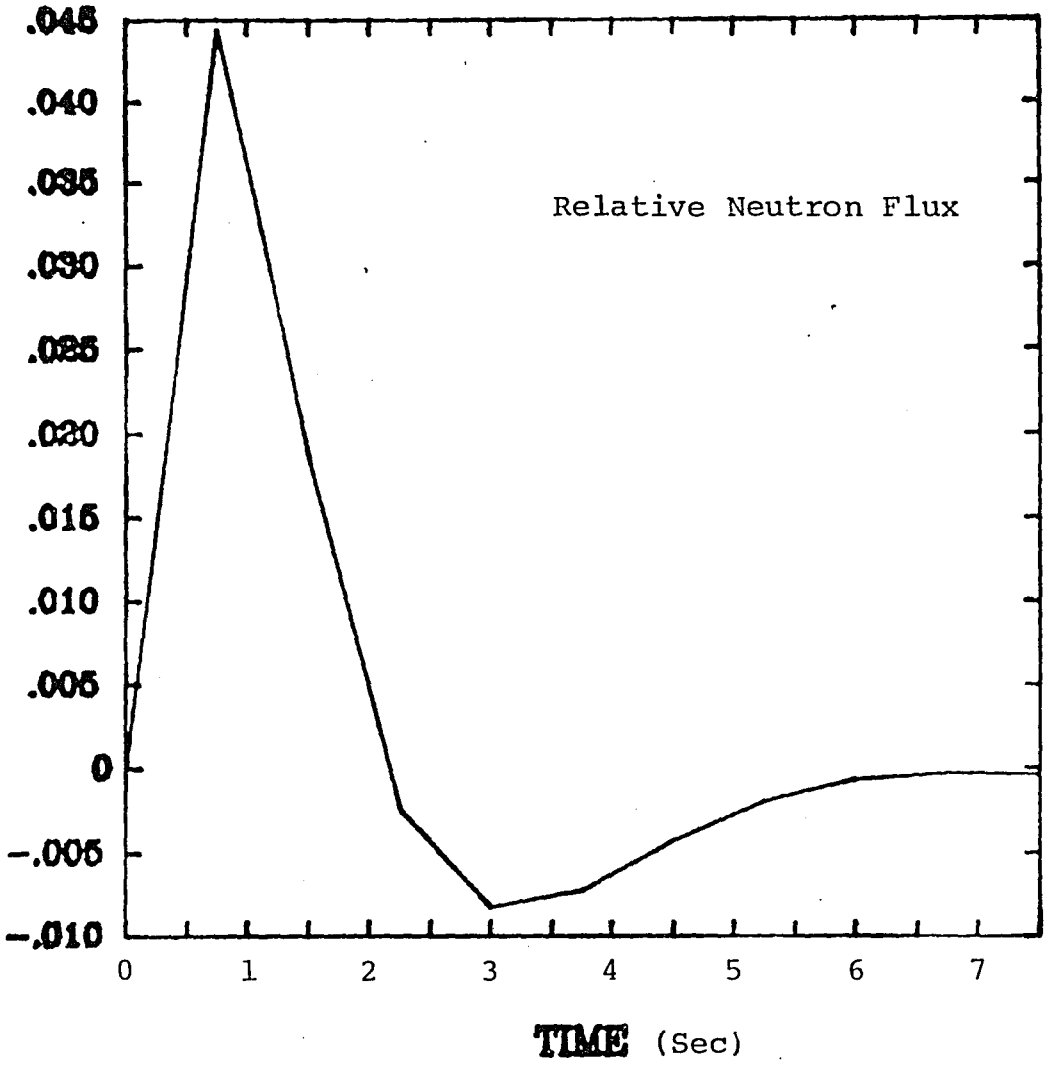



Figure 4.2. Neutron flux transient, following a 5% step increase in the recirculation flow.

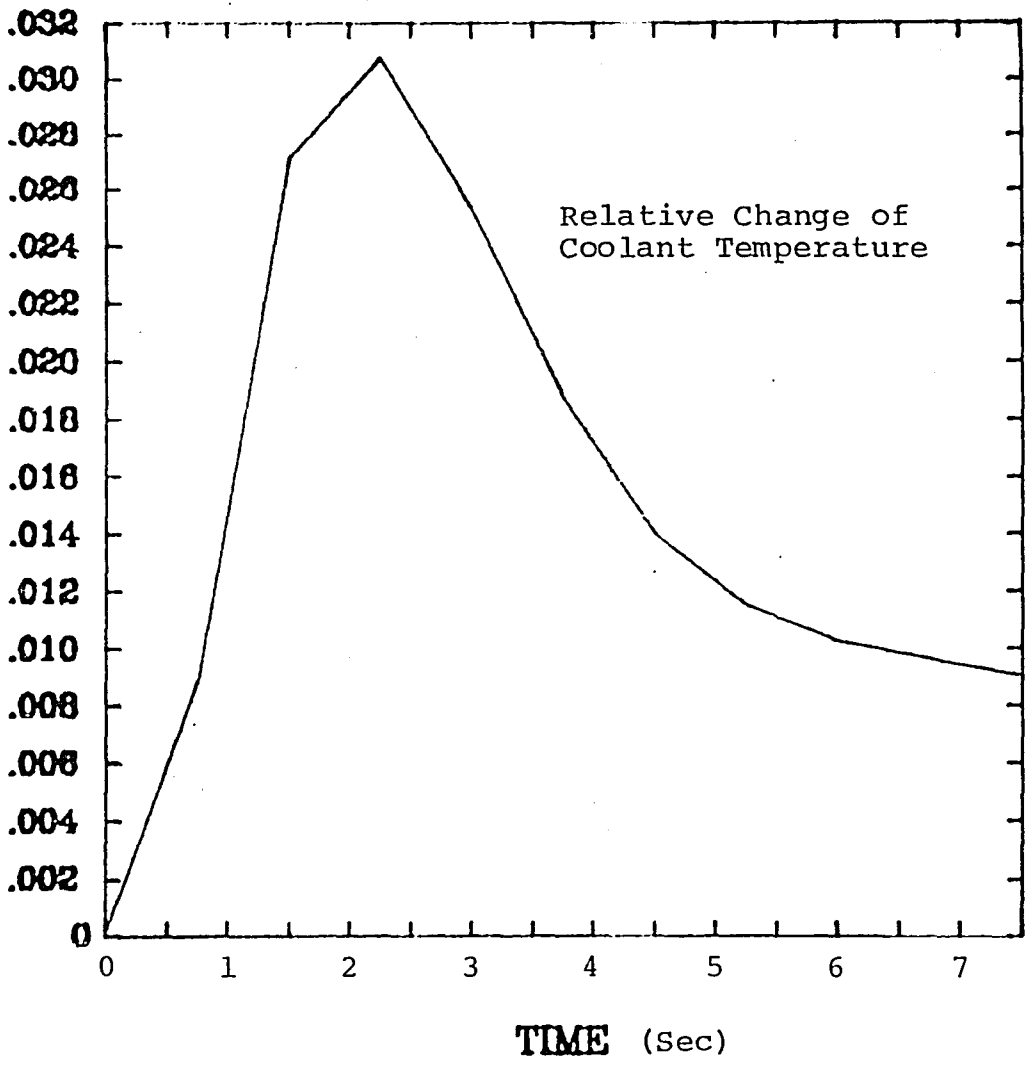



Figure 4.3. Average coolant temperature transient, following a 5% increase in the recirculation flow.

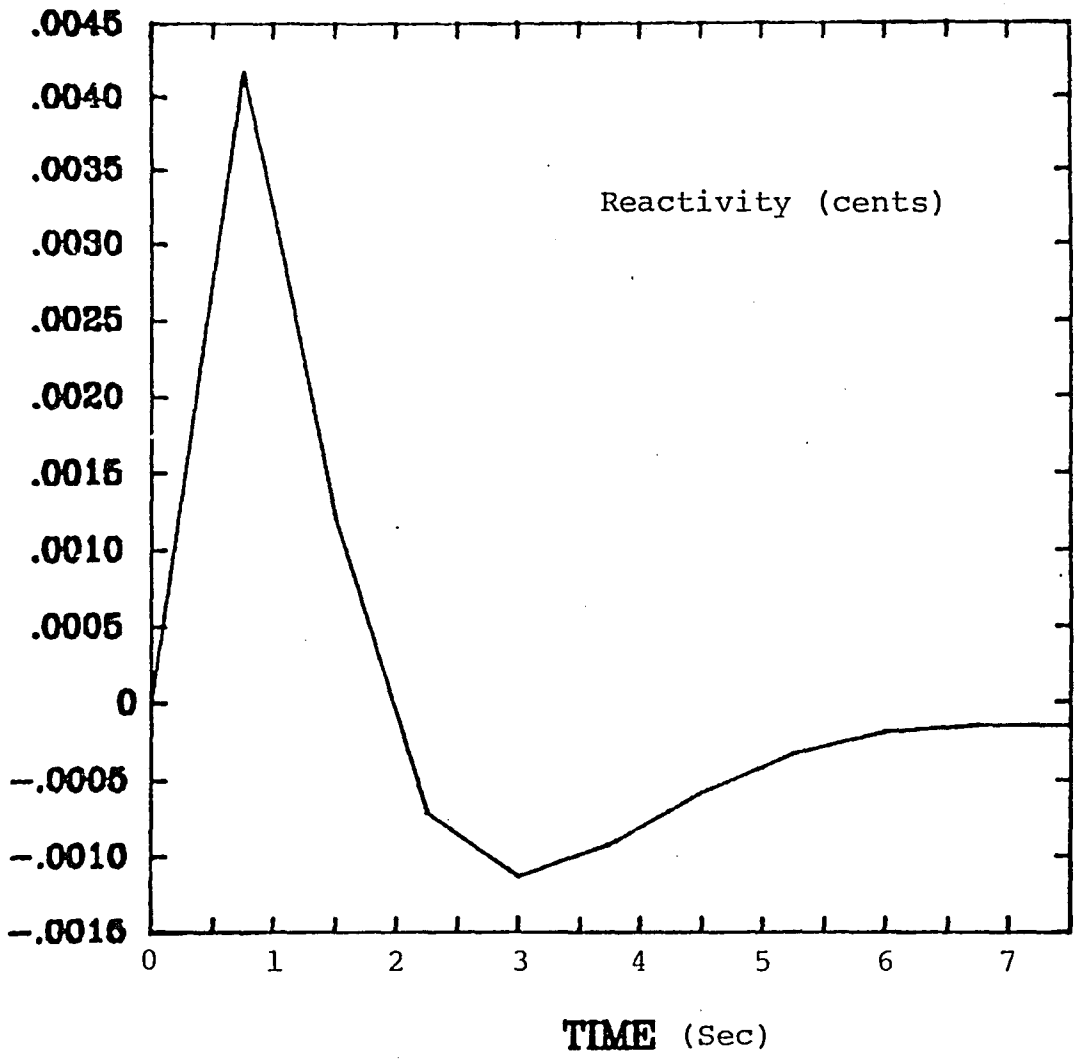



Figure 4.4. Reactivity transient, following a 5% step increase in the recirculation flow.

be consistent with the plant transient results for similar plants (Linford, 1973).

An increase of 5% of the recirculation flow causes a decrease in the average void fraction, Figure 4.1. The effectiveness of the coolant to thermalize increases and therefore the neutron flux rises, Figure 4.2. The coolant average temperature rises slowly to a maximum and decreases; its delay effect is clear, shown in Figure 4.3, where the lag is about 1 sec behind the neutron flux. Figure 4.4 shows the reactivity transient in dollars.

## 4.2 Comparison of Nonlinear Model with its Linear Version

Linear-Quadratic Optimal control theory demands a linear model of the processes. The subroutine (EIGEN) in the DARE P package (Korn and Wait, 1978) calculates the Jacobian of the nonlinear system. The writer modified this subroutine to calculate also the B-matrix at some selected inputs. With the system linearized about the equilibrium state, the two models were studied when subjected to small perturbations. Figures 4.5-4.7 show the results of the transients. It was observed that at a 5% increase in the circulation flow, the linear model fairly represents the original nonlinear system, Figure 4.7. For larger increases in the recirculation flow, the linearized model represents the nonlinear system less accurately (Figures 4.6 and 4.7).

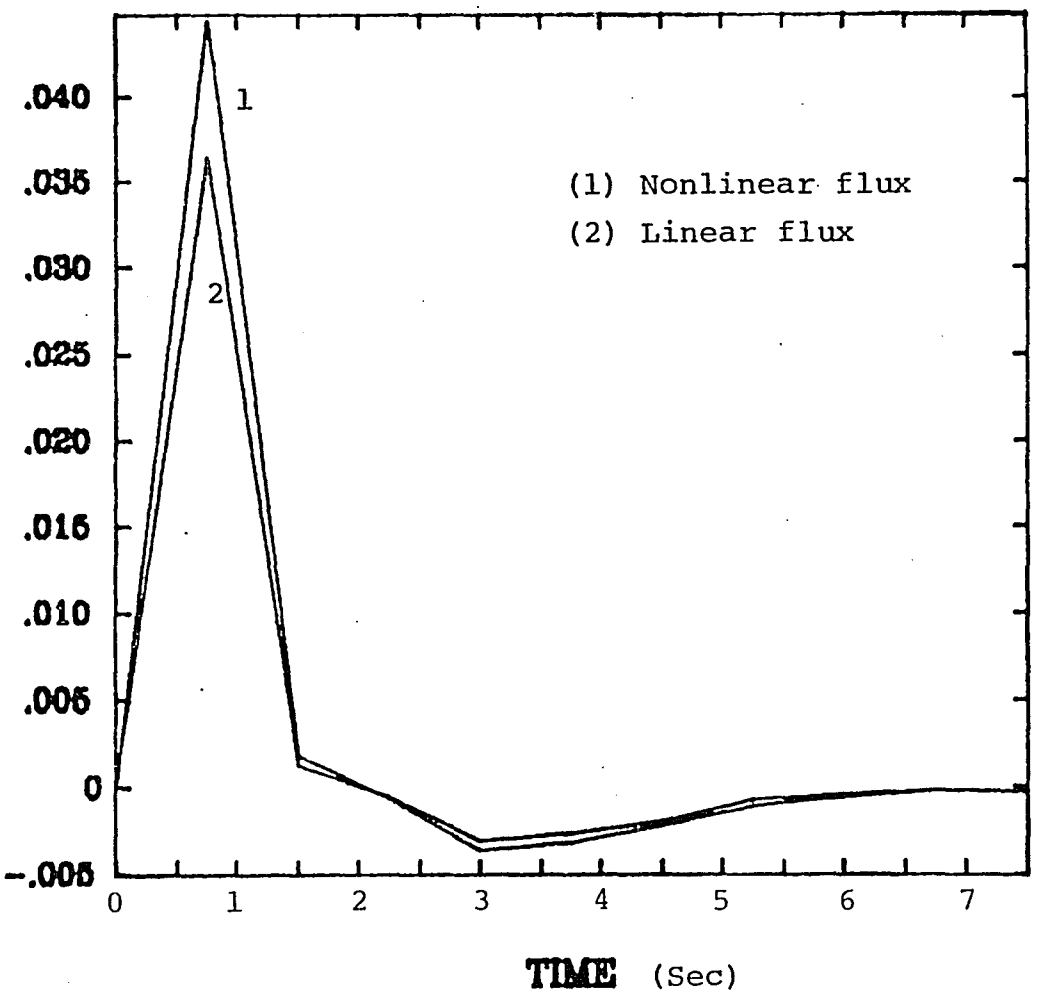



Figure 4.5. Transient for a 5% step increase in the recirculation flow.

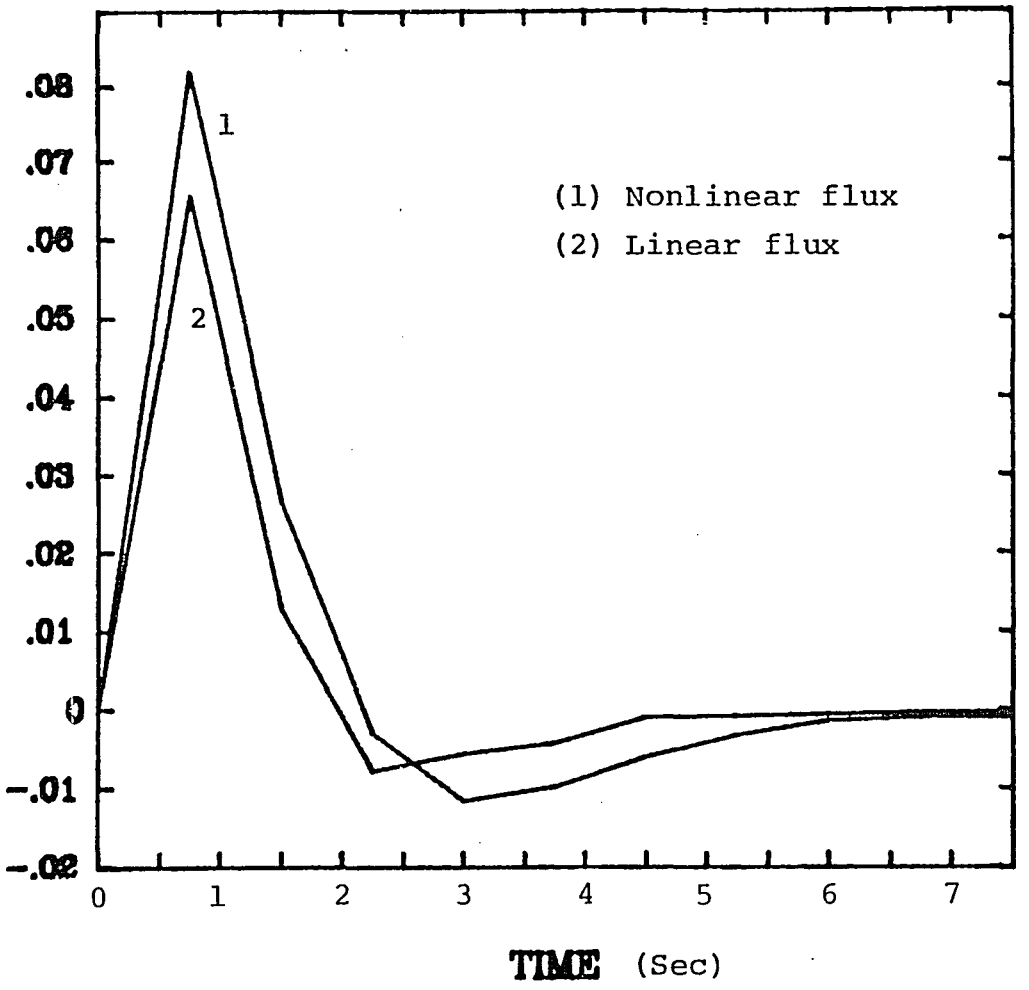



Figure 4.6. Transient for a 10% step increase in the recirculation flow.

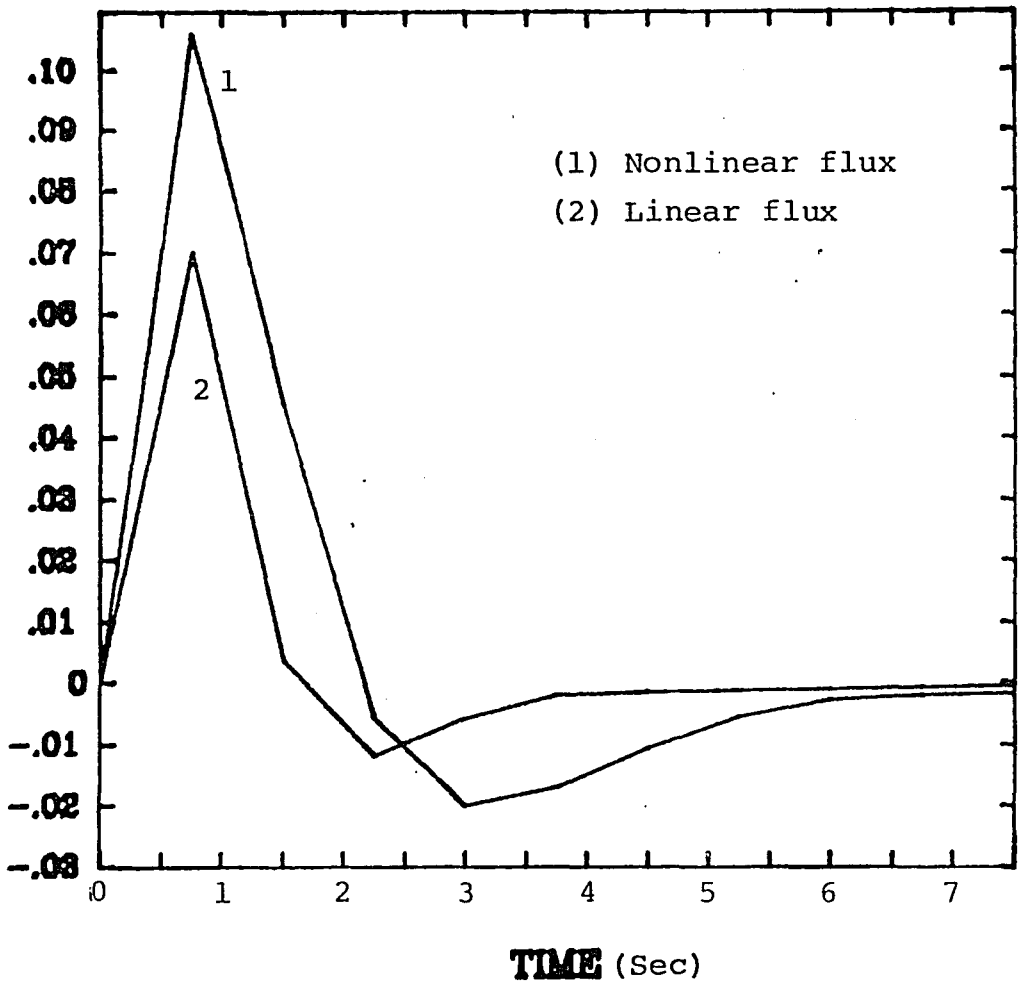



Figure 4.7. Transient for a 20% step increase in the recirculation flow.

## 4.3 Implementation of the Partitioning Scheme

The scheme for partitioning a nonlinear system into "fast" and "slow" subsystems described in Chapter 2 has been implemented in a DARE P input file in a subroutine known as EIGEN. The procedure may be summarized in the flow chart of Figure 4.8. The EISPACK routine referred to in the chart is a well known eigenvalue-eigenvector subroutine employing the QR algorithm (Smith et al., 1976).

As may be noted from the definition of $L = -M_{21}M_{11}^{-1}$, numerical difficulties arise if $M_{11}$ is singular or almost singular. To avoid these difficulties, it is desirable that $||M_{11}||$ be as large as possible. This means that the variables must be reordered before the partitioning is carried out. This reordering is done by means of the "coupling coefficients" which are explained as follows. We first normalize the columns of $[M_{11}|M_{21}]^T$ so that each eigenvector has unit norm. We then compute the row norms of this matrix and call these norms the coupling coefficients. Now to reorder the system variables, we first rearrange the coupling coefficients so that they are in decreasing order, and then use this new ordering to determine a state permutation matrix (SPM). Finally, we apply the SPM to the state variables, resulting in a reordering for which $||M_{11}||$ is maximized. This reordering also insures that the eigenvalues will be properly partitioned into the blocks

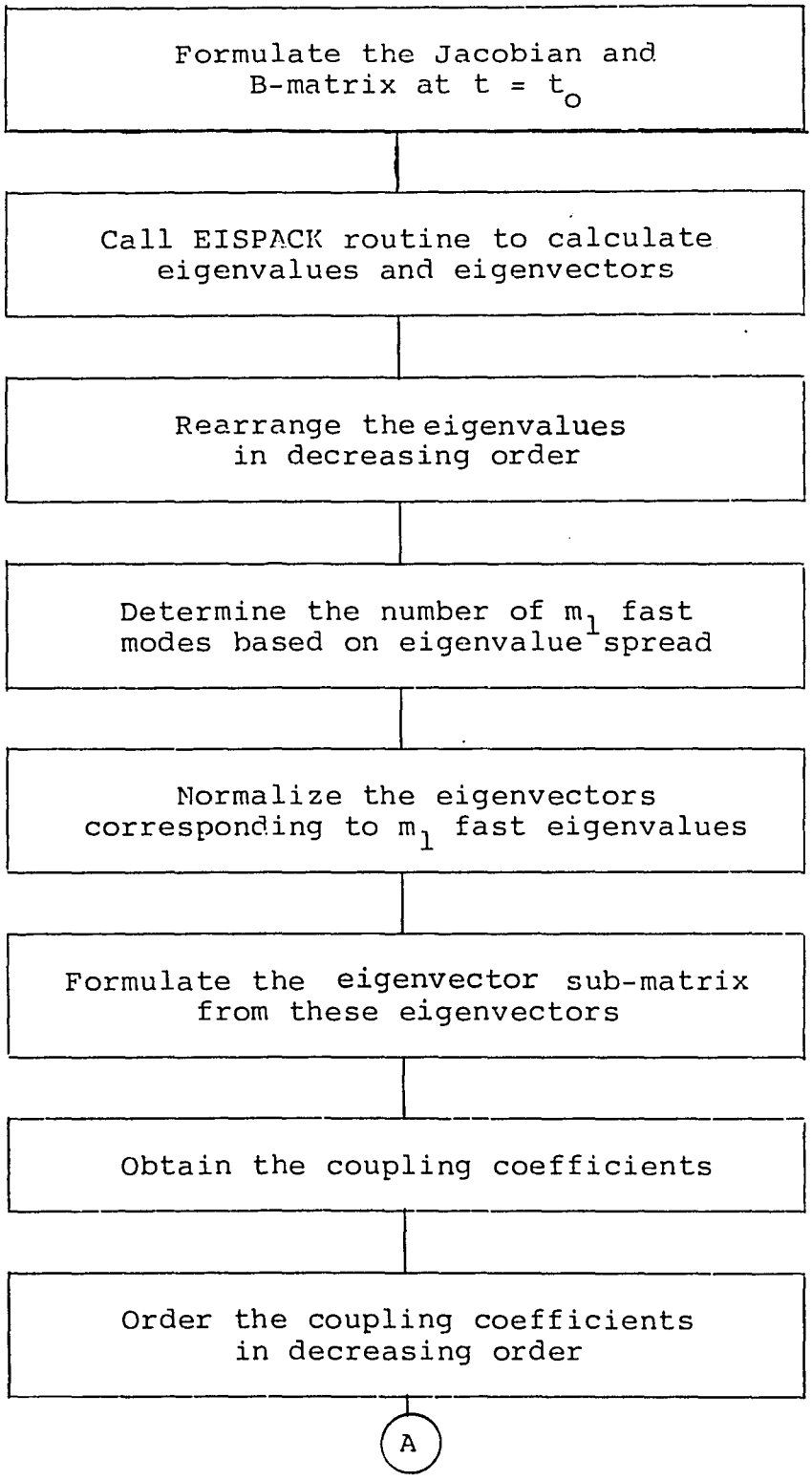



Figure 4.8. Flow chart to determine the transformation matrix "T."

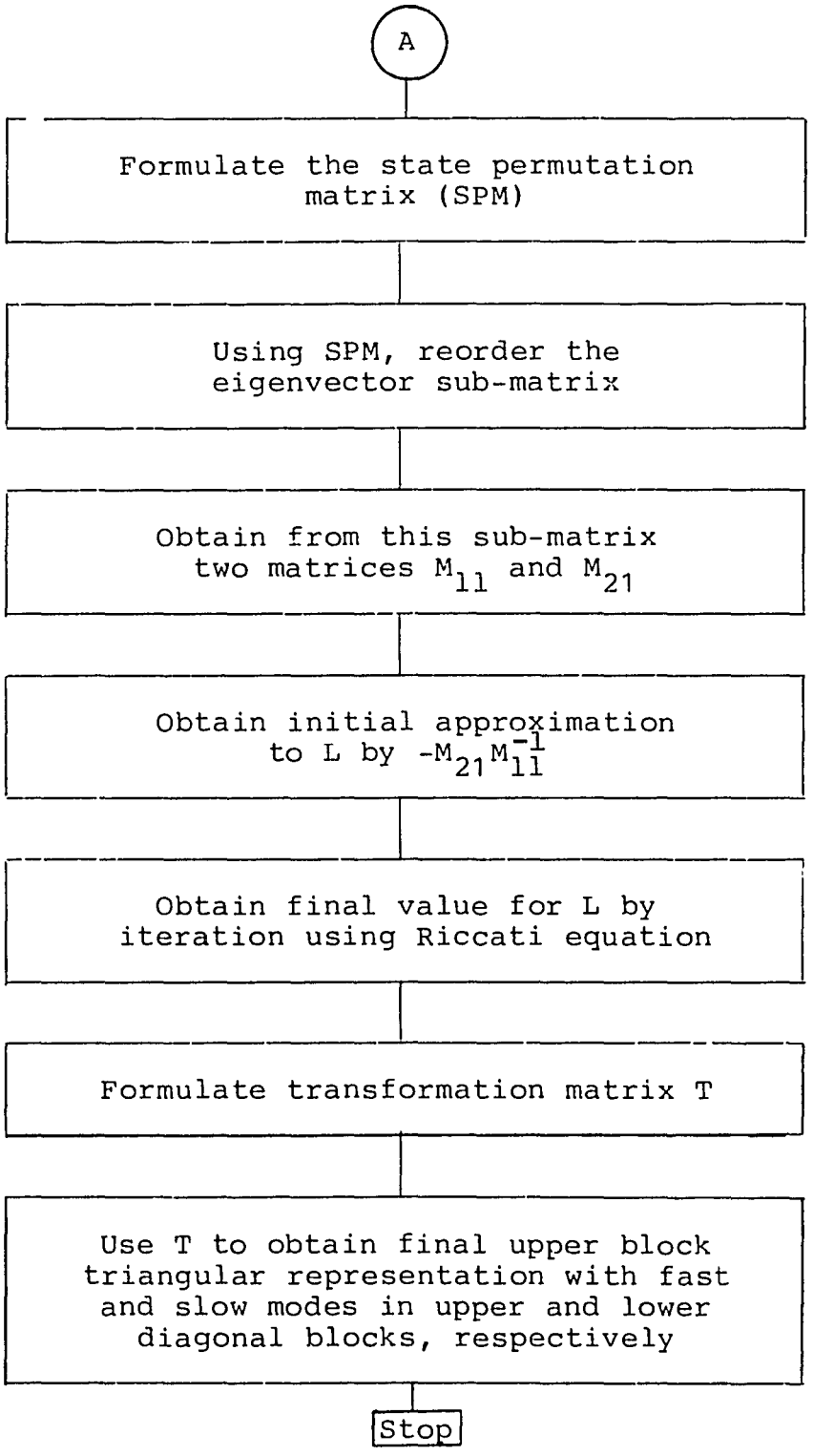



Figure 4.8--Continued

$J_1$, and $J_2$ of Equation 2.14. The theory as described in Chapter 2 predicts that the fast and slow eigenvalues will be contained in blocks $\tilde{A}_{11}$ and $\tilde{A}_{22}$, respectively (see Equation 2.18). Numerical tests were carried out to verify this, with pleasing results. The eigenvalues were properly placed into blocks $\tilde{A}_{11}$ and $\tilde{A}_{22}$, and, furthermore, there was very close agreement between the eigenvalues of these blocks and the original system eigenvalues. Differences in the eigenvalues are due to the computational difficulties in the evaluation of eigenvectors for a 48th order system.

## 4.4 Compensator Performance

The slow system

$$\dot{z}_2 = \tilde{A}_{22} z_2 + (LB_1 + B_2)u$$

in Equation 2.18 was optimized using the optimal control algorithm, described in Chapter 2. In this work, the $B_1$-matrix was set to zero. The algorithm to solve the algebraic Riccati equation is given in Appendix D.

Initial set of weighting matrices, Q and R, were selected using Bryson's (1975) rule. Further entries were found by trial and error until satisfactory feedback gain was found. The suboptimized system was then coupled to the fast subset to see the performance of the complete system. The performance was satisfactory. Transformation back to the original state variables was done with the inverse

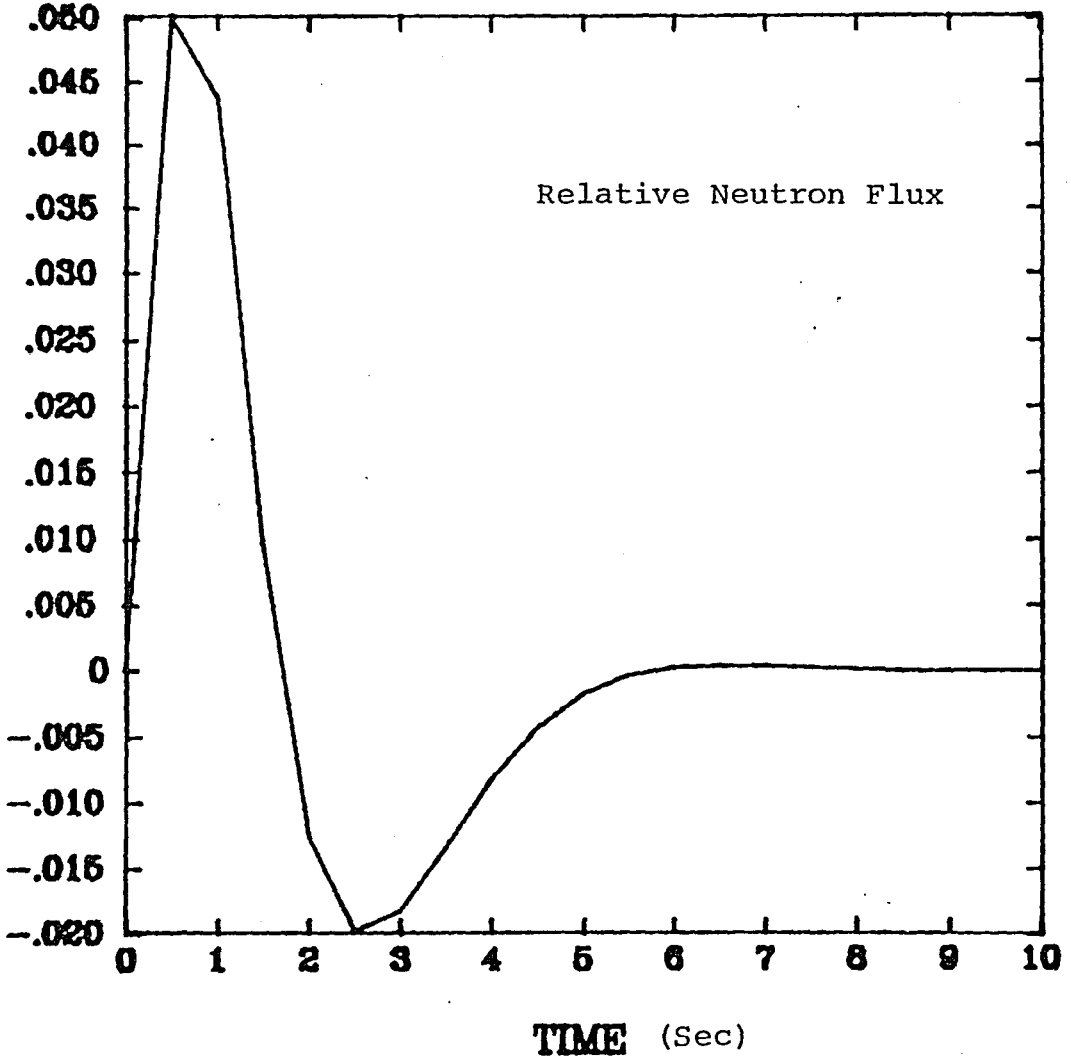



Figure 4.9. Compensator performance characteristics for the neutron flux.

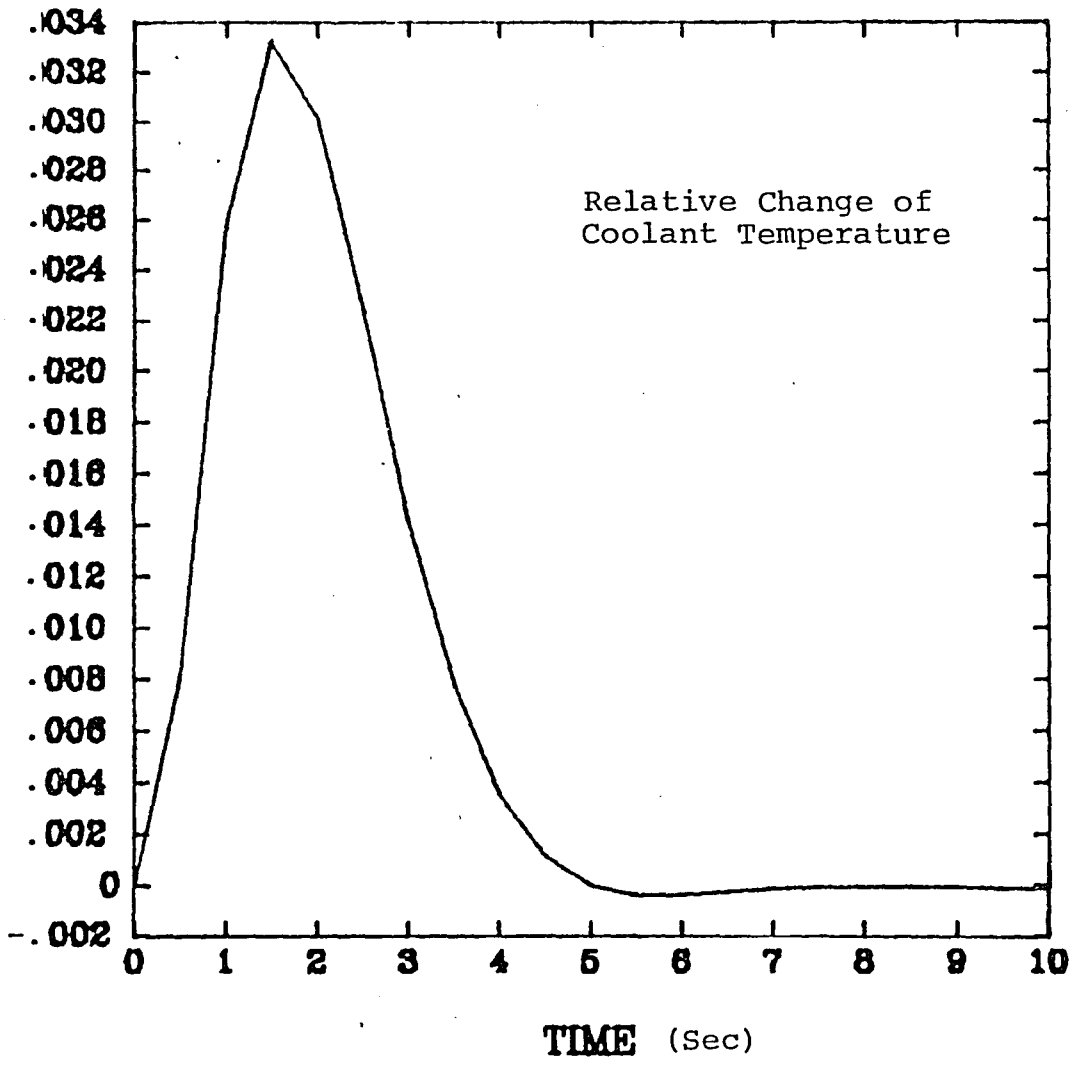



Figure 4.10. Compensator performance characteristics for the coolant temperature.

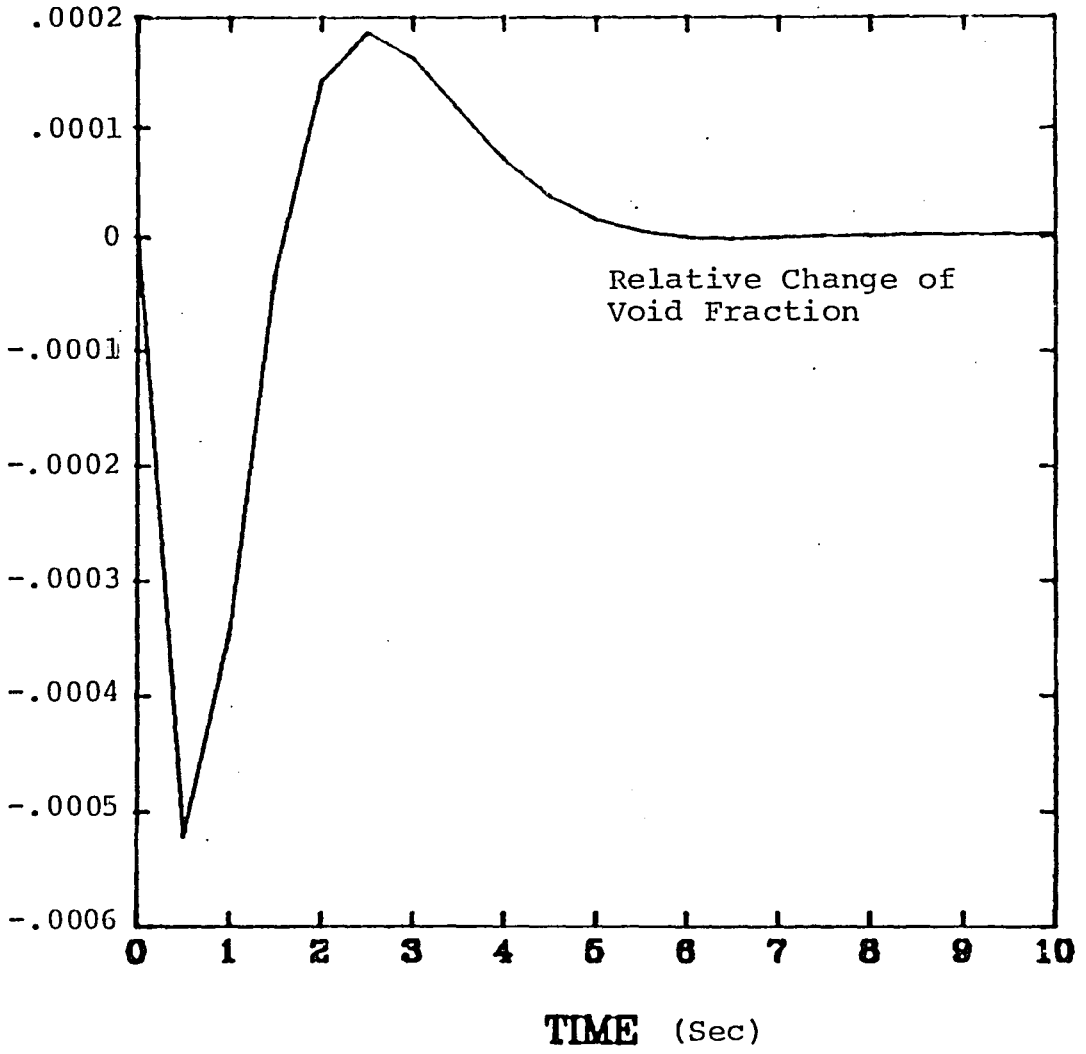



Figure 4.11. Compensator performance characteristics for the void fraction.

Figure 4.12. Closed-loop response to a 5% step increase of recirculating flow, with the proposed control system.

a. Relative neutron flux, page 83.

b. Relative change of coolant temperature, page 84.

c. Relative change of void fraction, page 85.

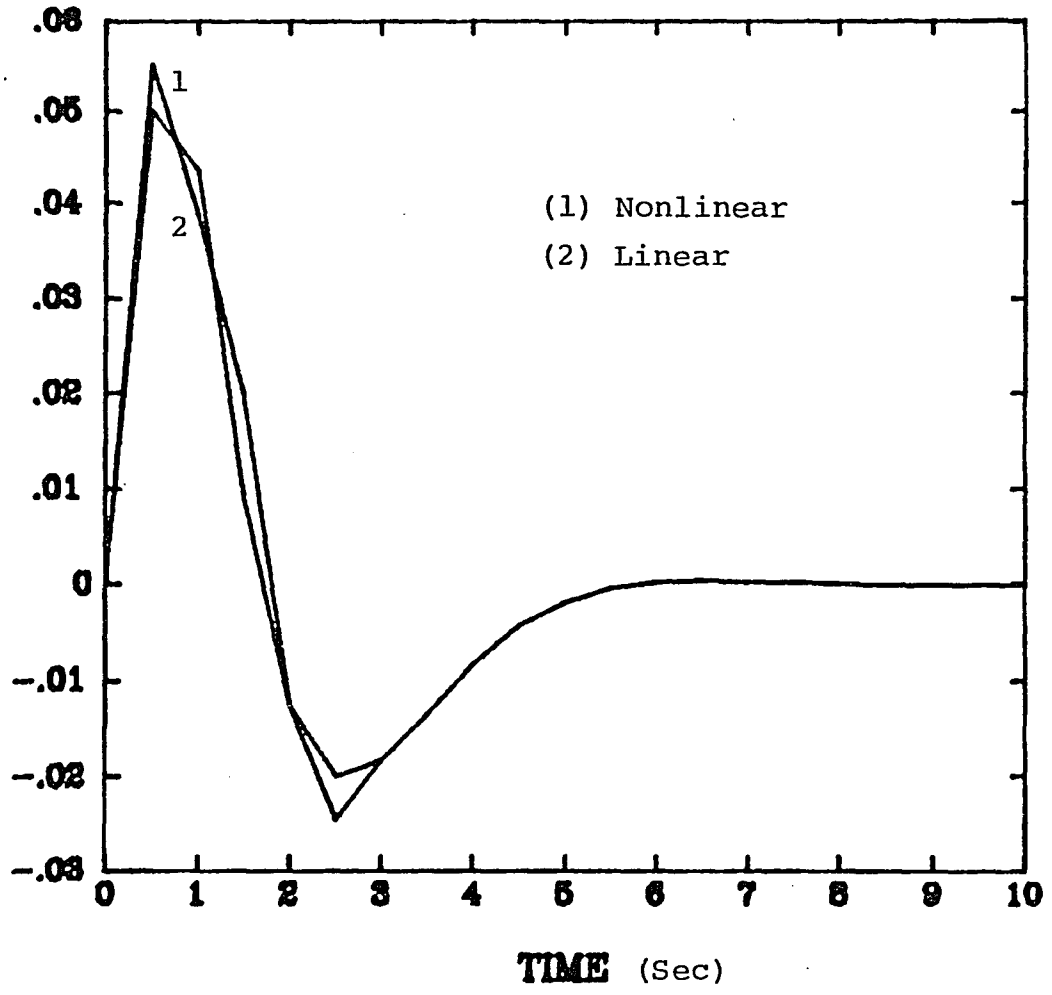



Figure 4.12a. Relative neutron flux.

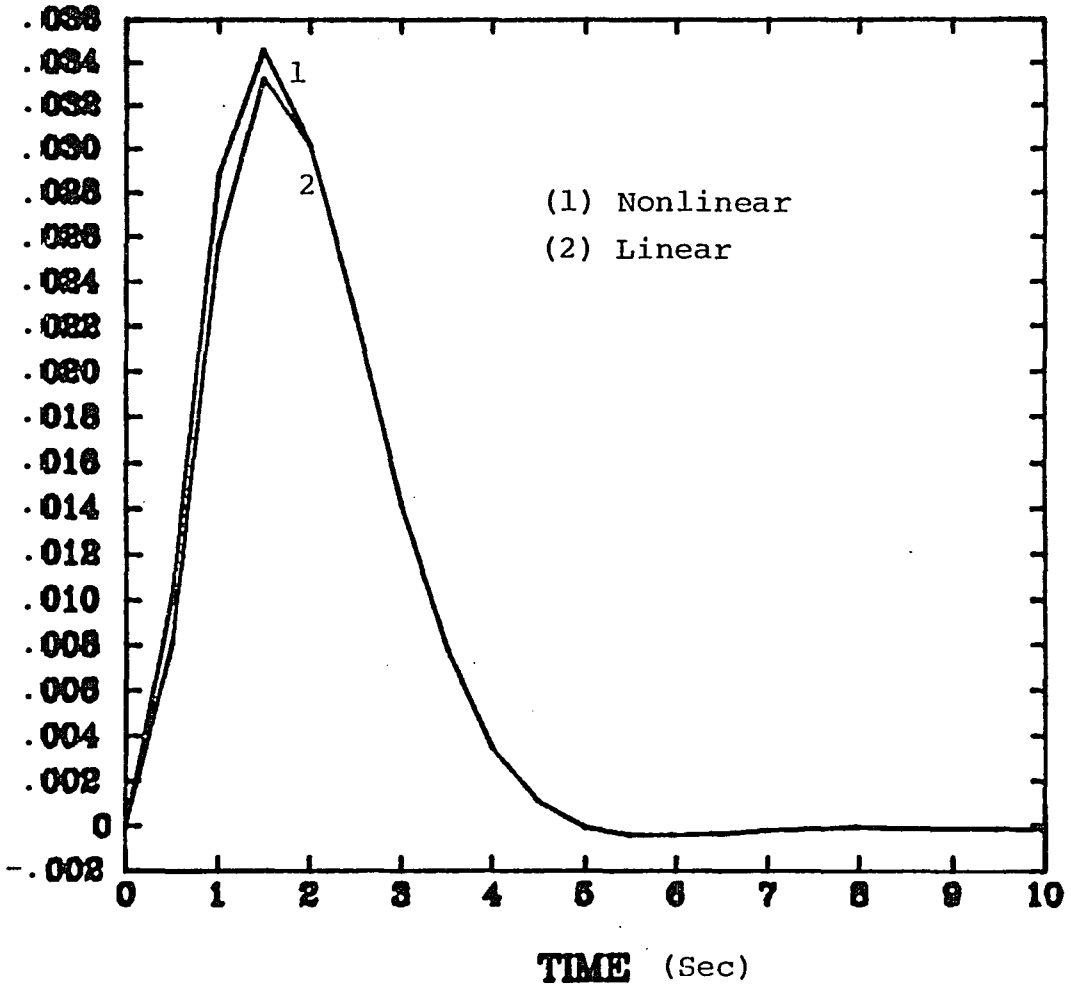



Figure 4.12b. Relative change of coolant temperature.

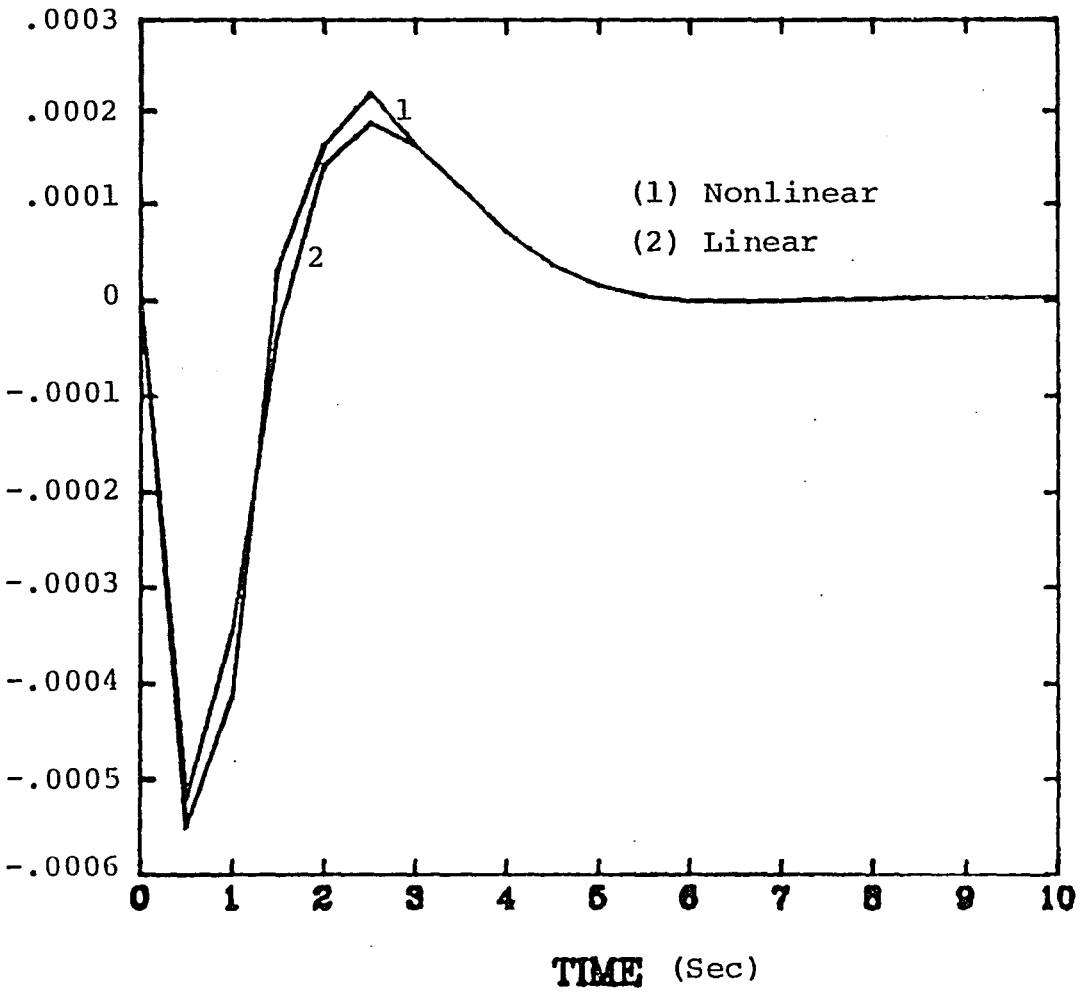



Figure 4.12c. Relative change of void fraction.

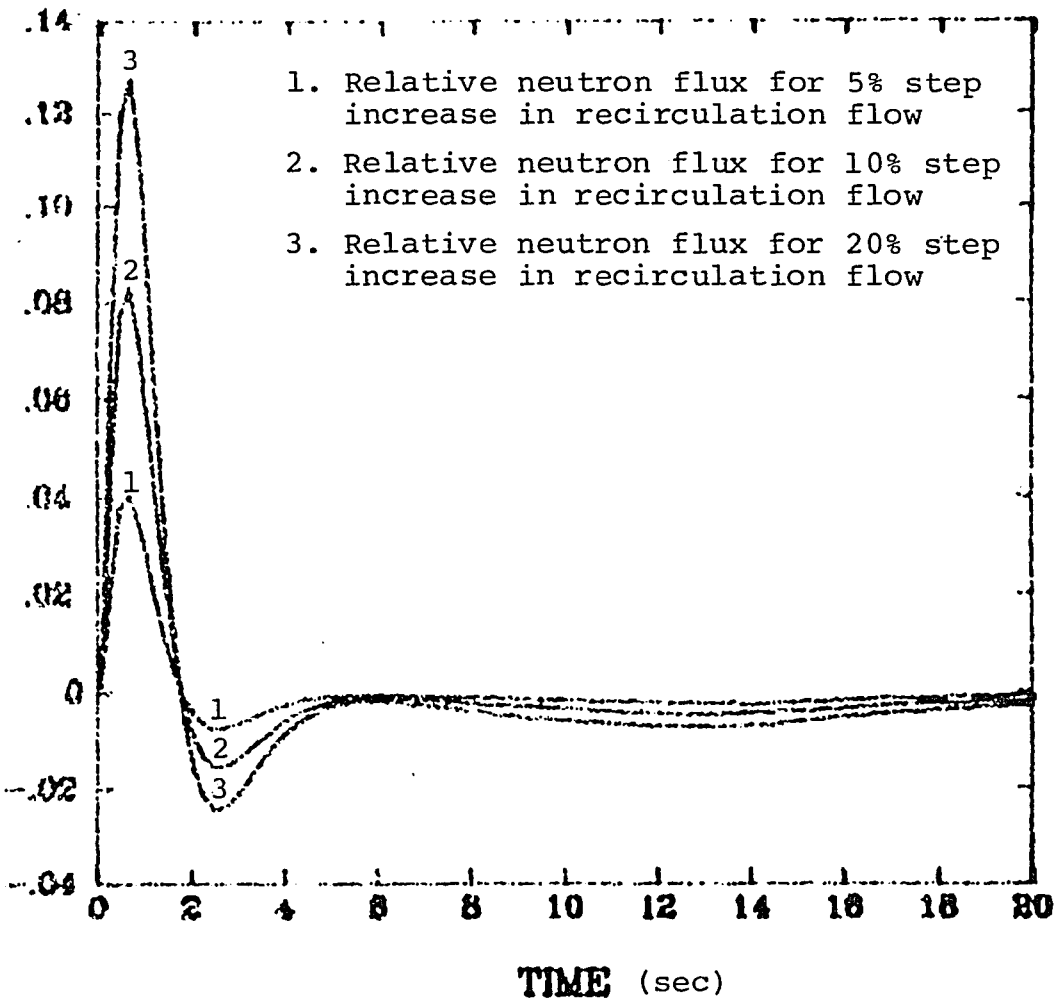



Figure 4.13. Open-loop system transients for the nonlinear model.

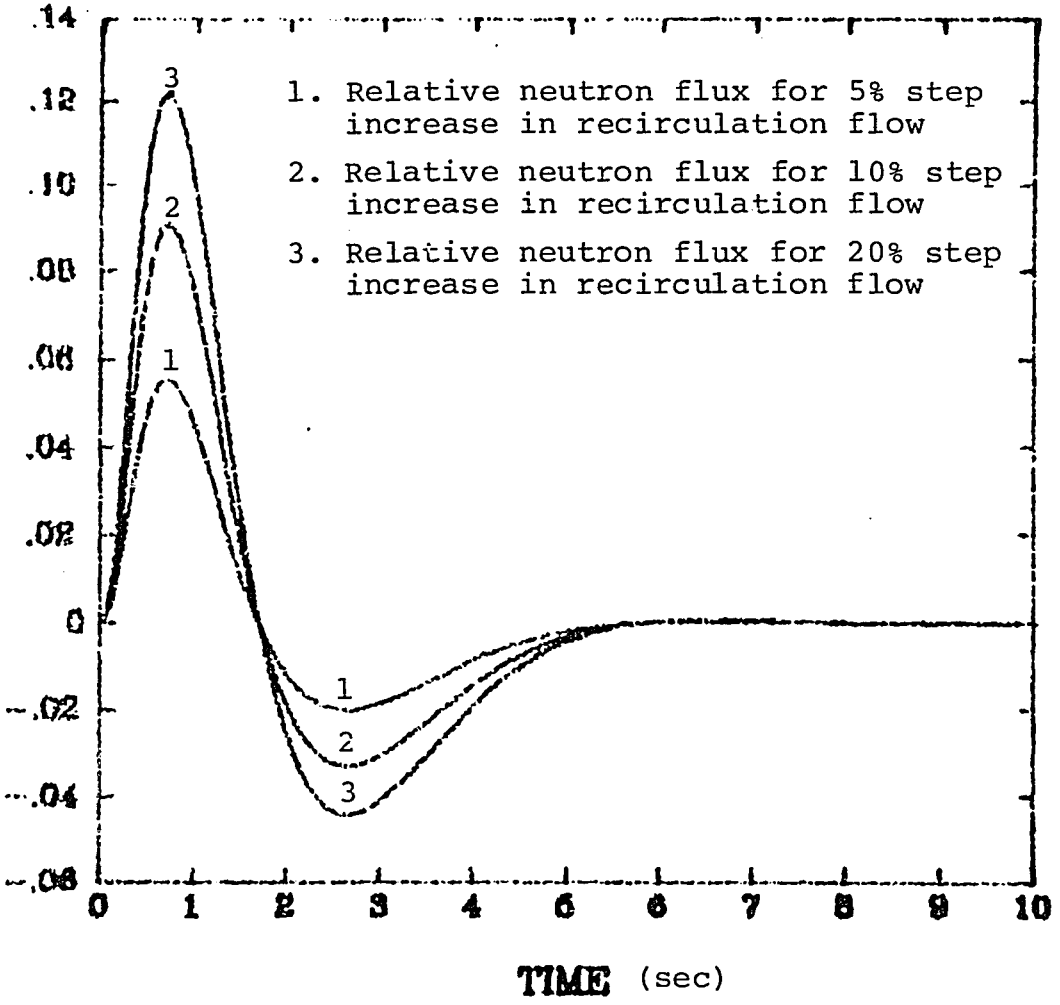



Figure 4.14. Closed-loop system performances for the proposed control model.

transformation T-matrix. A transient following a 5% step increase in recirculation flow is shown in Figures 4.9-4.11. In Figure 4.12, a transient with the proposed controller incorporated in the original nonliner model is shown. In all cases, an integral feedback action effectively returns the system to its original value, i.e., steady state. The steady state errors in the open-loop responses in Figure 4.3 are completely eliminated in Figure 4.14 with the proposed control system.

# CHAPTER 5

## CONCLUSIONS

A suboptimal controller has been developed for a non-linear simulation model of a BWR nuclear power plant (the Laguna Verde Nuclear Power Plant in Mexico). A linear model which fairly represents the original model has been developed about the equilibrium state. Using partitioning, the linear model has been transformed into a block triangular form.

It has been demonstrated that a suboptimal control can be utilized for the feedback of the simulated BWR plant. The manipulated variables have been circulation flow, reactivity, and the turbine bypass value. The variables to be controlled have mainly been the average neutron density, coolant average temperature and average void fraction.

A square performance index has been minimized to obtain a constant feedback gain which is optimal for a linear time invariant system. The obtained feedback law is able to compensate for a variety of plant transients. Initial entries in the weighting matrices Q and R can be selected using Bryson's rule or model-following techniques (Bryson, 1975). Further entries can be assigned by an

experienced engineer if he knows what plant response he desires, since the performance index actually reflects his objectives to a large extent. A variety of responses can be obtained by changing the weighting matrices.

The importance of obtaining a meaningful linear model is strongly emphasized. A compromise should be found when determining the order of the linear model; advantages in computations and ease in design of the compensator favor low-order models, while a fair representation of the physical effects tends towards a high-order model.

The concept of suboptimal, easily implemented, controllers is very attractive from a practical point of view, and there are several reasons for investigating the concept. One of the main reasons is that in the literature many procedures are proposed, but little technical experience about those is available so far. On the other hand, there seem to exist some technical advantages especially if multivariable systems are to be controlled simultaneously.

In a strongly coupled subsystem, bounds on the resulting suboptimality of performance and the norm of the interconnections should be established, since these determine whether the interactions are beneficial or nonbeneficial. The selection of results presented here is chosen mainly to illustrate what the transients look like, rather than as evidence of good performance. The main

control objective was to keep the mean square error response of the system as small as possible. Naturally, the system's perfrmance with the suboptimal controller should be compared with optimal performance; such a comparison could be the basis for deciding on the acceptabiilty of the proposed suboptimal design. Unfortunately, such a comparison could not be done in this work because that would involve solving the Riccati equation for a higher order system, which is very difficult to do.

# APPENDIX A

## MATRICES AND SOME EIGENVALUES FOR THE LINEAR MODEL

SLOW SUBSYSTEM MATRICES

$$A = \begin{bmatrix}
-10.0 & 10.0 & -11.6 & -108.8 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0767 & -0.0767 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0715 & 0.0 & -0.125 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & -2.72 & 5.17 & 1.19 & -2.178 & -4.62 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & -0.0833 & -0.0035 & 0.0 & 0.0217 & 0.0 & 0.0 & 0.0 & 0.0087 \\
0.0 & 0.0 & 4.38 & 0.0 & 0.0 & -2.72 & -2.72 & 0.0 & 0.0 & 0..85 & -0.011 & 0.0 \\
0.0 & 0.0 & 0.0 & -0.567 & 0.0 & 0.0 & -3.33 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.124 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -0.0628 & 0.00374 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -0.182 & .182 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 2.2 & 0.0 & -2.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -4.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -0.067
\end{bmatrix}$$

$$
B = \begin{bmatrix}
10.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.308 & 0.0 & 0.0 \\
0.0 & 3.33 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & -0.1047 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 2.0 & -0.314 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.4 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.067
\end{bmatrix}
$$

WEIGHTING MATRICES

$$Q = \begin{bmatrix} 0.1 & 0.0 & 0.0 & 0.15 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.3 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.7 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.15 & 0.0 & 0.0 & 0.3 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.05 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.05 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.05 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.3 & 5.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 5.0 & 14.0 & 1.5 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.5 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$$R = \begin{bmatrix} 75.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 10.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 10.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 50.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.50 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 75.0 \end{bmatrix}$$

SOLUTION TO RICCATI EQUATION

$$S = \begin{bmatrix}
0.0059 & 0.0319 & 0.0875 & -0.0616 & -0.4789 & -0.0131 & 0.0093 & 0.5562 & 0.0579 & -0.0105 & 0.0030 & -0.0160 \\
0.0319 & 3.2150 & 0.3708 & -1.3714 & -23.5366 & -0.5222 & 0.4571 & 21.3077 & 1.0450 & -0.5226 & 0.1099 & -0.8802 \\
0.0875 & 0.3708 & 13.0327 & -2.7563 & -40.3534 & -0.8406 & 0.6520 & 54.1231 & 7.2441 & -0.8984 & 0.3060 & -1.3261 \\
-0.0616 & -1.3714 & -2.7563 & 2.5273 & 21.5077 & 0.7807 & -0.5758 & -24.1019 & -2.1330 & 0.4723 & -0.1205 & 0.6383 \\
-0.4789 & -23.5366 & -40.3534 & 21.5077 & 444.1153 & 8.5993 & -7.7890 & -418.0533 & -22.2091 & 10.2335 & -2.2539 & 18.4813 \\
-0.0131 & -0.5222 & -0.8406 & 0.7807 & 8.5993 & 0.3192 & -0.2584 & -9.5397 & -0.7767 & 0.1956 & -0.0465 & 0.2349 \\
0.0093 & 0.4571 & 0.6520 & -0.5758 & -7.7890 & -0.2584 & 0.2355 & 8.5470 & 0.6579 & -0.1822 & 0.0418 & -0.2108 \\
0.5562 & 21.3077 & 54.1231 & -24.1019 & -418.0533 & -9.5397 & 8.5470 & 488.8890 & 56.3123 & -8.9436 & 2.5714 & -12.4695 \\
0.0579 & 1.0450 & 7.2441 & -2.1330 & -22.2091 & -0.7767 & 0.6579 & 56.3123 & 31.8931 & 1.5866 & 0.2476 & -0.3228 \\
-0.0105 & -0.5226 & -0.8984 & 0.4723 & 10.2335 & 0.1956 & -0.1822 & -8.9436 & 1.5866 & 0.6397 & -0.0523 & 0.3411 \\
0.0030 & 0.1099 & 0.3060 & -0.1205 & -2.2539 & -0.0465 & 0.0418 & 2.5714 & 0.2476 & -0.0523 & -.5317 & -0.0707 \\
-0.0160 & -0.8802 & -1.3261 & 0.6383 & 18.4813 & 0.2349 & -0.2108 & -12.4695 & -0.3228 & 0.3411 & -0.0707 & 8.8570
\end{bmatrix}$$

CLOSED LOOP GAIN

$$K = R^{-1}B^{T}S$$

$$K = \begin{bmatrix} -0.0008 & -0.0043 & -0.0117 & 0.0082 & 0.0639 & 0.0017 & -0.0012 & -0.0742 & -0.0077 & 0.0014 & -0.0004 & 0.0021 \\ -0.0031 & -0.1522 & -0.2171 & 0.1917 & 2.5937 & 0.0861 & -0.0784 & -2.8461 & -0.2191 & 0.0607 & -0.0139 & 0.0702 \\ 0.0021 & 0.1045 & 0.1797 & -0.0945 & -2.0467 & -0.0391 & 0.0364 & 1.7887 & -0.3173 & -0.1279 & 0.0105 & -0.0682 \\ 0.0012 & 0.0446 & 0.1129 & -0.0523 & -0.8641 & -0.0207 & 0.0183 & 1.0263 & 0.1327 & -0.0159 & 0.0053 & -0.0254 \\ -0.0024 & -0.0880 & -0.2448 & 0.0964 & 1.8032 & 0.0372 & -0.0334 & -2.0571 & -0.1980 & 0.0419 & -0.4253 & 0.0565 \\ 0.0000 & 0.0008 & 0.0012 & -0.0006 & -0.0165 & -0.0002 & 0.0002 & 0.0111 & 0.0003 & -0.0003 & 0.0001 & -0.0079 \end{bmatrix}$$

CLOSED LOOP MATRIX

(A - BK)

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10.0078 | 9.9574 | -11.7166 | -108.7179 | 0.6385 | 0.0175 | -0.0124 | -0.7416 | -0.0772 | 0.0139 | -0.0040 | 0.0213 |
| 0.0767 | -0.0767 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 0.0715 | 0.0000 | -0.1250 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 0.0000 | 0.0000 | 0.0000 | -2.2700 | 5.1700 | 1.1000 | -0.1780 | -4.6200 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 0.0000 | 0.0000 | 0.0000 | 0.0000 | -0.0833 | -0.0035 | 0.0000 | 0.0217 | 0.0000 | 0.0000 | 0.0000 | 0.0087 |
| 0.0004 | 0.0137 | 4.4148 | -0.0161 | -0.2661 | -2.7264 | -2.7143 | 0.3161 | 0.0409 | 0.1801 | -0.0094 | -0.0078 |
| -0.0103 | -0.5069 | -0.7230 | 0.0715 | 8.6371 | 0.2866 | -3.5912 | -9.4776 | -0.7295 | 0.2021 | -0.0464 | 0.2337 |
| -0.0001 | -0.0047 | 0.1122 | 0.0055 | 0.0905 | 0.0022 | -0.0019 | -0.1075 | -0.0139 | -0.0611 | 0.0032 | 0.0027 |
| 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | -0.1820 | -.1820 | 0.0000 | 0.0000 |
| 0.0038 | 0.1950 | 0.3239 | -0.1725 | -3.8221 | -0.0717 | 0.0671 | 5.4552 | -0.6763 | -2.2509 | 0.0193 | -0.1285 |
| -0.0010 | -0.0352 | -0.0979 | 0.0385 | 0.7213 | 0.0149 | -0.0134 | -0.8228 | -0.0792 | 0.0167 | -0.5701 | 0.0226 |
| 0.0000 | 0.0001 | 0.0001 | 0.0000 | -0.0011 | 0.0000 | 0.0000 | 0.0007 | 0.0000 | 0.0000 | 0.0000 | -0.0675 |

Eigenvalues of the (A - BK) matrix

| Real Part | Complex Part |
|---|---|
| -9.9261 | 0.0000 |
| -3.7455 | +0.6057 |
| -3.7455 | -0.6057 |
| -2.0144 | 0.0000 |
| -0.9197 | +0.8277 |
| -0.9197 | -0.8277 |
| -0.5658 | 0.0000 |
| -0.2327 | 0.0648 |
| -0.2327 | -0.0648 |
| -0.0779 | 0.0000 |
| -0.0667 | 0.0000 |
| -0.0618 | 0.0000 |

# APPENDIX B

# FUEL THERMAL MODEL

# FUEL THERMAL MODEL

Temperature distribution in the cylindrical fuel rods depends upon the heat balance in discrete cylindrical elements, as indicated in Figure B-1.1. The following assumptions have been made in the fuel model:

1. The fuel distribution in the fuel element is independent of axial and angular space variables.
2. Fuel conductivity is a function of nodal average temperature but constant across each cylindrical node.
3. Fuel density and specific heat are constant across the fuel element.
4. Temperature at mid-node radius is the volumetric average temperature of the node.
5. Storage of heat in the fuel-cladding air gap is negligible.
6. Heat generation in the clad is negligible.

The boundary conditions on the temperature distribution are

1. Fuel cladding air gap

$$q_G = (KA \frac{\partial T}{\partial r})\Big|_{r=R_f} = A_g h_g (T_{fo} - T_{ci}) \qquad (B.1)$$

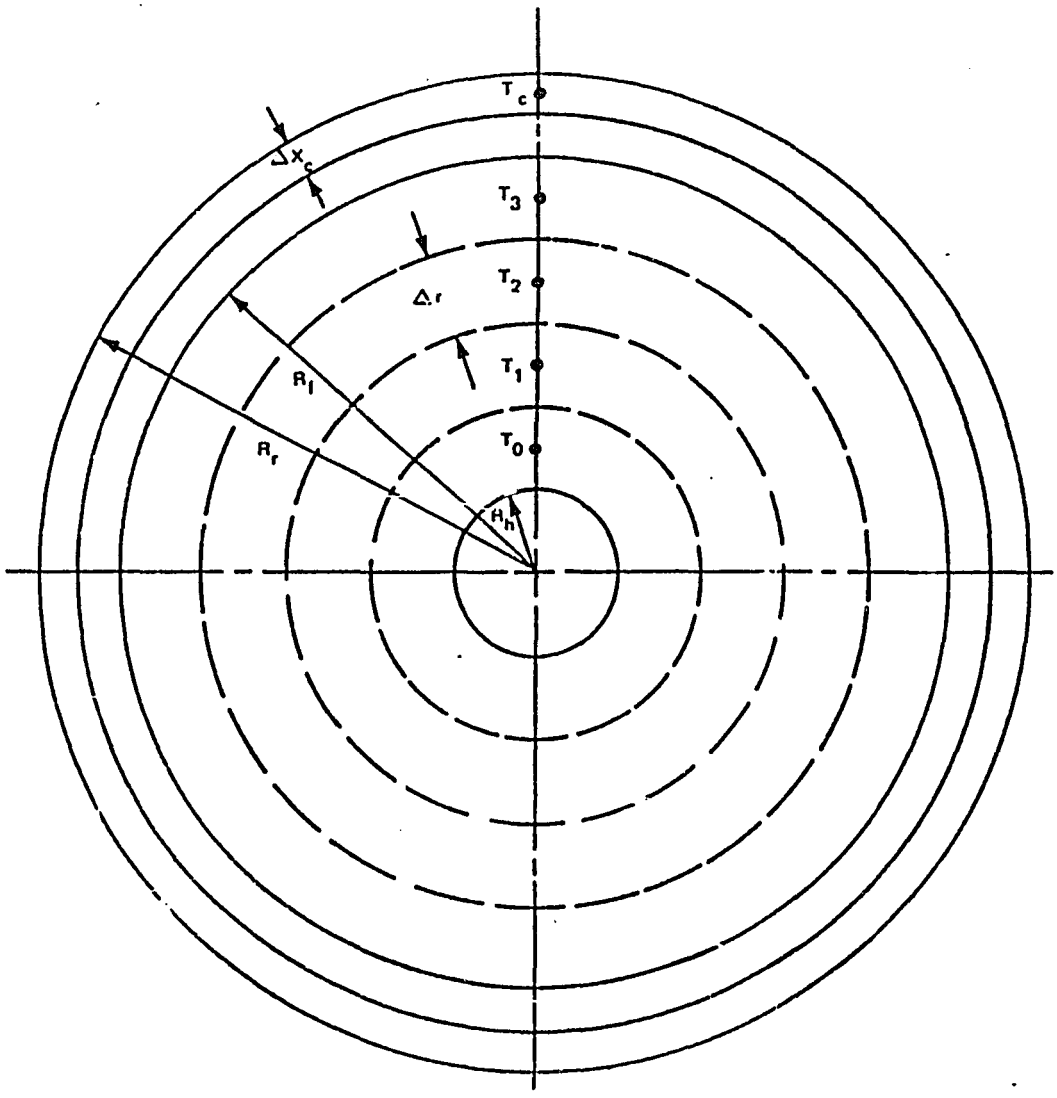



Figure B-1.1. Transient fuel node model.

2. Cladding moderator interface

$$(-K_c \frac{\partial T}{\partial r})\Big|_{r=R_r} = h_B(T_{co} - T_{Bl}) \qquad (B.2)$$

The detailed heat balance in node i of the fuel can be written in the following form:

$$\rho_f V_f C_{pf} \frac{dT_i}{dt} = (\text{Incoming Heat})_i - (\text{Outgoing Heat})_i + (\text{Heat Generated})_i) \qquad (B.3)$$

Inside the fuel elements, the heat transfer is effected by conduction; thus, the heat flow is given by Fourier's law

$$q = -KA \frac{\partial T}{\partial r} \qquad (B.4)$$

If the nodal spacing of cylinders in Figure B-1.1 is small enough, the linear heat flux is adequately approximated by the steady state relationship:

$$q = \frac{2\pi K(T_a - T_b)}{\ln(r_b/r_a)} \quad \text{for } r_b > r_a \qquad (B.5)$$

Applying Equation B.5 for the incoming and outgoing heat equation (B.3) for the various nodes gives the following equations:

$$\rho_f C_f V_{fl} \dot{T}_{fl} = 4\pi H K_{fl}(T_{f2} - T_{fl}) + \frac{Q_{fR}}{\phi_R}\phi \qquad (B.6)$$

$$\rho_f C_f V_{f2} \dot{T}_{f2} = 4\pi H K_{f1}(T_{f1} - T_{f2}) - \frac{2\pi H}{\ln(\frac{R_h + \Delta r}{R_h})} K_{f2}(T_{f2} - T_{f3}) + \frac{Q_{fR}}{\phi_R}\phi \quad (B.7)$$

$$\rho_f C_f V_{f3} \dot{T}_{f3} = \frac{2\pi H}{\ln(\frac{R_h + \Delta r}{R_h})} K_{f2}(T_{f2} - T_{f3}) - \frac{2\pi H}{\ln(\frac{R_h + 2\Delta r}{R_h + \Delta r})} K_{f3}(T_{f3} - T_{f4}) + \frac{Q_{fR}}{\phi_R}\phi \quad (B.8)$$

$$\rho_f C_f V_{f4} \dot{T}_{f4} = \frac{2\pi H}{\ln(\frac{R_h + 2\Delta r}{R_h + \Delta r})} K_{f3}(T_{f3} - T_{f4}) - A_G h_G(T_{f4} - T_c) + \frac{Q_{fR}}{\phi_R}\phi \quad (B.9)$$

$$\rho_c C_c V_c \dot{T}_c = A_G h_G(T_{f4} - T_c) - Q_f + \frac{Q_{fR}}{\phi_R}\phi \quad (B.10)$$

where

$$Q_f = \frac{2\pi H}{\ln[\frac{R_h + 3\Delta r + \Delta G + \Delta C}{R_h + 3\Delta r + \Delta G}]} K_c(T_c - T_{sat})$$

$$= A_f h_{fm}(T_c - T_{sat})$$

## B.2 Core Exit Mass Flow Equations

Mass balance equation applied to the core is given by

$$W_c - W_{ce} = \dot{m}_{fc} + \dot{m}_{gc} + \dot{m}_{sub} \tag{B.12}$$

From Equation 3.57)

$$\dot{m}_{gc} = \frac{\dot{\alpha}_c V_{cT}}{v_g} - \frac{m_{gc}\dot{v}_g}{v_g} \tag{B.13}$$

Also from Equation 3.58

$$m_{fc} = - \frac{\dot{\alpha}_c V_{cT}}{v_{fc}} - \frac{\dot{m}_{sub} V_{sub}}{v_{fc}} - \frac{m_{sub}\dot{v}_{sub}}{v_{fc}} - \frac{m_{fc}\dot{v}_{fc}}{v_{fc}} \tag{B.14}$$

Substituting for $\dot{m}_{gc}$ and $\dot{m}_{fc}$ into Equation B.12 and solving for $w_{ce}$ gives:

$$w_{ce} = w_c + \dot{\alpha}_c V_{Tc}\left[\frac{1}{v_{fc}} - \frac{1}{v_g}\right] + \frac{m_{gc}}{v_g}\dot{v}_g + \frac{m_{fc}}{v_{fc}}\dot{v}_{fc} - \left(\dot{m}_{sub} - \frac{\dot{m}_{sub} V_{sub}}{v_{fc}}\right)$$

$$= w_c + \dot{\alpha}_c V_{Tc}\frac{v_{fg}}{v_f v_g} + \dot{P}\left[m_{fc}\frac{dv_f}{dp}\frac{1}{v_f} + m_g\frac{dv_g}{dp}\frac{1}{v_g}\right] - \left(1 - \frac{v_{sub}}{v_f}\right)\dot{m}_{sub}$$

This is Equation 3.65.

## B.3 Jet Pump Momentum Exchange Balance

The inertia free momentum balance for the momentum exchange region can be written generally for one-dimensional flow in the following form:

$$\sum F = (M_{out}V_{out} - M_{in}V_{in})/g_c \tag{B.12}$$

where $V_{out}$ and $V_{in}$ are assumed to carry their own sign at this point. The ideealized geometry of the nozzle, suction and throat regions is shown in Figure 3.3.

$$M_{out}V_{out} = \frac{(1/2W_c)^2}{\bar{\rho}A_{th}} \tag{B.15}$$

$$M_{in}V_{in} = \frac{(1/2W_c - W_d)^2}{\bar{\rho}A_{suc}} + \frac{w_d^2}{\rho A_{jet}} \tag{B.16}$$

$$\sum F = - \Delta P_j A_{tn} \tag{B.17}$$

and

$$\Delta P_j = \left( \frac{(1/2W_c - W_d)^2}{A_{suc}} + \frac{W_d^2}{A_{jet}} - \frac{W_c^2}{4A_{th}} \right) / \bar{\rho} g_c A_{th} \tag{B.18}$$

# APPENDIX C

## A DARE INPUT FILE FOR THE BWR MODEL (SCHWARZBLAT ET AL., 1982)

```
****** BWR NUCLEAR POWER ******
       OPEN(UNIT=10,NAME='TEK.DAT',TYPE='NEW')
 $L
       OUTPUT WFW
10     CALL RESETI
       CALL SAVE
       CALL RUN
       IF(TMAX.GT.0.0) GOTO 10
       CONTINUE
 $M3
 $D1
    DISPLAY XN,PV
       PROCED DUMY=FALSO
       DIMENSION DUMY(1)
       ENDPRO

********SATURATED STEAM AND LIQUID PROPERTIES********
      HF= HF0 + DHFDP * (PV - PREF)
      HF1= HF0 + DHFDP * ((PV+DPV)-PREF)
      VG= VG0 + DVGDP * (PV - PREF)
      VF= VF0 + DVFDP * (PV - PREF)
      HG= HG0 + DHGDP * (PV - PREF)
      HFG= HG - HF
      VFG= VG - VF
      VGSL= VG0 + DVGDP * (PSL-PREF)
      RHOF= 1./VF
      RHOG= 1./VG
      TSAT= TSAT0 + DTSAT * (PV - PREF)
      VGRE=VG0 + DVGDP*(PRE-PREF)
      VGPR= TAVG(PR)
      HSX=(HG0+DHGDP*(PT-PREF))*1055.
*.......REACTOR KINETICS.........
       READ= -0.23 + 0.00098 * TAVER
       R1= 2.2512 + 30.3064*ALFAC - 27.2562*ALFAC**2
       R2=-24.061*ALFAC**3+91.0021*ALFAC**4-69.5349*ALFAC**5
       REAA= R1 + R2
       READD=READ-READ0
       REAAA=REAA-REAA0
       REAC= REAK - READD-REAAA
       FXN= ((REAC-1.)*XN+0.0265*Y1+0.244*Y2)/TAU0
       XN.= FXN
       FY1= 0.252*XN - 0.0265*Y1
       Y1.= FY1
       FY2= 0.748*XN - 0.244*Y2
       Y2.= FY2
**********FUEL HEAT BALANCE**********
       FT1= FAC1*(T2-T1)/FAC3 + FAC2*XN
       T1.= FT1
```

```
      FT2= (FAC1/3.)*((T1-T2)/FAC3-(T2-T3)/FAC4) + FAC2*XN
      T2.= FT2
      FT3= (FAC1/5.)*((T2-T3)/FAC4-(T3-T4)/FAC5) + FAC2*XN
      T3.= FT3
      FT4= (FAC1/7.)*((T3-T4)/FAC5-(T4-TC)/FAC6) + FAC2*XN
      T4.= FT4
      TAVER= T1/16. + 3.*T2/16. + 5.*T3/16. + 7.*T4/16.
      FTC=  FAC7*((T4-TC)/FAC6-(TC-TSAT)/FAC8)
      TC.= FTC
      XK1= COND(T1)
      XK2= COND(T2)
      XK3= COND(T3)
      XK4= COND(T4)
      FAC3= (ALOG(2.E0)/XK1) + (ALOG(1.5E0)/XK2)
      FAC4= (ALOG(4.E0/3.E0)/XK2) + (ALOG(5.E0/4.E0)/XK3)
      FAC5= (ALOG(6.E0/5.E0)/XK3) + (ALOG(7.E0/6.E0)/XK4)
         A= ALOG(8.E0/7.E0)/XK4
      FAC6= A + B + C
      QF= ((2.0*PI*(TC-TSAT))/FAC8)*XNF*XLCORE
      QT= QF + FP * QTR * XN
      BB= XMSUB * VSUB / VCT
      BB1= BB * BB
      BB2= BB1* BB
      FQ= (0.007238 - 0.08489*BB + 2.945*BB1 - 1.870*BB2)*
     $  XXK
      QSUB= FQ * QT
      QDOB= WC * HSUB
      QSAT= QT - QSUB
      QLIN= (RF**2) * PI * QWATT
**********DOWNCOMER AND CORE INLET ENTHALPY**********
      WDW= WC - WFW
      HDW= WDW*HF1/WC + WFW*HFW/WC
*      TAUDW= (VDW/VSUB)/WC
      HSUC= HDP - HDW
      FHDP= 2.0/TAUDW*(HDW-HSUC)
      HDP.= FHDP
      TAUP1= (VP1/VSUB)/WD
      HNOZX= HP1 - HSUC
      FHP1= 2.0/TAUP1*(HSUC-HNOZX)
      HP1.= FHP1
      HDIFF= (1.0-2.0*WD/WC)*HSUC +(2.0*WD/WC)*HNOZX
      TAUIP= (VIP/VSUB)/WC
      HCDOT= (HDIFF-HC)/TAUIP
      HC.= HCDOT
**********SATURATED MASS IN THE VESSEL**********
      VC= VCT - XMSUB*VSUB
```

```
      XMGC= (ALFAC*VCT)/VG
      XMFC= (VC-ALFAC*VCT)/VF
       XMGP= (ALFAP*VP)/VG
       XMFP= ((1.-ALFAP)*VP)/VF
       XMGD= (VD - XMFD*VF)/VG
       XMG= XMGD + XMGC + XMGP
       XMF= XMFD + XMFC + XMFP
        VV= VC + VP + VD
**********VESSEL PRESSURE RATE OF CHANGE*******
       A3X=XMG*DHGDP+XMF*DHFDP-(HFG/VG)*(XMG*DVGDP
     $       +XMF*DVFDP)-VV/XJ
       A3= XMG*DHGDP + XMF*DHFDP - (HFG/VFG)*(XMG*DVGDP
     $       +XMF*DVFDP) - VV/XJ
       PVDOT= (QSAT+HFG*(VF*WFW-VG*W1)/(VG-VF))/A3
         PVDOTX=(QSAT-HFG*W1)/A3
         PV.= PVDOT
**********BOILING BOUNDARY**********
        HSDOT= FSAVG * DHFDP *PVDOT+(1.0-FSAVG) * HCDOT
          HS.= HSDOT
        FSAVG= 0.330 + 0.01716*BB + 0.03811*BB1 + 0.1130*BB2
       XMSUBD=(WC*HSUB-QSUB+XMSUB*(HSDOT-(VSUB*PVDOT)/XJ))/
     $ (HF-HSUBAV)
       XMSUB.=XMSUBD
        HSUB= HF - HC
        HSUBAV= FSAVG * HSUB + HC
**********CORE EXIT MASS FLOW AND QUALITY**********

         A2= XMFC*DHFDP + XMGC*DHGDP -(HFG/VFG)*
     $          (XMFC * DVFDP +XMGC * DVGDP) - VC/XJ
         XE= (QSAT - A2*PVDOT)/(HFG*WC)
         XEDOB= WFW/WC
        WCE= WC+ (VFG/(VF*VG))*ALCDT*VCT - (1.- VF/VSUB)*
     $    XMSUBD
     $         +(XMFC*DVFDP/VF + XMGC*DVGDP/VG) * PVDOT
       ALFAEC= 1.0/(1.0+VF/VG*(1.0-XE)/XE)
        TAUPS= (XMGP+XMFP)/WCE
       ALPDOT= (ALFAEC-ALFAP)/TAUPS
       ALFAP.= ALPDOT
       WPE= WCE + (VFG/(VG*VF))*ALPDOT*VP +
     $  ((ALFAP/VG**2)*DVGDP + ((1.-ALFAP)/VF**2)*DVFDP)*
     $ PVDOT*VP
**********CORE AVERAGE VOID FRACTION**********
       ALFA1= ALFAT(HSUB,XE)
       ALFA2= ALFA1 + (CRG1+CRG2*WC/W3RTED+CRG3*(WC/W3RTED)
     $ **2)/100.
       ALFA3= ALFA2 * VG/VGO
       YSC= SWP2*ALOG(XE)+SWP3*XE+SWP1
       VIN= WC * VSUB/ACORE
```

```
      XLBOI= XLCORE - BB*XLCORE
     TAUBAR= XLBOI * YSC/VIN
        TAUSC= 0.2886 * TAUBAR
    FALCD= (ALFA3-ALFAC-2.*DAMP*TAUSC*ALCDT)/TAUSC**2
    ALCDT.= FALCD
    ALFAC.= ALCDT
**********FEEDWATER FLOW AND WATER LEVEL**********
*      FXMFD= (XMFP/(XMFP+XMGP))*WPE - WDW
       FXMFD=WFW-W1
      XMFD.= FXMFD
         HCM= XMFD*VF/AC - HCO
       FXWF= (WFSP - XWF)/TAUF
       XWF.= FXWF
**********JET PUMP PRESSURE DROP**********
       DELPJ=(VSUB/(32.2*ATH))*((WC/2.0-WD)**2/ASUC
     $        +(WD**2)/AJET-(WC**2)/(4.0*ATH))
       DELPC= (DPFORM + DPFRIC + DPGRAV)/144.
            G= WC*3600./AFLUJO
       DPFORM= XKFOR*G**2*FIF/(64.4*RHOF*1.296E+7)
         FIF= (1.-XE)**2 / (1.-ALFAC) + RHOF*XE**2/(
     $  RHOG*ALFAC)
       DPFRIC= XKFRIC*G**2*FILO/(64.4*RHOF*(3600.**2))
         FILO= FUN*(1.2*(RHOF/RHOG-1.)*XE**0.824)+1.
         FUN= 1.26-0.0004*PV+0.119*(1E+6/G)+0.00028*PV
     $ *(1E+6/G)
       DPGRAV= (RHOF*(1.-ALFAC)+RHOG*ALFAC)*XLCORE
         FWC=(32.2/A11)*(DELPJ+DELZX/VSUB-(XKSUC*VSUB)*
     $        (WC/2.0-WD)**2-(XKDIF*VSUB/4.0)*(WC**2)-D
     $  ELPC*144.0)
         WC.= FWC
         DWD=(WDSP-XWD)/TAUJ1
        XWD.=DWD
*.....................................................
       PROCED WD,DCWD=XWF,XWD,DWD,WDR,WFW
       WFW= XWF
            IF(WFW.GT. 1.15*WFW ) WFW= 1.15*WFW
            IF(WFW.LT.0.0) WFW= 0.0
       DCWD=DWD
       IF(ABS(DCWD).GT.WDRMAX)DCWD=WDRMAX*SIGN(1.0,DCWD)
       WD=XWD
       IF(WD.GT.1.20*WDR)WD=1.20*WDR
       IF(WD.LT.1000.0)WD=1000.0
       ENDPRO
* ....................................................
* NORMAIZE PRESSURE AND CALCULATE SENSER INPUTS*       *
*.....................................................
```

```
        PTNOR= PT/PTRA
        PBNOR= PB/PBR
         FPTS= (PTNOR-PTS)/TAUTS
         PTS.= FPTS
         FPBS= (PBNOR-PBS)/TAUTS
         PBS.= FPBS
*......STEAM LINE STEAM FLOW********
         W1= XKSL * XMW1
        XMW1= SQRT(ABS(DPW1)) * SIGN(1.0,DPW1)
        DPW1= (PV-PSL)/VG
        PROCED PT,WSL,VGPB,WCT,WBT,WRE=VGO,PREF,DVGDP,PSL,PB
     $ ,AT,AB,
     $       A6,A7,AREAT,AREAB,VGSL,XKSL,PTI,XKRE,PRE,WALIV
        IF(T.EQ.0.0) DUMY(1)=PTI
        PT= DUMY(1)
*********PERFORM A LINEAR INTERPOLATION**********
        MAXIT= 2000
        DO 15 JITER=1,MAXIT
        VGPT= VGO + (PT-PREF) * DVGDP
        DPSL= (PSL-PT)/VGSL
        XMSL= SQRT(ABS(DPSL)) * SIGN(1.0,DPSL)
        WSL= XKSL * XMSL
        VGPB= VGO + DVGDP * (PB-PREF)
        DPRE= (PSL-PRE)/VGSL
        XWRE= SQRT(ABS(DPRE)) * SIGN(1.0,DPRE)
        WRE= XKRE * XWRE
        PRATIO= PB/PT
        PBIG= PT
        VGBIG= VGPT
        FLWSGN= 1.0
        IF(PRATIO.LE.1.0) GO TO 63
        PRATIO= PT/PB
        PBIG= PB
        VGBIG= VGPB
        FLWSGN= -1.0
   63   CTV1= 0.26119 + 0.5896 * PRATIO
        IF(PRATIO.GE.0.6) GO TO 50
        PRATIO= 0.6
        CTV1= 0.7628
   50   CTV= CTV1 * AT
        RAIZX=((PBIG/VGBIG)/A6*(PRATIO)**A7*(1.0-(PRATIO)
     $ **A6))*32.174
        IF(RAIZX.LT.0.0) GO TO 35
        WCT= FLWSGN * CTV * AREAT * SQRT(ABS(RAIZX))/12.0
        WBT= 0.7628 * AREAB * AB * SQRT(ABS(PT/VGPT))/12.0
        WSLN= WCT + WBT
        ERROR= ABS((WSL-WSLN)/WSL)
*********SET CONVERGENCE CRITERIA***********
        EPS= 1.0E-4
```

```
      IF(ERROR.LE.EPS) GO TO 25
      GITER= 0.005
      DPT= GITER * (WSL-WSLN)
      PT1= PT
      PT= PT + DPT
      IF(JITER.GE.2)PT=PT1+(PT2-PT1)*(WSL-WSLN)/(WSLN2+WSL
     $ -WSLN-WSL2)
      PT2= PT1
      WSL2= WSL
      WSLN2= WSLN
   45 FORMAT(10X,'ITERATION ',I3,3X,'WSL= ',1PE16.4,3X,
     $'WSLN= ',
     $  1PE16.4,3X,'PT= ',1PE16.4,3X,'ERROR= ',1PE10.3)
   15 CONTINUE
   35 WRITE(6,55) JITER
   55 FORMAT(10X,'ERROR: SQRT OR OF A NEGATIVE NUMBER IS
     $ ENCOUNTERED '
     $         'OR CONVERGENCE CRITERIA HAS FAILED AT ',
     $         I3,' ITERATIONS')
   25 CONTINUE
      DUMY(1)=PT
      ENDPRO
      WT= CT * SQRT(PB/VGPB)
      FPSL= (W1-WSL-WRE-WALIV)/(-DVGDP*VSLT/(VGSL*VGSL))
      PSL.= FPSL
      FPB= (WCT-WT)/(-DVGDP*VBOWL/(VGPB*VGPB))
      PB.= FPB
*
**********TURBINE CONTROL AND BYPASS VALVES DYNAMICS*****
*
      ATD=(ATSP-XAT)/TAUTV
      ABD=(ABSP-XAB)/TAUBV
*      XAT.=DCAT
*      XAB.=DCAB
*************************************************************
          PROCED U1P,U2P=Z1PD,Z2PD
              IF(Z1P.GE.1..AND.Z1PD.GE.0.)THEN
       U1P=0.
       Z1P=1.
       ELSE IF(Z1P.LE.0..AND.Z1PD.LE.0.) THEN
       U1P=0.
       Z1P=0.
       ELSE
       U1P=1.
       ENDIF
            IF(Z2P.LE.0..AND.Z2PD.LE.0.)THEN
            U2P=0.
            Z2P=0.
```

```
                ELSE IF(Z2P.GE.1..AND.Z2PD.GE.0.)THEN
                U2P=0.
                Z2P=1.
                ELSE
 U2P=1.
   ENDIF
 ENDPRO
                 Z1PD=XK1P*(PV- PSET+TAU1P*PVDOT)
                 Z1PDOT=U1P*XK1P*(PV- PSET+TAU1P*PVDOT)
                 Z2PD=-XK2P*(PSETX-PV)
                 Z2PDOT=-U2P*XK2P*(PSETX-PV)
                 Z1P.=Z1PDOT
                 Z2P.=Z2PDOT
              PROCED XAT,XAB=Z1P,Z2P
                XAT=SATAM(Z1P,0.,1.)
                XAB=SATAM(Z2P,0.,1.)
              ENDPRO
        PROCED AT,AB,DCAT,DCAB=XAT,XAB,ATD,ABD,CVCMAX,CVOMAX,
     $          BVCMAX,BVOMAX
        IF(ATD)102,105,107
102     IF(ATD.LT.CVCMAX) DCAT=CVCMAX
        GO TO 105
107     IF(ATD.GT.CVOMAX) DCAT=CVOMAX
105     AT= XAT
        IF(XAT.LT.1.0) GO TO 110
        IF(ATD.LE.0.0) GO TO 110
        DCAT= 0.0
        AT= 1.0
110     IF(XAT.GT.0.0) GO TO 120
        IF(ATD.GE.0.0) GO TO 120
        DCAT= 0.0
        AT= 0.0
120     DCAB= ABD
        IF(ABD) 122,125,127
122     IF(ABD.LT.BVCMAX) DCAB= BVCMAX
        GO TO 125
127     IF(ABD.GT.BVOMAX) DCAB= BVOMAX
125     AB= XAB
        IF(XAB.LT.1.0) GO TO 130
        IF(ABD.LE.0.0) GO TO 130
        DCAB= 0.0
        AB= 1.0
130     IF(XAB.GT.0.0) GO TO 100
        IF(ABD.GE.0.0) GO TO 100
        DCAB= 0.0
```

```
      AB= 0.0
100   CONTINUE
      ENDPRO
**********TURBINE MODEL**********
        DHCX=(WT*HSX-W2*HCX)/(DC*VCX)-(HCX/DC)*DDC
        HCX.= DHCX
        DDC=(WT-W2)/VCX
        DC.= DDC
        W2= AK2*SQRT(ABS(DC*PB-D2*PR))
         X1= (H2-THF)/THFG
        D2= 1./(X1*VGPR+(1.-X1)*VFN)
        DW2II=((W2-WBHP-WREC)-W2II)/TW2
        W2II.= DW2II
        WREC=OKREC*W2
        WBHP= WBHP1+WBHP2
        WBHP1= OKBHP1*W2
        WBHP2= OKBHP2*W2
        WMS= (W2-WBHP-WREC)-W2I
        W2I= (H2-THF)*W2II/THFG
         H2= HSX-ETAHPA*(HSX-H2I)
        THF= TATHF(PR)*1055.
         TS= TSAT0+DTSAT*(PT-PREF)
        THFG= TATHFG(PR)*1055.
        DDR= (W2I-W3)/VR
        DR.= DDR
        DHR= (QR+W2I*THG-W3*HR)/(DR*VR)-HR*(W2I-W3)/(VR*DR)
        HR.= DHR
         PR= 1.2464*((HR/1055.)-833.)/((1./DR)-0.0123)
          W3= OK3*SQRT(ABS(PR*DR))
        W3N= W3-WBOMB
        WBOMB= OKBOMB*W3
         THG= TATHG(PR)*1055.
        DWPR1= (WRE+WREC-WPR1)/TR1
        WPR1.= DWPR1
        DQR= ((WRE+WREC+WPR1)/(2.*TR2)*HQR)*(TS-TR)-QR/TR2
         QR.= DQR
         TR= PR/(DR*R1X)-460
       DW3I=(W3N-WBLP-W3I)/TW3
       W3I.= DW3I
       WBLP= WBLP1+WBLP2+WBLP3+WBLP4
       WBLP1=OKBLP1*W3N
       WBLP2=OKBLP2*W3N
       WBLP3=OKBLP3*W3N
       WBLP4=OKBLP4*W3N
       DHFW1= QH1/(TH1*WFW) + (HO-HFW1)/TH1
        HFW1.= DHFW1
```

```
      QH1= HHFW1*(WHP1+WBLP4)
      DWHPO= (WBLP4+WHP1-WHPO)/TH1P
      WHPO.= DWHPO
      DHFW2= QH2/(TH2*WFW) + (HFW1-HFW2)/TH2
      HFW2.= DHFW2
      QH2= HHFW2*(WHP2+WBLP3)
      WHP1.= (WBLP3+WHP2-WHP1)/TH2P
      DHFW3= QH3/(TH3*WFW) + (HFW2-HFW3)/TH3
      HFW3.= DHFW3
      QH3= HHFW3*(WHP3+WBLP2)
      WHP2.= (WBLP2+WHP3-WHP2)/TH3P
      DHFW4= QH4/(TH4*WFW) + (HFW3-HFW4)/TH4
      HFW4.= DHFW4
      QH4= HHFW4*(WHP4+WBLP1)
      WHP3.= (WBLP1+WHP4-WHP3)/TH4P
      DHFW5= QH5/(TH5*WFW) + (HFW4-HFW5)/TH5
      HFW5.= DHFW5
      QH5= HHFW5*(WHP5+WBHP2+WMS)
      WHP4.= (WHP5+WBHP2+WMS-WHP4)/TH5P
      DHFWX= QH6/(TH6*WFW) + (HFW5-HFWX)/TH6
      HFWX.= DHFWX
      HFW = HFWX/1055.
      QH6= HHFW6*(WPR1+WBHP1)
      WHP5.= (WPR1+WBHP1-WHP5)/TH6P
      DXCON= (W3I+WBT+WBOMB+WHPO+WALIV-WFW)
      XCON.=DXCON
      THPF1= OKT*(HSX-H2I)*ETAHP/OMEGA
      THP= THPF1*W2III
      W2III=0.5*(W2+W2II)
      W3II= 0.5*(W3N+W3I)
      TLP= OKT*W3II*(HR-H4I)*ETALP/OMEGA
      H2I= (1067.+0.37*(PR-200.)-0.0011*((PR-200.)**2)-
     $        0.1*(PB-1000.))*1.055E+03
      ETAHP= ETACHP*ETAHPA
      ETALP= ETACLP*ETALPA
      ETACHP=1.
      ETACLP=1.
       TORQ= THP+TLP
      POWER= TORQ*OMEGA/737562
*************************************************
**                                             **
*************************************************
 $F
      FUNCTION COND(X1)
C******THERMAL CONDUCTIVITY (BTU/SG.FT.F)******
      COND= 2.088E-4+1./(2.24813*(X1-32.))
      RETURN
      END
```

```
**********    STEAM  TABLE     **********
 $T
 ALFAT,6,14
 .01,.02,.04,.06,.08,.1,.12,.14,.16,.18,.20,.24,.28,.32
 0.,.123,.182,.272,.328,.372,.412,.445,.472,.492,.518,.5
$48,.57,.598,.62
 10.,.099,.144,.228,.288,.334,.375,.41,.448,.464,.488,.5
$08,.544,.572,.6
 20.,.092,.132,.208,.266,.314,.354,.388,.416,.442,.466,.4
$88,.522,.554,.5
$82
 40.,.087,.12,.187,.24,.284,.322,.356,.386,.412,.436,.455,
$.493,.524,.554
 60.,.083,.118,.176,.226,.268,.302,.334,.362,.388,.412,.434,
$.47,.502,.53
$2
 80.,.07,.111,.172,.216,.254,.288,.319,.344,.37,.392,.414,.45,
$.482,.516
 $T
 TAVG,5
 196.,2.333
 198.,2.310
 200.,2.288
 205.,2.234
 210.,2.183
 $T
 TAVF,5
 196.,.01836
 198.,.01838
 200.,.01839
 205.,.01842
 210.,.01844
 $T
 TATHF,5
 196.,353.55
 198.,354.46
 200.,355.36
 205.,357.58
 210.,359.77
 $T
 TATHFG,5
 196.,844.5
 198.,843.7
 200.,843.
 205.,841.1
 210.,839.2
```

```
 $T
 TATHG,5
 196.,1198.1
 198.,1198.2
 200.,1198.4
 205.,1198.7
 210.,1199.0

END
*       ***  INITAL CONDITIONS BLOCK  ***
*
          Y1= 0.252*XN/0.0265
          Y2= .748*XN/.244
          HDP=  1054.2
          HP1=  1054.2
          HC= 527.1
          XWF= 2306.3889
          WFW= 2306.3889
          XN=1.00000000000
          T1= 1406.3315
          T2= 1303.6598
          T3= 1125.9091
          T4= 893.98309
          TC= 618.63708
          HS=  0.26725833E+03
          XMSUB= 7917.6807
          ALFAP= 0.75444001
          XMFD= 94841.90625
          XWD=  2930.5557
          WC= 17083.3333
          ALCDT= 0.000000000
          ALFAC= 0.38431510
          VSLT= 2570.0
          WDR=  10.55E6/3600.
          PV=  1032.24804688
          REAK= 0.00
          HC0=15.4406733
          TAU0= 3.23E-02
          RFU= 0.205
          RF= RFU/12.
          DELR= RF/4.
          RHOFX= 643.7529
          CPF= 0.076427
          PI= ATAN(1.0)*4.0
          XNF= 27528.0
          XLCORE= 12.5
```

```
      VTF= PI * (RF**2) * XLCORE * XNF
      POTEN= 1.8303318E06*(1.-FP)
      Q= POTEN/VTF
      QWATT=Q*3600.0*2.93
      DELXC= 0.032/12.
      GAP= 0.0045/12.
      XKC= 2.04E-03
      HGAP= 0.48959
      RHOC= 409.16
      CPC= 0.06687
      HB= 0.9405
      DELSQ= DELR *  DELR
      FAC1= 2./(DELSQ*RHOFX*CPF)
      FAC2= Q/(RHOFX*CPF)
      RR= RF+DELXC+GAP
      B= 1./(HGAP*RF)
      C1= (RR-DELXC/2.)/(RR-DELXC)
      C= ALOG(C1)/XKC
      FAC7= 2./(RHOC*CPC*(2.*RR*DELXC-DELXC*DELXC))
      C2= RR/(RR-DELXC/2.)
      D= ALOG(C2)/XKC
      FAC8= D + 1./(HB*RR)
      FP= 0.0430028
      QTR= 1830331.8
      VSUB= 0.02145
      ACORE= 48.791
      VCT= ACORE * XLCORE
      HFO= 539.3
      DHFDP= 0.155
      PREF= 980.0
*     DPV= 12.7
      DPV=-0.756627
      TAUDW= 0.80
      VP1= 687.0
      VIP= 2390.0
      VFO= 0.0215
      DVFDP= .5E-05
      VP= 1335.0
      VD= 4011.0
      HGO= 1192.6
      DHGDP= -0.375E-01
      XJ= 5.4
      TSATO= 542.17
      DTSAT= 0.12325
      CRG1= 8.88
      CRG2= -17.04
      CRG3= 8.16
```

```
          W3RTED= 17083.3333
          SWP1= 0.340
          SWP2= -0.1367
          SWP3= -0.1970
          DAMP=0.866
          TAUF= 2.5
          AC= 133.665
          WFSP= 2306.3889
          ATH=1.710422667
          ASUC=1.38642
          AJET=0.324004
          A11=0.16
          DELZX=12.5
          XKSUC=3.5986E-4
          XKDIF=1.88E-5
          WDSP=2930.5557
          TAUJ1=0.25
          AFLUJO= (15.824/144.)*444.
          XKFOR1= 18822413.000
          XKFOR= XKFOR1*1E-6
          XKFRIC= 3.290577
          WD= 2930.5557
        TAVERO= T1/16. + 3.*T2/16. + 5.*T3/16. + 7.*T4/16.
        READO= -0.23 + 0.00098 * TAVERO
        REAAO= R10 + R20
        R10= 2.2512 + 30.3064*ALFAC - 27.2562*ALFAC**2
   R20=-24.061*ALFAC**3+91.0021*ALFAC**4-69.5349*ALFAC**5
        XXK=1.10413193703
        PTS= 1.0
*..... TURBINE VALVE TIME CONSTANT...........
         TAUTV= 0.1
*....TURBINE CONTROL VALVE MAXIMUM CLOSING VELOCITY.......
         CVCMAX= -5.0
*...TURBINE CONTROL VALVE MAXIMUM OPENING VELOCITY.......
         CVOMAX= 5.0
*........BYPASS VALVE TIME CONSTANT.........
         TAUBV= 0.15
*.....TURBINE BYPASS VALVE MAXIMUM CLOSING VELOCITY.......
         BVCMAX= -5.0
*.....TURBINE BYPASS VALVE MAXIMUM OPENING VELOCITY.......
         BVOMAX= 5.0
         ATSP= 1.0
         ABSP= 1.
         PSL=1001.87
```

```
*.....REFERENCE SATURATED STEAM SPEC.VOLUME (FT**3/LB)
        VGO= 0.4557
*...DERIV.OF STEAM SPECIF.VOLUME WITH PRESSURE(FT**5/LB**2)
        DVGDP= -5.175E-04
*......RATED TURBINE BOWL PRESSURE
        PBR= 948.276
        PTRA= 973.81634
        TAUTS= 0.25
*......TURBINE FLOW COEFFICIENT.......
        A6= 0.20635
        A7= 1.5873
        CT= 48.571758
*.....TURBINE ADMISSION VALVE FLOW AREA.......
        AREAT=750.79266357
*........TURBINE BYPASS TOTAL FLOW AREA.......
        AREAB= 5.343051092
        XKSL=273.9743336
        PSET=1032.24804688
        PSETX=1032.24804688
*........TURBINE BOWL VOLUME.......
        VBOWL= 483.0
       Z1P= 1.0
       Z2P= 0.0
       PB= 948.276
       PBS= 1.0
       PRE=965.
       PTI= 973.81634
       XK1P=1.E-2
       XK2P=1.
       XKRE= 12.50568426
       WALIV=15.6390
       TAU1P=0.2
   AK2= 49.737895
   HO= 69.1*1.055E+03
   H4I= 1002.4*1.055E+03
   HHFW1= 142650.61
   HHFW2= 103656.24
   HHFW3= 152412.86
   HHFW4= 185049.58
   HHFW5= 334199.27
   HHFW6= 203364.81
   HQR= 23965.474
   OKT= 778./1.055E+03
```

```
      OK3= 194.0688
      OKREC= .036400902
      OKBOMB= .020611461
      OKBHP1= .053079077
      OKBHP2= .06466813
      OKBLP1= .060847908
      OKBLP2= .052789498
      OKBLP3= .041730824
      OKBLP4= .054443966
      ETAHPA= 0.8013193
      ETALPA= 0.86
      OMEGA= 120.*3.1416
      R1X= 0.57
       TH1= 100.
       TH2= 90.
       TH3= 75.
       TH4= 60.
       TH5= 45.
       TH6= 30.
       TH1P= 70.
       TH2P= 60.
       TH3P= 60.
       TH4P= 50.
       TH5P= 40.
       TH6P= 30.
       TR1= 3.
       TR2= 4.
       TW2= 0.32
       TW3= 10.
       TCOND= 120.
        VCX= 1437.6
        VFN= 0.0184
        VR= 146259.
*        W2A= 2041.05
*        W3A= 942.5731
*        WL= 0.02
        DC= 2.12090
        DR= .35375338793
        HCX= 1258437.625
        HFW1= 135145.5000
        HFW2= 176501.5000
        HFW3= 232944.0000
        HFW4= 294767.000
        HFW5= 392460.000
```

```
      HFWX= 397.8*1055.
      HR= 1.3460920E+06
      QR= 1.326849920E+08
      W2II= 1841.2833
      W3I= 1251.0609131
      WHP0= 1006.3832
      WHP1= 920.1860
      WHP2= 854.1166
      WHP3= 770.5388
      WHP4= 674.2027
      WHP5= 308.6945
      WPR1= 193.1499
      XCON=0.0
*      ATD = 1.
*      ABD = 1.
**   PARAMETERS FOR THE FEEDWATER CONTROLLER
**
       AKF1=50.0
       BKF1=0.4*AKF1
       CKF1=1.0
       TTF1=0.2
       AKF2=-150.0
       BKF2=-AKF2
       CKF2=1.0
       CKF3=1.059513163
       TTF3=0.25
       TTF4=0.2
       AKF5=0.25
       BKF5=0.4*AKF5
       AKF6=-75.
       BKF6=-AKF6
       AKF7=-0.1
       BKF7=-AKF7
       WFREF=2306.3889
       WFSEN=2306.3889
       WSSEN=2176.8384
**
**   NOMINAL VALUES FOR TESTING THE CONTROLLER
**
**   NOT USED WHEN THE CONTROL SYSTEM IS COUPLED TO THE PLANT
```

```
**
*       WSL=2306.3889
*       WFW=2306.3889
*       HCM=0.0
**
**   PARAMETERS FOR THE PRESSURE REGULATOR
**
       TTP1=0.25
       CKP2=5.0E-4
       BKP2=0.2
       TTP2=0.5
       TTP3=5.0
       CKP3=1.0
       AKP4=0.2
       CKP4=0.8
       TTP4=1.0
       CKP5=1.2
       TTP5=5.0
       CKP6=-1.0
       TTP6=25.0
       AKP7=-0.001
       BKP7=-AKP7
       AKP8=0.3
       AKP9=0.1*AKP8
       CKP9=1.0
       TTP9=0.5
       BKBP=0.03
       TTBP1=0.5
       TTBP2=0.5
       SPDEM=0.0
       DELWT=50.
       TDEM=0.0
       TTWT=1.0
       WTREF0=2176.8389
       WTSEN=2176.8389
**
**   NOMINAL VALUES FOR TESTING THE CONTROLLER
**
**   NOT USED WHEN THE CONTROL SYSTEM IS COUPLED TO THE PLANT
**
*       PTS=1.0
*       TTRIP=5.0
*       TTAU=2.0
*       SPTT=0.0
**
**   PARAMETERS FOR THE RECIRCULATION FLOW CONTROLLER
**
```

```
      CKPR=2300.
      BKR1=0.0
      CKR1=1.0
      BKR20=0.0
      DELWD=100.
      TCON=0.0
      TTWD=1.0
      AKR3=2.6E-4
      BKR3=0.025*AKR3
      CKR3=20.0*AKR3
      TTR3=10.0
      AKR4=-100.
      BKR4=0.005
      CKR4=1.0
      AKR5=0.4
      BKR5=0.1*AKR5
      TTR51=0.5
      TTR52=0.2
      CKR5=1.0
      AKR6=-0.1
      BKR6=-AKR6
      CKR6=2.5E-5
      TTR6=0.2
      AKR7=-1.0E-4
      BKR7=-AKR7
      AKR8=1333.
      BKR8=0.05*AKR8
      CKR8=10.0*AKR8
      TTR8=5.0
      AKR9=-40.0
      BKR9=-AKR9
      WDSEN=2930.5557
      WDREF=2930.5557
**
**   NOMINAL VALUES FOR TESTING THE CONTROLLER
**
**   NOT USED WHEN THE CONTROL SYSTEM IS COUPLED TO THE PLANT
**
*       SPXN=0.0
*       XNO=1.0
*       WD=2930.5557
*       TAUJ1=0.25
**
```

```
      TMAX=3.0
      DT=1.0E-5
      DTMAX=1.0E-05
      DTMIN=1.0E-07
      EMAX=1.0E-05
      EMIN=1.0E-07
      NPOINT=11

END
LIST WT(1),XN(1),PV(1),XE(1),ALFAC(1),Z1PDOT(1),PVDOT(1)
LIST WC(1),QSAT(1),QSUB(1),HCM(1),XAT(1),XAB(1),Z2PDOT(1)
LIST REAC(1),READD(1),REAAA(1),Y1(1),Y2(1),TC(1),Z2PD(1)
LIST TAVER(1),PB(1),ATD(1),ABD(1),Z1P(1),Z2P(1),Z1PD(1)
LIST FLUX(1),XN(1)-1.0,TC(1)-618.637
TEK  FLUX(1)
PLOT FLUX(1)
PLOT XN(1)-1.0
END
```

# APPENDIX D

## A SCHUR METHOD TO SOLVE ALGEBRAIC RICCATI EQUATIONS (LAUB, 1978)

# A SCHUR METHOD TO SOLVE ALGEBRAIC RICCATI EQUATIONS

In this appendix we provide FORTRAN source listings for one possible implementation of the Schur vector approach as described by Laub (1978). Subroutines to solve both the continuous-time algebraic Riccati equation (RICCND) and the discrete-time algebraic Riccati equation (RICDSD) are given. The subroutine names are derived from the following nomenclature convention for a family of subroutines to solve Riccati and various other matrix equations.

Subroutine name: XXXYYZ

where

| | | |
|---|---|---|
| XXX = | RIC | Riccati equation |
| | LYP | Lyapunov equation |
| | SYL | Sylvester equation |
| YY = | CN | Continuous-time version |
| | DS | DS discrete-time version |
| Z = | S | Single (short) precision version |
| | D | Double (long) precision version |

Subroutine RICCND calls or further requires the following additional subroutines:

BALANC, BALBAK, DDCOMP, DSOLVE, EXCHNG, HQR3, MLINEQ, ORTHES, ORTRAN, QRSTEP, SPLIT

Subroutine RICDSD requires each of the 11 subroutines above, as well as the two additional subroutines MULWOA, MIILWOB.

All the additional subroutines required have also been listed here, with the exception of BALANCE, BALBAK, ORTHES, and ORTRAN, which are available in EISPACK (Smith et al., 1976).

Finally, we add two additional technical notes:

Note 1: A fairly reliable estimate of the condition number of $U_{11}$ with respect to inversion is returned by RICCND or RICDSD in WORK.

Note 2: The subroutine HQR3 contains a small error which can occasionally cause RICCND or RICDSD to give erroneous or misleading information. The trouble arises when ORTHES produces an upper Hessenberg form with a zero on the fist subdiagnonal. HQR3 then correctly orders the resulting RSF both above and below that zero element but not necessarily globally. In practice, this almost never happens, and it has only ever been observed for certain low-order examples with all coefficient matrices diagonal.

The error can either be ignored (a safe strategy for virtually all "real problems") or temporarily patched by the following scheme.

Let $a_{i+1,i}$ be a zero element of the upper Hessenberg matrix A (the output of ORTHES). Then before HQR3 is called, $a_{i+1,i}$ should be replaced by $\varepsilon \cdot (|a_{i,i}| + |a_{i+1,i+1}|)$ where $\varepsilon$ is the machine precision (EPS) defined by

$$\varepsilon = \min_{\delta}\{\delta : FL(1 + |\delta|) \neq 1\}$$

where

FL(·) denotes floating point operation

The source listings now follow.

```
      SUBROUTINE RICCND (NZ,NF,NG,NH,N,NN,Z,W,F,G,H,ER,EI,
     +                   WORK,SCALE,ITYPE,IPVL,IPVS)
C
C     *****PARAMETERS:
      INTEGER NZ,NF,NG,NH,N,NN,ITYPE(NN),IPVL(NN),IPVS(N)
      DOUBLE PRECISION Z(NZ,NN),W(NZ,NN),F(NF,N),G(NG,N),
     +                 H(NH,N),ER(NN),EI(NN),WORK(N),SCALE(NN)
C
C     *****LOCAL VARIABLES:
      INTEGER I,J,LOW,IGH,NLOW,NUP
      DOUBLE PRECISION DABS,DSQRT,ZNORM
C
C     SUBROUTINES CALLED:
C     BALANC,BALBAK,HQR3,MLINEQ,ORTHES,ORTRAN
C     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
C     *****PURPOSE:
C     THIS SUBROUTINE SOLVES THE CONTINUOUS-TIME
C     ALGEBRAIC MATRIX RICCATI EQUATION
C                          T
C                        F *X + X*F - X*G*X + H = 0
C     BY LAUB'S VARIANT OF THE HAMILTONIAN-EIGENVECTOR
C     APPROACH.
C     *****PARAMETER DESCRIPTION:
C     ON INPUT:
C
C     NZ,NP,NG,NH  ROW DIMENSIONS OF THE ARRARYS CONTAINING
C                     Z (AND W),F,G,AND H, RESPECTIVELY,AS
C                     DECLARED IN THE CALLING PROGRAM
C                     DIMENSION STATEMENT;
C
C        N            ORDER OF THE MATRICES F,G,H;
C
C        NN           = 2*N = ORDER OF THE INTERNALLY
C                     GENERATED MATRICES Z AND W;
C
C        F            AN N X N (REAL) MATRIX;
C
C        G,H          N X N SYMMETRIC, NONNEGATIVE DEFINITE
C                     (REAL) MATRICES.
C
C     ON OUTPUT:
C
C        H            AN N X N ARRAY CONTAINING THE UNIQUE
C                     POSITIVE (OR NONNEGATIVE) DEFINITE
C                     SOLUTION OF THE RICCATI EQUATION;
C
C        ER,EI REAL SCRATCH VECTORS OF LENGTH 2*N; ON OUT-
C              PUT (ER(I),EI(I)),I=1,N CONTAIN THE REAL AND
```

```
C                       IMAGINARY PARTS,RESPECTIVELY, OF THE N
C                       CLOSED LOOP EIGENVALUES (I.E.,THE
C                       SPECTRUM OF F- G*X);
C
C         Z,W           2*N X 2*N REAL SCRATCH ARRAYS USED FOR
C                       COMPUTATIONS INVOLVING THE HAMITONIAN
C                       MATRIX ASSOCIATED WITH THE RICCATI
C                       EQUATION;
C
C         WORK,SCALE    REAL SCRATCH VECTORS OF LENGTHS N,2*N,
C                       RESPECTIVELY; ON OUTPUT, WORK(1)
C                       CONTAINS A CONDITION NUMBER ESTIMATE
C                       FOR THE FINAL NTH ORDER LINEAR MATRIX
C                       EQUATION SOLVED;
C
C         ITYPE,IPVL,IPVS  INTEGER SCRATCH VECTORS OF LENGTHS
C                       2*N, 2*N, N, RESPECTIVELY.
C
C      ***NOTE: ALL SCRATCH ARRAYS MUST BE DECLEARED AND
C               INCLUDED IN THE CALL.***
C
C      *****ALGORITHM NOTES:
C      IT IS ASSUMED THAT G AND H ARE NONNEGATIVE DEFINITE
C      AND THAT (F,B) IS STABILIZABLE AND (C,F) IS DETECTABLE
C                T                                        T
C      WHERE B*B  = G (B OF FULL RANK = RANK(G)) AND C*C = H
C      (C OF FULL RANK = RANK(H)) IN WHICH CASE THE SOLUTION
C      (RETURNED IN THE ARRAY H) IS UNIQUE AND NONNEGATIVE
C      DEFINITE.
C
C      ::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
C      EPS IS AN INTERNALLY GENERATED MACHINE DEPENDENT
C      PARAMETER SPECIFYING THE RELATIVE PRECISION OF
C      FLOATING POINT ARITHMETIC. FOR EXAMPLE, EPS =
C      16.0D0**(-13)
C      FOR DOUBLE PRECISION ARITHMETIC ON IBM S360/S370.
C
       EPS=1.0D0
 5     EPS=0.5D0*EPS
       EPSP1=EPS+1.0D0
       IF (EPSP1.GT.1.0D0) GO TO 5
       EPS=2.0D0*EPS
C
C      SET UP HAMILTONIAN MATRIX
C
       DO 20 J=1,N
          DO 10 I=1,N
                Z(I,J)=F(I,J)
```

```
                Z(N+I,J)=-H(I,J)
                Z(I,N+J)=-G(I,J)
                Z(N+I,N+J)=-F(J,I)
10          CONTINUE
20      CONTINUE
C
C       BALANCE Z
C
        CALL BALANC (NZ,NN,Z,LOW,IGH,SCALE)
C
C       COMPUTE 1-NORM OF Z
C
        ZNORM =0.0D0
        DO 40 J=1,NN
           T=0.0D0
           DO 30 I=1,NN
                T=T+DABS(Z(I,J))
30         CONTINUE
           IF (T.GT.ZNORM) ZNORM=T
40      CONTINUE
        ALPHA=DSQRT(ZNORM)+1.0D0
C
C                                    -1
C       COMPUTE  W=(ALPHA*I + Z) * (ALPHA*I- Z),AN
C       ANALYTIC FUNCTION Z MAPPING THE LEFT HALF PLANE
C       TO THE EXTERIOR OF THE UNIT DISK. THIS PERMITS
C       DIRECT APPLICATION OF HQR3. THIS STEP MAY BE
C       REMOVED IF HQR3 IS MODIFIED APPROPRIATELY.
        DO 60 J=1,NN
           DO 50 I=1,NN
                W(I,J)=-Z(I,J)
50         CONTINUE
           W(J,J)=ALPHA+W(J,J)
           Z(J,J)=ALPHA+Z(J,J)
60      CONTINUE
        CALL MLINEQ (NZ,NZ,NN,NN,Z,W,COND,IPVL,ER)
C
C       REDUCE W TO REAL SCHUR FORM WITH EIGENVALUES OUT-
C       SIDE THE UNIT DISK IN THE UPPER LEFT NXN UPPER
C       QUASI-TRIANGULAR BLOCK
C
        NLOW=1
        NUP=NN
        CALL ORTHES (NZ,NN,NLOW,NUP,W,ER)
        CALL ORTRAN (NZ,NN,NLOW,NUP,W,ER,Z)
        DO 15 I=2,NN
             IF(W(I,I-1).EQ.0.0D0) W(I,I-1)=1.0D-14
   15 CONTINUE
```

```
      CALL HQR3 (W,Z,NN,NLOW,NUP,EPS,ER,EI,ITYPE,NZ,NZ)
C
C     COMPUTE SOLUTION OF THE RICCATI EQUATION FROM
C     THE ORTHOGONAL MATRIX NOW IN THE ARRAY Z.
C     STORE THE RESULT IN THE ARRAY H.
C
      CALL BALBAK (NZ,NN,LOW,IGH,SCALE,NN,Z)
      DO 80 J=1,N
         DO 70 I=1,N
            F(I,J)=Z(J,I)
            H(I,J)=Z(N+J,I)
70       CONTINUE
80    CONTINUE
      CALL MLINEQ (NF,NH,N,N,F,H,COND,IPVS,WORK)
      WORK(1)=COND
C
C     TRANSFORM BACK TO GET THE CLOSED LOOP SPECTRUM
C
      DO 110 I=1,N
         IF (ITYPE(I).GE.0) GO TO 90
         WRITE(6,44400) I
44400    FORMAT(1X,I4,1X,'TH EIGENVALUE NOT SUCCESS
     +   FULLY CALCULATED')
         RETURN
 90      IF (ITYPE(I).GT.0) GO TO 100
         ER(I)=ALPHA*(1.0D0-ER(I))/(1.0D0+ER(I))
         EI(I)=0.0D0
         GO TO 110
 100     IF (ITYPE(I).EQ.2) GO TO 110
         T=ALPHA/((1.0D0+ER(I))**2+EI(I)**2)
         ER(I)=(1.0D0-ER(I)**2-EI(I)**2)*T
         EI(I)=-2.0D0*EI(I)*T
         ER(I+1)=ER(I)
         EI(I+1)=-EI(I)
 110  CONTINUE
      RETURN
C
C     LAST LINE OF RICCND
C
      END
```

```
      SUBROUTINE RICDSD (NZ,NF,NG,NH,N,NN,Z,W,F,G,H,ER,EI
     +                  ,WORK,SCALE,ITYPE,IPVT)
C
C     *****PARAMETERS:
      INTEGER NZ,NF,NG,NH,N,NN,ITYPE(NN),IPVT(N)
      DOUBLE PRECISION Z(NZ,NN),W(NZ,NN),F(NF,N),G(NG,N)
     +          ,H(NH,N),ER(NN),EI(NN),WORK(N),SCALE(NN)
C
C     *****LOCAL VARIABLES:
      INTEGER I,J,K,LOW,IGH,NLOW,NUP
      DOUBLE PRECISION EPS,EPSP1,COND,CONDP1
C
C     *****SUBROUTINES CALLED:
C     BALANC,BALBAK,DDCOMP,DSOLVE,HQR3,MLINED,MULWDA,
C     MULWDB,ORTHES,ORTRAN
C
C     ::::::::::::::::::::::::::::::::::::::::::::::::::
C     *****PURPOSE:
C     THIS DOUBLE PRECISION SUBROUTINE SOLVES THE DIS-
C     CRETE-TIME ALGEBRAIC MATRIX RICCATI EQUATION
C
C        T       T                T       -1     T
C     X=F *X*F-F *X*G1*((G2+G1 *X*G1)   )*G1 *X*F+H
C
C     BY LAUB,S VARIANT OF THE HAMILTONIAN-EIGENVECTOR
C     APPROACH. THE MATRIX F IS ASSUMED TO BE NONSING-
C     ULAR  AND THE MATRICES G1 AND G2 ARE ASSUMED TO
C     BE COMBINED INTO THE SQUARE ARRAY G AS FOLLOWS:
C                              -1    T
C                   G = G1*G2  *G1
C     *****PARAMETER DESCRIPTION:
C     ON INPUT:
C        NZ,NF,NG,NH  ROW DIMENSIONS OF THE ARRAYS
C        CONTAINING Z (AND W),F,G,AND H,RESPECTIVELY,
C        AS DECLARED IN THE CALLING PROGRAM DIMENSION
C        STATEMENT;
C
C        N ORDER OF THE MATRICES F,G,H;
C        NN =2*N=ORDER OF THE INTERNALLY GENERATED
C        MATRICES Z AND W;
C
C        F A NONSINGULAR NXN (REAL) MATRIX;
C
C        G,H NXN SYMMETRIC, NONNEGATIVE DEFINITE
C        (REAL) MATRICES.
```

```
C      ON OUTPUT:
C         H     AN N X N ARRAY CONTAINING THE UNIQUE
C               POSITIVE (OR NONNEGATIVE) DEFINITE
C               SOLUTION OF THE RICCATI EQUATION;
C
C         ER,EI REAL SCRATCH VECTOR OF LENGTH 2*N;
C               ON OUTPUT (ER(I),EI(I)),I=1,N CONTAIN THE
C               REAL AND IMAGINARY PARTS,RESPECTIVELY, OF
C               THE N CLOSED LOOP EIGENVALUES (I.E., THE
C               SPECTRUM OF
C                                    T       -1     T
C                F-G1*((G2+G1 *X*G1)   )*G1 *X*F
C                      -T
C                =F-G*F   *(X-H));
C
C         Z,W     2*N X 2*N REAL SCRATCH ARRAYS USED FOR
C                 COMPUTATIONS INVOLVING THE SYMPLECTIC
C                 MATRIX ASSOCIATED WITH THE RICCATI
C                 EQUATION;
C
C         WORK,SCALE  REAL SCRATCH VECTORS OF LENGTHS N,2*N,
C                  RESPECTIVELY; ON OUTPUT, WORK(1) CONTAINS
C                  A CONDITION NUMBER ESTIMATE FOR THE FINAL
C                  NTH ORDER LINEAR MATRIX EQUATION SOLVED;
C         ITYPE,IPVT  INTEGER SCRATCH VECTORS OF LENGTHS 2*N,
C                  N,RESPECTIVELY.
C       *****NOTE: ALL SCRATCH ARRAYS MUST BE DECLARED AND
C                   INCLUDED IN THE CALL.***
C       IT IS ASSUMED THAT:
C          (1) F IS NONSINGULAR
C          (2) G AND H ARE NONNEGATIVE DEFINITE
C          (3) (F,G1) IS STABILIZABLE AND (C,F) IS DETECTABLE
C               WHERE
C                T
C               C *C =H (C OF FULL RANK=RANK(H)).
C       UNDER THESE ASSUMPTIONS THE SOLUTION (RETURNED IN THE
C       ARRAY H) IS UNIQUE AND NONNEGATIVE DEFINITE.
C
C       ::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
C       EPS IS AN INTERNALLY GENERATED MACHINE DEPENDENT
C       PARAMETER SPECIFYING THE RELATIVE PRECISION OF
C       FLOATING POINT ARITHMETIC.
C       FOR EXAMPLE,EPS=16.0D0**(-13) FO DOUBLE PRECISION
C       ARITHMETIC ON IBM S360/S370.
```

```
      EPS=1.0D0
5     EPS=0.5D0*EPS
      EPSP1=EPS+1.0D0
      IF (EPSP1.GT.1.0D0) GO TO 5
      EPS=2.0D0*EPS
C     SET UP SYMPLECTIC MATRIX Z
C
      DO 20 J=1,N
         DO 10 I=1,N
            Z(N+I,N+J)=F(J,I)
 10      CONTINUE
 20   CONTINUE
      CALL DDCOMP (NF,N,F,COND,IPVT,WORK)
      CONDP1=COND+1.0D0
      IF (CONDP1.GT.COND) GO TO 30
      WRITE (6,44400)
44400 FORMAT (' MATRIX IS SINGULAR TO WORKING PRECISION')
      RETURN
30    DO 60 J=1,N
         DO 40 I=1,N
               WORK(I)=0.0D0
40          CONTINUE
            WORK(J)=1.0D0
            CALL DSOLVE (NF,N,F,WORK,IPVT)
            DO 50 I=1,N
               Z(I,J)=WORK(I)
50              CONTINUE
60    CONTINUE
      DO 80 J=1,N
         DO 70 I=1,N
             F(I,J)=Z(I,J)
70       CONTINUE
80    CONTINUE
      CALL MULWOA (NH,NF,N,H,F,WORK)
      DO 120 J=1,N
         DO 90 I=1,N
              Z(I,N+J)=0.0D0
              Z(N+I,J)=H(I,J)
90        CONTINUE
            DO 110 K=1,N
                 DO 100 I=1,N
                   Z(I,N+J)=Z(I,N+J)+F(I,K)*G(K,J)
100              CONTINUE
110         CONTINUE
120   CONTINUE
      CALL MULWOB (NH,NG,N,H,G,WORK)
      DO 140 J=1,N
         DO 130 I=1,N
```

```
                Z(N+I,N+J)=Z(N+I,N+J)+G(I,J)
130         CONTINUE
140      CONTINUE
C
C        BALANCE Z
C
         CALL BALANC (NZ,NN,Z,LOW,IGH,SCALE)
C
C        REDUCE Z TO REAL SCHUR FORM WITH EIGENVALUES
C        OUTSIDE THE UNIT DISK IN THE UPPER LEFT N X N
C        UPPER QUASI-TRIANGULAR BLOCK
C
         NLOW=1
         NUP=NN
         CALL ORTHES (NZ,NN,NLOW,NUP,Z,ER)
         CALL ORTRAN (NZ,NN,NLOW,NUP,Z,ER,W)
         CALL HQR3 (Z,W,NN,NLOW,NUP,EPS,ER,EI,ITYPE,NZ,NZ)
C
C        COMPUTE SOLUTION OF THE RICCATI EQUATION FROM THE
C        ORTHOGONAL MATRIX NOW IN THE ARRAY W. STORE THE
C        RESULT IN THE ARRAY H.
C
         CALL BALBAK (NZ,NN,LOW,IGH,SCALE,NN,W)
         DO 160 J=1,N
            DO 150 I=1,N
               F(I,J)=W(J,I)
               H(I,J)=W(N+J,I)
150         CONTINUE
160      CONTINUE
         CALL MLINEQ (NF,NH,N,N,F,H,COND,IPVT,WORK)
         WORK(1)=COND
C
C        TRANSFORM TO GET THE CLOSED LOOP SPECTRUM
C
         DO 190 I=1,N
            IF (ITYPE(I).GE.0) GO TO 170
            WRITE (6,44410) I
44410       FORMAT (1X,I4,1X,'TH EIGENVALUE NOT SUCCESSFULLY
     +      CALCULATED')
            RETURN
170         IF (ITYPE(I).GT.0) GO TO 180
            ER(I)=1.0D0/ER(I)
            EI(I)=0.0D0
            GO TO 190
180         IF (ITYPE(I).EQ.2) GO TO 190
            T=ER(I)**2+EI(I)**2
            ER(I)=ER(I)/T
            EI(I)=EI(I)/T
            ER(I+1)=ER(I)
```

```
      ER(I+1)=-EI(I)
190   CONTINUE
      RETURN
C
C     LAST CARD OF RICDSD
C
      END
```

```
                SUBROUTINE DDCOMP (NA,N,A,COND,IPVT,WORK)
C
C       *****PARAMETERS:
        INTEGER NA,N,IPVT(N)
        DOUBLE PRECISION A(NA,N),COND,WORK(N)
C
C       *****LOCAL VARIABLES:
        INTEGER NM1,I,J,K,KP1,KB,KM1,M
        DOUBLE PRECISION EK,T,ANORM,YNORM,ZNORM
C
C       *****FUNCTIONS:
        DOUBLE PRECISION DABS
C       THIS SUBROUTINE COMPUTES AN LU-DECOMPOSITION
C       OF THE REAL N X N MATRIX A BY GAUSSIAN ELIMIN-
C       ATION WITH PARTIAL PIVOTING.
C       A CONDITION NUMBER OF A IS ESTIMATED.
C
C       *****PARAMETER DESCRIPTION:
C       ON INPUT:
C          NA  ROW DIMENSION OF THE ARRAY CONTAINING A
C          AS DECLARED IN THE CALLING PROGRAM DIMENSION
C          STATEMENT;
C
C       N  ORDER OF THE MATRIX;
C
C       A  N X N MATRIX TO BE TRIANGULARIZED.
C
C       ON OUTPUT:
C
C        A     N X N ARRAY CONTAINING AN UPPER TRIANGULAR
C              MATRIX U AND A PERMUTED VERSION OF A LOWER
C              TRIANGULAR MATRIX I-L SO THAT
C              (PERMUTATION MATRIX)*A=L*U.
C        COND  AN ESTIMATE OF THE CONDITION OF A FOR THE
C              LINEAR SYSTEM
C                     A*X=B.
C              CHANGES IN A AND B MAY CAUSE CHANGES COND
C              TIMES AS LARGE IN X.  IF COND+1.0D0=COND,
C              A IS SINGULAR TO WORKING PRECISION.  COND
C              IS SET TO 1.0D+32 IF "EXACT" SINGULARITY
C              IS DETECTED.
C
C        IPVT  PIVOT VECTOR OF LENGTH N.
C        IPVT(K)=THE INDEX OF THE K-TH PIVOT ROW.
```

```
C       IPVT(N)= (-)**(NUMBER OF INTERCHANGES).
C
C       WORK     REAL SCRATCH VECTOR OF LENGTH N.
C                ITS INPUT CONTENTS ARE IGNORED. ITS OUTPUT
C                CONTENTS ARE USUALLY UNIMPORTANT.
C       *****APPLICATIONS AND USAGE RESTRICTIONS:
C       DDCOMP CAN BE USED IN CONJUCTION WITH DSOLVE TO COMPU
C       -TE SOLUTIONS TO SYSTEMS OF LINEAR EQUATIONS.IF NEAR-
C       SINGULARITY IS DECTECTED SOLUTIONS ARE MORE RELIABLY
C       COMPUTED VIA SINGULAR VALUE DECOMPOSITION OF A.
C       DDCOMP CAN ALSO BE USED TO COMPUTE THE DETERMINATE
C       OF A. ON OUTPUT SIMPLY COMPUTE:
C          DET(A) = IPVT(N)*A(1,1)*A(2,2)*...*A(N,N).
C
C       *****ALGORITHM NOTES:
C       DDCOMP IS A DOUBLE PRECISION ADAPTATION OF THE
C       SUBROUTINE DECOMP (SEE REFERENCE (1) FOR DETAILS).
C       THIS ALGORITHM IMPLEMENTS GAUSSIAN ELIMINATION IN
C       A MODERATELY UNCONVENTIONAL MANNER TO PROVIDE
C       POTENTIAL EFFICIENCY ADVANTAGES UNDER CERTAIN
C       OPERATING SYSTEMS (SEE REFERENCE (2) FOR DETAILS).
C
C       *****REFERENCES:
C       (1)   FORSYTHE,G.E., MALCOLM,M.A.,AND MOLER,C.B.,
C             COMPUTER METHODS FOR MATHEMATICAL COMPUTATIONS,
C             PRENTICE-HALL, 1977.
C       (2)   MOLER,C.B., MATRIX COMPUTATIONS WITH FORTRAN AND
C             PAGING, COMM. ACM, 15(1972), 268-270.
C
C       ::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
      IPVT(N)=1
      IF (N.EQ.1) GO TO 80
      NM=N-1
C
C     COMPUTE 1-NORM OF A
C
      ANORM=0.0D0
      DO 10 J=1,N
         T=0.D0
         DO 5 I=1,N
            T=T+DABS(A(I,J))
 5       CONTINUE
         IF (T.GT.ANORM) ANORM=T
10    CONTINUE
```

```
C
C      GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
C
       DO 35 K=1,NM1
          KP1=K+1
C
C       FIND PIVOT
C
           M=K
           DO 15 I=KP1,N
                IF (DABS(A(I,K)).GT.DABS(A(M,K))) M=I
15         CONTINUE
           IPVT(K)=M
           IF (M.NE.K) IPVT(N)=-IPVT(N)
           T=A(M,K)
           A(M,K)=A(K,K)
           A(K,K)=T
C
C      SKIP STEP IF PIVOT IS ZERO
C
           IF (T.EQ.0.0D0) GO TO 35
C
C      COMPUTE MULTIPLIERS
C
           DO 20 I=KP1,N
               A(I,K) =-A(I,K)/T
20         CONTINUE
C
C      INTERCHANGE AND ELIMINATE BY COLUMNS
C
           DO 30 J=KP1,N
                T=A(M,J)
                A(M,J)=A(K,J)
                A(K,J)=T
                IF (T.EQ.0.0D0) GO TO 30
                DO 25 I=KP1,N
                     A(I,J)=A(I,J)+A(I,K)*T
25              CONTINUE
30         CONTINUE
35      CONTINUE
C
C       COND=(1-NORM OF A)*(AN ESTIMATE OF 1-NORM OF
C       A-INVERSE). ESTIMATE OBTAINED BY ONE STEP OF
C       INVERSE ITERATION FOR THE SMALL SINGULAR VECTOR
C                      T
C       OF EQUATIONS: A *Y = E AND A*Z =Y WHERE E IS
C       A VECTOR OF +1 OR -1 CHOSEN TO CAUSE GROWTH IN
C       Y. ESTIMATE =(1-NORM OF Z)/(1-NORM OF Y).
C
```

```
C                  T
C       SOLVE    A *Y = E
C
      DO 50 K=1,N
         T=0.0D0
         IF (K.EQ.1) GO TO 45
         KM1=K-1
         DO 40 I=1,KM1
              T=T+A(I,K)*WORK(I)
40       CONTINUE
45       EK=1.0D0
         IF (T.LT.0.0D0) EK=-1.0D0
         IF (A(K,K).EQ.0.0D0) GO TO 90
         WORK(K)=-(EK+T)/A(K,K)
50     CONTINUE
       DO 60 KB =1,NM1
          K=N-KB
          T=0.0D0
          KP1=K+1
          DO 55 I=KP1,N
               T=T+A(I,K)*WORK(K)
55        CONTINUE
          WORK(K)=T
          M=IPVT(K)
          IF (M.EQ.K) GO TO 60
          T=WORK(M)
          WORK(M)=WORK(K)
          WORK(K)=T
60     CONTINUE
       YNORM=0.0D0
       DO 65 I=1,N
          YNORM =YNORM+DABS(WORK(I))
65     CONTINUE

C      SOLVE A*Z = Y
C
       CALL DSOLVE (NA,N,A,WORK,IPVT)

       ZNORM= 0.0D0
       DO 70 I=1,N
          ZNORM=ZNORM+DABS(WORK(I))
70     CONTINUE

C       ESTIMATE CONDITION
C
       COND=ANORM*ZNORM/YNORM
       IF (COND.LT.1.0D0) COND=1.0D0
       RETURN
```

```
C       1-BY-1 CASE
C
80      COND=1.0D0
        IF (A(1,1).NE.0.0D0) RETURN
C
C       "EXACT" SINGULARITY
C
90      COND=1.0D+32
        RETURN
C
C       LAST LINE OF DDCOMP
C
        END
C
        SUBROUTINE DSOLVE (NA,N,A,B,IPVT)
C
C       *****PARAMETERS:
        INTEGER NA,N,IPVT(N)
        DOUBLE PRECISION A(NA,N),B(N)
C
C       *****LOCAL VARIAPLES:
        INTEGER KB,KM1,NM1,KP1,I,K,M
        DOUBLE PRECISION T
C
C       ::::::::::::::::::::::::::::::::::::::::::::
C
C       *****PURPOSE:
C       THIS SUBROUTINE SOLVES THE LINEAR SYSTEM
C       A*X = B BY FORWARD ELIMINATION AND BACK
C       SUBSTITUTION USING THE TRIANGULAR FACTORS OF
C       A PROVIDED BY DDCOMP.
C
C       *****PARAMETER DESCRIPTION:
C       ON INPUT:
C       NA    ROW DIMENSION OF THE ARRAY CONTAINING
C             A AS DEECLEARED IN THE CALLING PROGRAM
C             DIMENSION STATEMENT;
C
C       N     ORDER OF THE MATRIX A:
C
C       A     TRIANGULARIZED MATRIX OBTAINED FROM
C             DDCOMP;
C
C       B     RIGHT HAND SIDE VECTOR OF LENGHT N;

C       IPVT   PIVOT VECTOR OF LENGHT N OBTAINED
C             FROM DDCOMP.
C
```

```
C       ON OUTPUT:
C
C        B     SOLUTION VECTOR ,X, OF LENGTH N.
C
C       ******APPLICATIONS AND USAGE RESTRICTIONS:
C       DSOLVE SHOULD NOT BE USED IN CASE DDCOMP
C       HAS DETECTED NEAR-SINGULARITY. SINGULAR
C       VALUES ANALYSIS IS THEN MORE REALIABLE.
C
C       *****ALGORITHM NOTES:
C       DSOLVE IS A DOUBLE PRECISION ADAPTATION
C       OF THE SUBROUTINE SOLVE (SEE REFERENCE
C       (1) IN THE DDCOMP DOCUMENTATION FOR DETAILS).
C
C       ::::::::::::::::::::::::::::::::::::::::::::::::
C
C       FORWARD ELIMINATION
        IF (N.EQ.1) GO TO 50
        NM1=N-1
        DO 20 K =1,NM1
           KP1=K+1
           M=IPVT(K)
           T=B(M)
           B(M)=B(K)
           B(K)=T
           DO 10 I=KP1,N
                B(I)=B(I)+A(I,K)*T
10         CONTINUE
20      CONTINUE
C
C       BACK SUBSTITUTION
C
        DO 40 KB=1,NM1
           KM1=N-KB
           K=KM1+1
           B(K)=B(K)/A(K,K)
           T=-B(K)
           DO 30 I=1,KM1
              B(I)=B(I)+A(I,K)*T
30         CONTINUE
40      CONTINUE
50      B(1)=B(1)/A(1,1)
        RETURN
C
C       LAST LINE OF DSOLVE
C
        END
```

```
      SUBROUTINE MLINEQ (NA,NB,N,M,A,B,COND,IPVT,WORK)
C
C     *****PARAMETERS:
      INTEGER NA,NB,N,M,IPVT(N)
      DOUBLE PRECISION A(NA,N),B(NB,M),COND,WORK(N)
C
C     ******LOCAL VARIABLES:
      INTEGER I,J,KIN,KOUT
      DOUBLE PRECISION CONDP1
C
C     *****SUBROUTINES CALLED:
C     DDCOMP,DSOLVE
C
C     ::::::::::::::::::::::::::::::::::::::::::::::::
C
C     *****PURPOSE:
C     THIS SUBROUTINE SOLVES THE MATRIX LINEAR
C           EQUATION  A*X = B
C     WHERE A IS AN N X N (INVERTIBLE) MATRIX AND B IS
C     AN N X M MATRIX. SUBROUTINE DDCOMP IS CALLED ONCE
C     FOR THE LU-DECOMPOSITION OF A AND SUBROUTINE
C     DSOLVE IS CALLED M TIMES FOR FORWARD ELIMINATION
C     AND BACK SUBSTITUTION TO PRODUCE THE M COLUMNS
C     OF THE SOLUTION MATRIX X = (A-INVERSE)*B. AN
C     ESTIMATE OF THE CONDITION OF A IS RETURNED.SHOULD
C     A BE SINGULAR TO WORKING ACCURACY,A MESSAGE TO THAT
C     EFFECT IS PRODUCED.
C
C     *****PARAMETER DESCRIPTION:
C     ON INPUT:
C
C     NA,NB  ROW DIMENSION OF THE ARRAYS CONTAINING A
C            AND B,RESPECTIVELY,AS DECLARED IN THE
C            CALLING PROGRAM DIMENSION STATEMENT;
C
C     N      ORDER OF THE MATRIX A AND NUMBER OF ROWS
C            OF THE MATRIX B;
C
C     M      NUMBER OF COLUMNS OF THE MATRIX B;
C
C     A      N X N COEFFICIENT MATRIX;
C
C     B      N X M RIGHT HAND SIDE MATRIX.
C
C     ON OUTPUT:
C
C     B      SOLUTION MATRIX X = (A-INVERSE)*B;
```

```
C
C      COND     AN ESTIMATE OF THE CONDITOIN OF A;
C
C      IPVT     PIVOT VECTOR OF LENGTH N (SEE DDCOMP
C               DOCUMENTATION);
C
C      WORK     A REAL SCRATCH VECTOR OF LENGTH N.
C
C      *****APPLICATIONS AND USAGE RESTRICTIONS:
C
C      (1)THE VALUE OF COND SHOULD ALWAYS BE CHECKED BY
C         CALLING PROGRAM. SHOULD A BE NEAR-SINGULAR
C         (OR SINGULAR TO WORKING ACCURACY) THE DATA
C         SHOULD BE INVESTIGATED FOR POSSIBLE ERRORS.
C         IF THERE ARE NONE AND THE PROBLEM IS
C         APPARENTLY WELL-POSED AND/OR MEANINGFUL,
C         SINGULAR VALUE ANALYSIS IS THEN A MORE REALI-
C         ABLE SOLUTION TECHIQUE (CF. EISPACK SUBROUT-
C         INES SVD AND MINFIT).
C      (2)MLINEQ CAN BE USED TO COMPUTE THE INVERSE
C         OF A: SIMPLY SOLVE A*X=I WHERE I IS THE N X N
C         IDENTITY MATRIX.
C      (3)IF THE SOLUTION TO X*A=B  (X=B*(A-INVERSE)) IS
C         DESIRED,SIMPLY TRANSPOSE THE SOLUTION OF
C           T     T
C          A *X= B .
C      *****ALGORITHM NOTES:
C      THE CONTENTS OF A ARE MODIFIED BY THIS SUBROUTINE.
C      SHOULD THE ORIGINAL COEFFICIENTS OF A BE NEEDED
C      SUBSEQUENTLY, THE CONTENTS OF A SHOULD BE SAVED
C      PROIR TO THE CALL TO MLINEQ.
C
C      ::::::::::::::::::::::::::::::::::::::::::::::::
C
       COMMON/INOU/KIN,KOUT
       CALL DDCOMP (NA,N,A,COND,IPVT,WORK)
       CONDP1=COND+1.0D0
       IF (CONDP1.GT.COND) GO TO 100
       WRITE (KOUT,44400)
44400  FORMAT (1X,'MATRIX IS SINGULAR TO WORKING
      +PRECISION')
       RETURN
100    DO 400 J=1,M
          DO 200 I=1,N
             WORK(I)=B(I,J)
200       CONTINUE
C
C  COMPUTE (J-TH COLUMN OF X)=(A-INVERSE)*(J-TH CO-
C  LUMN OF B)
```

```
C          CALL DSOLVE (NA,N,A,WORK,IPVT)
           DO 300 I=1,N
                B(I,J)=WORK(I)
300        CONTINUE
400     CONTINUE
        RETURN
C
C       LAST LINE OF MLINEQ
C
        END
```

```
      SUBROUTINE MULWOA(NA,NB,N,A,B,WORK)
C     *****PARAMETERS:
      INTEGER NA,NB,N
      DOUBLE PRECISION A(NA,N),B(NB,N),WORK(N)
C     *****LOCAL VARIABLES:
      INTEGER I,J,K
C     *****SUBROUTINE CALLED:
C     NONE
C     ::::::::::::::::::::::::::::::::::::::::
C     *****PURPOSE:
C     THIS SUBROUTINE OVERWRITES THE ARRAY A WITH
C     THE MATRIX PRODUCT  A*B. BOTH A AND B ARE
C     N X N ARRAYS AND MUST BE DISTINCT
C     *****PARAMETR DESCRIPTION:
C     ON INPUT:
C        NA,NB    ROW DIMENSIONS OF THE ARRAYS CONTAINING
C                 A AND B,RESPECTIVELY,AS DECLARED IN THE
C                 CALLING PROGRAM DIMENSION STATEMENT;
C
C        N        ORDER OF THE MATRICES A AND B;
C
C        A        AN N X N MATRIX;
C
C        B        AN N X N MATRIX.
C
C     ON OUTPUT:
C
C        A        AN N X N ARRAY CONTAINING A*B;
C
C        WORK     A REAL SCRATCH VECTOR OF LENGTH N.
C     ::::::::::::::::::::::::::::::::::::::::::::::::::
      DO 40 I=1,N
         DO 20 J=1,N
            WORK(J)=0.0D0
            DO 10 K=1,N
               WORK(J)=WORK(J)+A(I,K)*B(K,J)
10          CONTINUE
20       CONTINUE
         DO 30 J=1,N
            A(I,J)=WORK(J)
30       CONTINUE
40    CONTINUE
      RETURN
C
C     LAST LINE OF MULWOA
C
      END
```

```
      SUBROUTINE MULWOB (NA,NB,N,A,B,WORK)
C     *****PARAMETERS:
      INTEGER NA,NB,N
      DOUBLE PRECISION A(NA,N),B(NB,N),WORK(N)
C     *****LOCAL VARIABLES:
      INTEGER I,J,K
C     *****SUBROUTINE CALLED:
C     NONE
C     ::::::::::::::::::::::::::::::::::::::::::
C     *****PURPOSE:
C     THIS SUBROUTINE OVERWRITES THE ARRAY B WITH
C     THE MATRIX PRODUCED A*B. BOTH A AND B ARE
C     N X N ARRAYS AND MUST BE DISTINCT.
C     *****PARAMETER DESCRIPTION:
C     ON INPUT:
C
C       NA,NB  ROW DIMENSIONS OF THE ARRAYS CONTAINING
C              A AND B,RESPECTIVELY,AS DECLARED IN THE
C              CALLING PROGRAM DIMENSION STATEMENT:
C
C       N      ORDER OF THE MATRICES A AND B;
C
C       A      AN N X N MATRIX;
C
C       B      AN N X N MATRIX.
C
C     ON OUTPUT:
C
C       B      AN N X N ARRAY CONTAINING A*B;
C
C       WORK   A REAL SCRATCH VECTOR OF LENGHT N.
C     ::::::::::::::::::::::::::::::::::::::::::::::
      DO 50 J=1,N
         DO 10 I=1,N
              WORK(I)=0.0D0
10       CONTINUE
         DO 30 K=1,N
              DO 20 I=1,N
                WORK(I)=WORK(I)+A(I,K)*B(K,J)
20            CONTINUE
30       CONTINUE
         DO 40 I=1,N
              B(I,J)=WORK(I)
40       CONTINUE
50    CONTINUE
      RETURN
C     LAST LINE OF MULWOB
C
      END
```

```
      SUBROUTINE  HHR3(A,V,N,NLOW,NUP,EPS,ER,EI,ITYPE,
     + NA,NV)
      INTEGER N,NA,NLOW,NUP,NV,ITYPE(N)
      DOUBLE PRECISION A(NA,N),EI(N),ER(N),EPS,V(NV,N)
C     HQR3 REDUCES THE UPPER HESSENBERG MATRIX A TO
C     QUASITRIANGULAR FORM BY UNITARY SIMILARITY
C     TRANSFORMATIONS. THE EIGENVALUES OF A,WHICH ARE
C     CONTAINED IN THE 1X1 AND 2X2 DIAGONAL BLOCKS OF
C     THE REDUCED MATRIX,ARE ORDERED IN DESCENDING
C     ORDER OF MAGITUDE ALONG THE DIAGONAL.
C     THE TRANSFORMATIONS ARE ACCUMULATED IN THE
C     ARRAY V. HQR3 REQUIRES THE SUBROUTINES EXCHNG,
C     QRSTEP, AND SPLIT. THE PARAMETERS IN THE
C     CALLING SEQUENCE ARE (STARRED PARAMETERS ARE
C     ALTERED BY THE SUBROUTINE)
C      *A       AN ARRAY THAT INITIALLY CONTAINS THE N X N
C               UPPER HESSENBERG MATRIX TO BE REDUCED. ON
C                  RETURN A CONTAINS THE REDUCED, QUASI-
C                  TRIANGULAR MATRIX.
C       *V      AN ARRAY THAT CONTAINS A MATRIX INTO WHICH
C                  THE REDUCING TRANSFORMATIONS ARE TO BE
C                  MULTIPLIED.
C          N       THE ORDER OF THE MATRICES A AND V.
C          NLOW    A(NLOW,NLOW-1) AND A(NUP,+1,NUP) ARE
C          NUP     ASSUMED TO BE ZERO, AND ONLY ROWS NLOW
C                  THROUGH NUP AND COLUMNS NLOW THROUGH
C                  NUP ARE TRANSFORMED, RESULTING IN THE
C          CALCULATION OF EIGENVALUES NLOW THROUGH NUP.
C          EPS     A CONVERGENCE CRITERION.
C      *ER       AN ARRAY THAT ON RETURN CONTAINS THE DOUBLE
C                PRECISION PARTS OF THE EIGENVALUES.
C         *EI      AN ARRAY THAT ON RETURN CONTAINS THE
C                  IMAGINARY PARTS OF THE EIGENVALUES.
C         *ITYPE    AN INTEGER ARRAY WHOSE I-TH ENTRY IS
C                     0     IF THE I-TH EIGENVALUE IS DOUBLE
C                           PRECISION,
C                     1     IF THE I-TH EIGENVALUE IS COMPLEX
C                           WITH POSITIVE IMAGINARY PART.
C                     2     IF THE I-TH EIGENVALUE IS COMPLEX
C                           WITH NEGATIVE IMAGINARY PART,
C                    -1     IF THE I-TH EIGENVALUE WAS NOT
C                           CALCULATED SUCCESSFULLY.
C          NA      THE FIRST DIMENSION OF THE ARRAY A,
C          NV      THE FIRST DIMENSION OF THE ARRAY V.
C     THE CONVERGENCE CRITERION EPS IS USED TO DETERMINE
C     WHEN A SUBDIAGONAL ELEMENT OF A IS NEGLIGIBLE.
C     SPECIFICALLY A(I+1,I) IS REGARDED AS NEGLIGIBLE
C     IF
C DABS(A(I+1),I)) .LE. EPS*(DABS(A(I,I))+DABS(A(I+1,I+1))).
```

```
C       THIS MEANS THAT THE FINAL MATRIX RETURNED BY THE
C       PROGRAM WILL BE EXACTLY SIMILAR TO A + E WHERE E IS
C      OF ORDER EPS*NORM(A), FOR ANY REASONABLY BALANCED NORM
C       SUCH AS THE ROW-SUM NORM.
C
C       INTERNAL VARIABLES
        INTEGER I, IT, L, MU, NL, NU
        DOUBLE PRECISION E1, E2, P, Q, R, S, T, W, X, Y, Z
        DOUBLE PRECISION DABS
        LOGICAL FAIL
C
C       INITIALIZE.
        DO 10 I=NLOW,NUP
          ITYPE(I)= -1
   10   CONTINUE
        T=0.0D0
C
C       MAIN LOOP. FIND AND ORDER EIGENVALUES.
        NU=NUP
   20   IF (NU.LT.NLOW) GO TO 240
        IT=0
C
C       QR LOOP. FIND NEGLIGIBLE ELEMENTS AND PERFORM
C       QR STEPS.
   30   CONTINUE
C
C       SEARCH BACK FOR NEGLIGIBLE ELEMENTS.
        L=NU
   40   CONTINUE
        IF (L.EQ.NLOW) GO TO 50
        alpha = DABS(A(L,L-1))
        beta  = DABS( A(L-1,L-1)+DABS(A(L,L)) )
        IF( alpha .LE. EPS*beta ) GO TO  50
        L=L-1
        GO TO 40
   50   CONTINUE
C
C       TEST TO SEE IF AN EIGENVALUE OR A 2X2 BLOCK
C       HAS BEEN FOUND.
        X=A(NU,NU)
        IF (L.EQ.NU) GO TO 160
        Y= A(NU-1,NU-1)
        W= A(NU,NU-1)*A(NU-1,NU)
        IF (L.EQ.NU-1) GO TO 100
C
C       TEST ITERATION COUNT. IF IT IS 30 QUIT.
C       IT IS 10 OR 20 SET UP AN AD-HOC SHIFT.
        IF (IT.EQ.30) GO TO 240
```

```
      IF (IT.NE.10 .AND. IT.NE.20) GO TO 70
C
C     AD-HOC SHIFT.
      T=T + X
      DO 60 I=NLOW,NU
        A(I,I) = A(I,I) - X
   60 CONTINUE
      S = DABS(A(NU,NU-1)) + DABS(A(NU-1,NU-2))
      X = 0.75D0*S
      Y= X
      W = -0.4375D0*S**2
   70 CONTINUE
      IT= IT + 1
C
C     LOOK FOR TWO CONSECUTIVE SMALL SUB-DIAGONAL
C     ELEMENTS.
      NL = NU - 2
   80 CONTINUE
      Z = A(NL,NL)
      R = X - Z
      S = Y - Z
      P = (R*S-W)/A(NL+1,NL) + A(NL,NL+1)
      Q = A(NL+1,NL+1) - Z - R - S
      R = A(NL+2,NL+1)
      S = DABS(P) + DABS(Q) + DABS(R)
      P = P/S
      Q = Q/S
      R = R/S
      IF (NL.EQ.L) GO TO 90
      IF (DABS(A(NL,NL-1))*(DABS(Q)+DABS(R)).LE.EPS
     + *DABS(P)*(DABS(A(NL-1,
     * NL-1))+DABS(Z)+DABS(A(NL+1,NL+1)))) GO TO 90
        NL = NL -1
        GO TO 80
   90 CONTINUE
C
C     PERFORM A QR STEP BETWEEN NL AND NU.
      CALL QRSTEP(A, V, P, Q, R, NL, NU, N, NA, NV)
      GO TO 30
C
C     2X2 BLOCK FOUND.
  100 IF (NU.NE.NLOW+1) A(NU-1,NU-2) = 0.0D0
      A(NU,NU) = A(NU,NU) + T
      A(NU-1,NU-1) = A(NU-1,NU-1) + T
      ITYPE(NU) = 0
      ITYPE(NU-1) =0
      MU = NU
C
```

```
C       LOOP TO POSITION 2X2 BLOCK.
  110   CONTINUE
        NL = MU - 1
C
C
C       ATTEMPT TO SPLIT THE BLOCK INTO TWO DOUBLE
C       PRECISION EIGENVALUES.
        CALL SPLIT(A, V, N, NL, E1, E2, NA, NV)
C
C       IF THE SPLIT WAS SUCCESSFUL, GO AND ORDER THE
C       DOUBLE PRECISION EIGENVALUES.
        IF (A(MU,MU-1).EQ.0.0D0) GO TO 170
C
C       TEST TO SEE IF THE BLOCK IS PROPERLY POSITIONED,
C       AND IF NOT EXCHANGE IT
        IF (MU.EQ.NUP) GO TO 230
        IF (MU.EQ.NU-1) GO TO 130
        IF (A(MU+2,MU+1).EQ.0.0D0) GO TO 130
C
C       THE NEXT BLOCK IS 2X2.
        IF (A(MU-1,MU-1)*A(MU,MU)-A(MU-1,MU)*A(MU,MU-1)
     + .GE.A(MU+1,
     * MU+1)*A(MU+2,MU+2)-A(MU+1,MU+2)*A(MU+2,MU+1))
     + GO TO 230
        CALL EXCHNG(A, V, N, NL, 2, 2, EPS, FAIL, NA, NV)
        IF (.NOT.FAIL) GO TO 120
        ITYPE(NL) = -1
        ITYPE(NL+1) = -1
        ITYPE(NL+2) = -1
        ITYPE(NL+3) = -1
        GO TO 240
  120   CONTINUE
        MU = MU + 2
        GO TO 150
  130   CONTINUE
C
C
C       THE NEXT BLOCK IS 1X1.
        IF (A(MU-1,MU-1)*A(MU,MU)-A(MU-1,MU)*A(MU,MU-1)
     + .GE.A(MU+1,
     * MU+1)**2) GO TO 230
        CALL EXCHNG(A, V, N, NL, 2, 1, EPS, FAIL, NA, NV)
        IF (.NOT.FAIL) GO TO 140
        ITYPE(NL) =-1
        ITYPE(NL+1) = -1
        ITYPE(NL+2) = -1
        GO TO 240
  140   CONTINUE
```

```
      MU = MU + 1
  150 CONTINUE
      GO TO 110
C
C     SINGLE EIGENVALUE FOUND.
  160 NL = 0
      A(NU,NU) = A(NU,NU) + T
      IF (NU.NE.NLOW) A(NU,NU-1) = 0.0D0
      ITYPE(NU) = 0
      MU = NU
C
C
C  LOOP TO POSITION ONE OR TWO DOUBLE PRECISION
C  EIGENVALUES.
  170 CONTINUE
C
C     POSITION THE EIGENVALUE LOCATED AT A(NL,NL).
  180 CONTINUE
      IF (MU.EQ.NUP) GO TO 220
      IF (MU.EQ.NUP-1) GO TO 200
      IF (A(MU+2,MU+1).EQ.0.0D0) GO TO 200
C
C     THE NEXT BLOCK IS 2X2.
      IF (A(MU,MU)**2.GE.A(MU+1,MU+1)*A(MU+2,MU+2)
     + -A(MU+1,MU+2)*
     * A(MU+2,MU+1)) GO TO 220
      CALL EXCHNG(A, V, N, MU, 1, 2, EPS, FAIL, NA, NV)
      IF (.NOT.FAIL) GO TO 190
      ITYPE(MU) = -1
      ITYPE(MU+1) = -1
      ITYPE(MU+2) = -1
      GO TO 240
  190 CONTINUE
      MU = MU + 2
      GO TO 210
  200 CONTINUE
C
C     THE NEXT BLOCK IS 1X1.
      IF (DABS(A(MU,MU)).GE.DABS(A(MU+1,MU+1))) GO TO 220
      CALL EXCHNG(A, V, N, MU, 1, 1, EPS, FAIL, NA, NV)
      MU = MU + 1
  210 CONTINUE
      GO TO 180
  220 CONTINUE
      MU = NL
      NL = 0
      IF (MU.NE.0) GO TO 170
C
```

```
C
C       GO BACK AND GET THE NEXT EIGENVALUE.
  230   CONTINUE
        NU = L - 1
        GO TO 20
C
C
C    ALL THE EIGENVALUES HAVE BEEN FOUND AND ORDERED.
C       COMPUTE THEIR VALUES AND ITYPE.
  240   IF (NU.LT.NLOW) GO TO 260
        DO 250 I=NLOW,NU
          A(I,I) = A(I,I) + T
  250   CONTINUE
  260   CONTINUE
        NU = NUP
  270   CONTINUE
        IF (ITYPE(NU).NE.-1) GO TO 280
        NU = NU - 1
        GO TO 310
  280   CONTINUE
        IF (NU.EQ.NLOW) GO TO 290
        IF (A(NU,NU-1).EQ.0.0D0) GO TO 290
C
C
C       2X2 BLOCK
        CALL SPLIT(A, V, N, NU-1, E1, E2, NA, NV)
        IF (A(NU,NU-1).EQ.0.0D0) GO TO 290
        ER(NU) = E1
        EI(NU-1) = E2
        ER(NU-1) = ER(NU)
        EI(NU) = -EI(NU-1)
        ITYPE(NU-1) = 1
        ITYPE(NU) = 2
        NU = NU - 2
        GO TO 300
  290   CONTINUE
C
C
C       SINGLE ROOT.
        ER(NU) = A(NU,NU)
        EI(NU) = 0.0D0
        NU = NU - 1
  300   CONTINUE
  310   CONTINUE
        IF (NU.GE.NLOW) GO TO 270
        RETURN
        END
```

```
      SUBROUTINE SPLIT(A, V, N, L, E1, E2, NA, NV)
      INTEGER L, N, NA, NV
      DOUBLE PRECISION A(NA,N), V(NV,N),E1,E2
C   GIVEN THE UPPER HESSENBERG MATRIX A WITH A 2X2 BLOCK
C   STARTING AT A(L,L), SPLIT DETERMINES IF THE
C CORRESPONDING EIGENVALUES ARE DOUBLE PRECISION OR COMPLEX.
C IF THEY ARE DOUBLE PRECISION,A ROTATION IS DETERMINED THAT
C      REDUCES THE BLOCK TO UPPER TRIANGULAR FORM WITH THE
C EIGENVALUES OF LARGEST ABSOLUTE VALUE APPEARING FIRST.THE
C  ROTATION IS ACCUMULATED IN V.THE EIGENVALUES (DOUBLE PRE-
C CISION OR COMPLEX) ARE RETURNED IN E1 AND E2.THE PARAME-
C TERS IN THE CALLING SQUENCE ARE (STARRED PARAMETERS ARE
C      ALTERED BY THE SUBROUTINE)
C     *A       THE UPPER HESSENBERG MATRIX WHOSE 2X2
C              BLOCK IS TO BE SPLIT.
C     *V       THE ARRAY IN WHICH THE SPLITTING TRANS-
C              FORMATION IS TO BE ACCUMULATED.
C          N   THE ORDER OF THE MATRIX A.
C          L   THE POSITION OF THE 2X2 BLOCK.
C         *E1  ON RETURN IF THE EIGENVALUES ARE COMPLEX
C   *E2  E1 CONTAINS THEIR COMMON DOUBLE PRECISION PART AND
C              E2 CONTAINS THE POSITIVE IMAGINARY PART.
C       IF THE EIGENVALUES ARE DOUBLE PRECISION, E1 CONTAINS
C              THE ONE LARGEST IN ABSOLUTE VALUE AND E2
C              CONTAINS THE OTHER ONE.
C          NA   THE FIRST DIMENSION OF THE ARRAY A.
C          NV   THE FIRST DIMENSION OF THE ARRAY V.
C      INTERNAL VARIABLES
      INTEGER I, J, L1
      DOUBLE PRECISION DABS,DSQRT
      DOUBLE PRECISION P, Q, R, T, U, W, X, Y, Z
      X = A(L+1,L+1)
      Y = A(L,L)
      W = A(L,L+1)*A(L+1,L)
      P = (Y-X)/2.
      Q = P**2 + W
      IF (Q.GE.0.0D0) GO TO 10
C
C     COMPLEX EIGENVALUE.
      E1 = P + X
      E2 = DSQRT(-Q)
      RETURN
   10 CONTINUE
C
C
C TWO DOUBLE PRECISION EIGENVALUES.SET UP TRANSFORMATION.
      Z = DSQRT(Q)
      IF (P.LT.0.0D0) GO TO 20
```

```
      Z = P + Z
      GO TO 30
   20 CONTINUE
      Z = P - Z
   30 CONTINUE
      IF (Z.EQ.0.0D0) GO TO 40
      R = -W/Z
      GO TO 50
   40 CONTINUE
      R = 0.0D0
   50 CONTINUE
      IF (DABS(X+Z).GE.DABS(X+R)) Z = R
      Y = Y - X - Z
      X = -Z
      T = A(L,L+1)
      U = A(L+1,L)
      IF(DABS(Y)+DABS(U).LE.DABS(T)+DABS(X))GO TO 60
      Q = U
      P = Y
      GO TO 70
   60 CONTINUE
      Q = X
      P = T
   70 CONTINUE
      R = DSQRT(P**2+Q**2)
      IF (R.GT.0.0D0) GO TO 80
      E1 = A(L,L)
      E2 = A(L+1,L+1)
      A(L+1,L) = 0.0D0
      RETURN
   80 CONTINUE
      P = P/R
      Q = Q/R
C
C
C       PREMULTIPLY.
      DO 90 J=L,N
       Z = A(L,J)
      A(L,J) = P*Z+Q*A(L+1,J)
      A(L+1,J) = P*A(L+1,J) - Q*Z
   90 CONTINUE
C
C
C       POSTMULTIPLY.
      L1 = L + 1
      DO 100 I=1,L1
      Z = A(I,L)
      A(I,L) = P*Z + Q*A(I,L+1)
```

```
      A(I,L+1) = P*A(I,L+1) - Q*Z
  100 CONTINUE
C
C
C     ACCUMULATE THE TRANSFORMATION IN V.
      DO 110 I=1,N
         Z = V(I,L)
         V(I,L) = P*Z + Q*V(I,L+1)
         V(I,L+1) = P*V(I,L+1) - Q*Z
  110 CONTINUE
      A(L+1,L) = 0.0D0
      E1 = A(L,L)
      E2 = A(L+1,L+1)
      RETURN
      END


      SUBROUTINE  EXCHNG(A, V, N, L, B1, B2, EPS, FAIL,
     + NA, NV)
      INTEGER B1, B2, L, NA, NV
      DOUBLE PRECISION A(NA,N), EPS, V(NV,N)
      LOGICAL FAIL
C     GIVEN THE UPPER HESSENBERG MATRIX A WITH CONSEC-
C     UTIVE B1XB1 AND B2XB2 DIAGONAL BLOCKS (B1,B2,.LE.2)
C     STARTING AT A(L,L), EXCHNG PRODUCES A UNITARY
C     SIMILARITY TRANSFORMATION THAT EXCHANGES THE BLOCKS
C     ALONG WITH THEIR EIGENVALUES. THE TRANSFORMATION
C     IS ACCUMULATED IN V. EXCHNG PEQUIRES THE SUBROUTINE
C     QRSTEP. THE PARAMETERS IN THE CALLING SEQUENCE ARE
C     (STARRED PARAMETERS ARE ALTERED BY THE SUBROUTINE)
C       *A        THE MATRIX WHOSE BLOCKS ARE TO BE
C                   INTERCHANGED.
C       *V        THE ARRAY INTO WHICH THE TRANSFORMATIONS
C                   ARE TO BE ACCUMULATED.
C          N        THE ORDER OF THE MATRIX A.
C          L        THE POSITION OF THE BLOCKS.
C          B1       AN INTEGER CONTAINING THE SIZE OF THE
C                   FIRST BLOCK.
C          B2       AN INTEGER CONTAINIG THE SIZE OF THE
C                   BLOCK.
C          EPS      A CONVERGENCE CRITERION (cf. HQR3).
C       *FAIL     A LOGICAL VARIABLE WHICH IS FALSE ON A
C                  NORMAL RETURN. IF THIRTY ITERATIONS WERE
C                 PERFORMED WITHOUT CONVERGENCE, FAIL IS SET
C                   TO TRUE AND THE ELEMENT
C                   A(L+B2,L+B2-1) CANNOT BE ASSUMED ZERO.
C           NA      THE FIRST DIMENSION OF THE ARRAY A.
C           NV      THE FIRST DIMENSION OF THE ARRAY V.
```

```
C
C
C       INTERNAL VARIABLES.
        INTEGER I, IT, J, L1, M
        DOUBLE PRECISION DABS,DSQRT,DMAX1
        DOUBLE PRECISION P, Q, R, S, W, X, Y, Z
        FAIL = .FALSE.
        IF (B1.EQ.2) GO TO 70
        IF (B2.EQ.2) GO TO 40
C
C
C       INTERCHANGE 1X1 AND 1X1 BLOCKS.
        L1 = L + 1
        Q = A(L+1,L+1) - A(L,L)
        P = A(L,L+1)
        R = DMAX1(DABS(P),DABS(Q))
        IF (R.EQ.0.0D0) RETURN
        P = P/R
        Q = Q/R
        R = DSQRT(P**2+Q**2)
        P = P/R
        Q = Q/R
        DO 10 J=L,N
          S = P*A(L,J) + Q*A(L+1,J)
          A(L+1,J) = P*A(L+1,J) - Q*A(L,J)
          A(L,J) = S
   10   CONTINUE
        DO 20 I=1,L1
          S = P*A(I,L) + Q*A(I,L+1)
          A(I,L+1) = P*A(I,L+1) - Q*A(I,L)
          A(I,L) = S
   20   CONTINUE
        DO 30 I=1,N
          S = P*V(I,L) + Q*V(I,L+1)
          V(I,L+1) = P*V(I,L+1) - Q*V(I,L)
          V(I,L) = S
   30   CONTINUE
        A(L+1,L) = 0.0D0
        RETURN
   40   CONTINUE
C
C
C
C       INTERCHANGE 1X1 2X2 BLOCKS.
        X = A(L,L)
        P = 1.
        Q = 1.
        R = 1.
```

```
      CALL QRSTEP(A, V, P, Q, R, L, L+2, N, NA, NV)
      IT = 0
   50 IT = IT + 1
      IF (IT.LE.30) GO TO 60
      FAIL = .TRUE.
      RETURN
   60 CONTINUE
      P = A(L,L) - X
      Q = A(L+1,L)
      R = 0.0D0
      CALL QRSTEP(A, V, P, Q, R, L, L+2, N, NA, NV)
      IF (DABS(A(L+2,L+1)).GT.EPS*(DABS(A(L+1,L+1))
     + +DABS(A(L+2,L+2))))
     * GO TO 50
      A(L+2,L+1) = 0.0D0
      RETURN
   70 CONTINUE
C
C
C     INTERCHANGE 2X2 AND B2XB2 BLOCKS.
      M = L + 2
      IF (B2.EQ.2) M = M + 1
      X = A(L+1,L+1)
      Y = A(L,L)
      W = A(L+1,L)*A(L,L+1)
      P = 1.
      Q = 1.
      R = 1.
      CALL QRSTEP(A, V, P, Q, R, L, M, N, NA, NV)
      IT = 0
   80 IT = IT + 1
      IF (IT.LE.30) GO TO 90
      FAIL = .TRUE.
      RETURN
   90 CONTINUE
      Z = A(L,L)
      R = X - Z
      S = Y - Z
      P = (R*S-W)/A(L+1,L) + A(L,L+1)
      Q = A(L+1,L+1) - Z - R - S
      R =A(L+2,L+1)
      S = DABS(P) + DABS(Q) + DABS(R)
      P = P/S
      Q = Q/S
      R = R/S
      CALL QRSTEP(A, V, P, Q, R, L, M, N, NA, NV)
C
      IF (DABS(A(M-1,M-2)).GT.EPS*(DABS(A(M-1,M-1))
```

```
     + +DABS(A(M-2,M-2))))
     * GO TO 80
       A(M-1,M-2) = 0.0D0
       RETURN
       END
C
C
C
       SUBROUTINE QRSTEP(A, V, P, Q, R, NL, NU, N, NA, NV)
       INTEGER N, NA, NL, NU, NV
       DOUBLE PRECISION A(NA,N), P, Q, R, V(NV,N)
C      QRSTEP PERFORMS ONE IMPLICIT QR STEP ON THE
C      UPPER HESSENBERG MATRIX A. THE SHIFT IS DETERMINED
C      BY THE NUMBERS P, Q, AND R, AND THE STEP IS APPLIED TO
C      ROWS AND COLUMNS NL THROUGH NU. THE TRANSFORMATIONS
C      ARE ACCUMULATED IN V. THE PARAMETERS IN THE CALLING
C      SEQUENCE ARE (STARRED PARAMETERS ARE ALTERED BY THE
C      SUBROUTINE)
C         *A       THE UPPER HESSENBERG MATRIX ON WHICH THE
C                  QR STEP IS TO BE PERFORMED.
C         *V       THE ARRAY IN WHICH THE TRANSFORMATIONS
C                  ARE TO BE ACCUMULATED
C         *P       PARAMETERS THAT DETERMINE THE SHIFT.
C         *Q
C         *R
C          NL      THE LOWER LIMIT OF THE STEP.
C          NU      THE UPPER LIMIT OF THE STEP.
C          N       THE ORDER OF THE MATRIX A.
C          NA      THE FIRST DIMENSION OF THE ARRAY A.
C          NV      THE FIRST DIMENSION OF THE ARRAY V.
C
C
C      INTERNAL VARIABLES.
       INTEGER I, J, K, NL2, NL3, NUM1
       DOUBLE PRECISION S, X, Y, Z
       DOUBLE PRECISION DABS,DSQRT
       LOGICAL LAST
       NL2 = NL + 2
       DO 10 I=NL2,NU
          A(I,I-2) = 0.0D0
   10  CONTINUE
       IF (NL2.EQ.NU) GO TO 30
       NL3 = NL + 3
       DO 20 I=NL3,NU
         A(I,I-3)=0.0D0
   20  CONTINUE
   30  CONTINUE
       NUM1 = NU - 1
```

```
      DO 130 K=NL,NUM1
C
C
C     DETERMINE THE TRANSFORMATION.
      LAST = K.EQ.NUM1
      IF (K.EQ.NL) GO TO 40
      P = A(K,K-1)
      Q = A(K+1,K-1)
      R = 0.0D0
      IF (.NOT.LAST) R = A(K+2,K-1)
      X = DABS(P) + DABS(Q) + DABS(R)
      IF (X.EQ.0.0D0) GO TO 130
      P =P/X
      Q = Q/X
      R = R/X
   40 CONTINUE
      S = DSQRT(P**2+Q**2+R**2)
      IF (P.LT.0.0D0) S = -S
      IF (K.EQ.NL) GO TO 50
      A(K,K-1) = -S*X
      GO TO 60
   50 CONTINUE
      IF (NL.NE.1) A(K,K-1) = -A(K,K-1)
   60 CONTINUE
      P = P + S
      X = P/S
      Y = Q/S
      Z = R/S
      Q = Q/P
      R = R/P
C
C
C     PREMULTIPLY.
      DO 80 J=K,N
        P = A(K,J) + Q*A(K+1,J)
        IF (LAST) GO TO 70
        P = P + R*A(K+2,J)
        A(K+2,J) = A(K+2,J) - P*Z
   70 CONTINUE
      A(K+1,J) = A(K+1,J) - P*Y
      A(K,J) = A(K,J) - P*X
   80 CONTINUE
C
C
C     POSTMULTIPLY.
      J = MINO(K+3,NU)
      DO 100 I=1,J
      P = X*A(I,K) + Y*A(I,K+1)
```

```
      IF (LAST) GO TO 90
      P = P + Z*A(I,K+2)
      A(I,K+2) = A(I,K+2) - P*R
   90 CONTINUE
      A(I,K+1) = A(I,K+1) - P*Q
      A(I,K) = A(I,K) - P
  100 CONTINUE
C
C
C     ACCUMULATE THE TRANSFORMATION IN V.
      DO 120 I=1,N
        P = X*V(I,K) + Y*V(I,K+1)
        IF (LAST) GO TO 110
        P = P + Z*V(I,K+2)
        V(I,K+2) = V(I,K+2) - P*R
  110   CONTINUE
        V(I,K+1) = V(I,K+1) - P*Q
        V(I,K) = V(I,K) - P
  120   CONTINUE
  130 CONTINUE
      RETURN
      END
```

REFERENCES


Akcasu, A. Z., "Theoretical Feedback Analysis in Boiling Water Reactors," ANL-6221, 1960.

Anderson, B. D. O. and J. B. Moore, Linear Optical Control, Prentice-Hall, Inc., 1971.

Anderson, P. M. and R. P. Thompson, "Dynamic Models for Steam and Hydro Turbines in Power System Studies," IEEE Committee Report, IEEE Trans. Power Apparatus and Systems, Vol. PAS-92, No. 6, pp. 1904-1915, 1973.

Atary, J. and M. M. Shah, "Modelling and Analytical Control System Design of a Complete Power Plant Prototype," Proc. IFAC 5th World Congress, Paris, June 12-17, 1972.

Baumister, T., Standard Handbook for Mechanical Engineers, McGraw-Hill, New York, 1981.

Bjorlo, T. J., B. Blomsnes, R. Grumbach, R. Josefsson, J. E. Lunde and K. Sato, "Application of Modern Control Theory for Regulation of the Nuclear Power and the Reactor Vessel Pressure of the HBWR," HPR-131, 1971.

Bjorlo, T. J., R. Gumbach, R. Josefsson and K. O. Solberg, "Digital Control of the Halden Boiling Water Reactor by a Concept Based on Modern Control Theory," Nucl. Sci. & Engin., Vol. 39, pp. 231-240, 1970.

Bryson, A. E., Jr. and Y. Ho, Applied Optimal Control, Optimization, Estimation, and Control, Hemisphere PUblishing Corp., 1975.

Chidambara, M. R. and E. J. Davison, On a Method for Simplifying Linear Dynamic Systems, IEEE Trans. Automat. Contr. AC-12, pp. 119-121, 1967.

Chow, J. H. and P. V. Kakotovic, Eigenvalue Placement in Two Time Scale Systems, Proceedings of IFAC Symposium on Large Scale Systems, Udaine, Italy, 1976.

Davison, E. J., A New Method for Simplifying Linear Dynamic Systems, IEEE Trans. Automat. Contr. AC-13, pp. 214-215, 1968.

Davison, E. J., A Method for Simplifying Linear Dynamic Systems, IEEE Trans. Automat. Contr. AC-11, pp. 93-101, 1966.

Duderstadt, J. J. and L. J. Hamilton, Nuclear Reactor Analysis, John Wiley and Sons, Inc., New York, 1976.

El-Wakil, M. M., Nuclear Heat Transport, Internat. Text Book Company, Scranton, 1971.

Forsythe, E. E., M. A. Malcom and C. B. Moler, Computer Methods for Mathematical Computations, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1977.

Fuller, T. W., "General Electric BWR/6 Nuclear Systems--A Performance Summary," NEDO-10755, 1973.

Garbow, B. S., F. M. Boyle, J. J. Dongarra and C. B. Moler, Matrix Eigensystem Routines--EISPACK Guide Extension, Lecture Notes in Computer Science, No. 51, 1977.

Gille, J. C., M. J. Pelgrin and P. Decaulne, Feedback Control Systems, McGraw-Hill Series in Control Systems Engineering, 1959.

Godet, J. and R. Gorez, "Optimal Control of Primary Coolant Temperature in a Nuclear Plant," Automatica, Vol. 18, No. 4, pp. 373-383, 1982.

Gonzalez-Santalo, J. M. and R. T. Lahey, Jr., "An Exact Solution of Flow Decay Transients in Two-Phase Systems by the Method of Characteristics," Trans. ASME, Series C, Journal of Heat Transfer, Vol. 95, pp. 470-476, 1973.

Haberman, D., "A Minimal Sensitivity Design Method for Control Systems," Ph.D. Dissertation, University of Arizona, Tucson, 1977.

Henry, A. F., Nuclear-Reactor Analysis, the MIS Press, Cambridge, Massachusetts, 1975.

Hetrick, D. L., Dynamics of Nuclear Reactors, University of Chicago Press, Chicago, 1971.

Hetrick, D. L., "Numerical Methods for Simulation of Large Systems," EPRI Research Project Report 1381-1, Palo Alto, California, 1982.

Hetrick, D. L., P. V. Girijashankar, O. A. Palusinski, W. N. Keepin and C. D. Roten, "Solution Methods for Simulation of Nuclear Power Systems," EPRI Research Project Report 13181-1, 1980.

Hindmarsh, A. C. and G. D. Gyrne, EPISODE: An Experimental Package for Integration of Systems of Ordinary Differential Equations, University of California (Lawrence Livermore Laboratory).

Kalman, R. E. and R. W. Koepke, "Optimal Synthesis of Linear Sampling Control Systems Using Generalized Performance Indexes," Trans. ASME, Series E, Journal of Applied Mechanics, Vol. 80, 1958.

Kleinman, D. L. and M. Athans, The Design of Suboptimal Linear-Time-Varying Systems, IEEE Trans. Automat. Contro., pp. 150-159, 1968.

Korn, G. A. and J. V. Wait, Digital Continuous-System Simulation, Prentice Hall, Englewood Cliffs, New Jersey, 1978.

Kokotovic', P. V., W. R. Perkins, J. B. Cruz and G. D'Ans, " -Coupling Method for Near-Optimum Design of Large-Scale Linear Systems," Proceedings of IEEE, 116, pp. 889-892, 1969.

Kokotovic, P. V. and G. Singh, "Optimization of Coupled Non-Linear Systems, Internat. Jour. of Contr., 14, pp. 51-64, 1971.

Kwakernaak, H. and R. Siavan, Linear Optimal Control Systems, Wiley-Interscience, John Wiley & Sons, Inc., 1972.

Laub, A. J., "A Schur Method for Solving Algebraic Riccati Equations," IEEE Trans. Automat. Contr., Vol. AC-24, 1979.

Lewis, E. E., Nuclear Power Reactor Safety, Wiley-Interscience, John Wiley & Sons, Inc., 1978.

Linford,R. B., "Analytical Method of Plant Transient Evaluations for the General Electric Boiling Water Reactor," NEDO-10802, 73NED29, 1973.

Luenberger,D. G., "Observers for Multivariable Systems," IEEE Trans. Automat. Contr., Vol. AC-11, pp. 190-197, 1966.

Lunde, J. E., "Advanced Control and Automation in Nuclear Power Plants," Nuclear Engineering International, Vol. 18, p. 42, 1973.

Lunde, J. E., Overview of Current Trends in Applications of Process Control Compuers to Nuclear Power Plants, Proc. 1973 Conf., Mathematical Models and Computational Techniques for Analysis of Nuclear Systems, Ann Arbor, Michigan, April 9-11, CONF-730414-P2, 1973.

McDonald, J. P. and H. G. Kwanty, "Design and Analysis of Boiler-Turbine-Generator Controls Using Optimal Linear Regulator Theory," IEEE Tran. Antomat. Contr., Vol. AC-18, pp. 202-209, 1973.

Mendel, J. M. and D. L. Gieseking, "Bibliography on the Linear-Quadratid-Gausian Problem," IEEE Trans. Automat. Contr., Vol. AC-16, 1971.

Mohler, R. R., "Optimal Control of Nuclear Reactor Systems," Trans. Amer. Nucl. Soc., Vol. 7, 1964.

O'Malley, R. E., Jr. and L. R. Anderson, "Singular Perturbations, Order Reduction, and Decoupling of Large Scale Systems," Numerical Analysis of Singular Perturbation Problems, P. W. Hemker and J. J. H. Miller, editors, Academic Press, London, 317-338, 1979.

O'Malley, R. E., Jr. and L. R. Anderson, "Decoupling and order Reduction for Linear Time-Varying Two-Time-Scale Systems," Report to Office of Naval Research, University of Arizona, Tucson, 1980.

Payne, H. J. and L. M. Silverman, "Matrix Riccati Equation and System Structure," Proc. IEEE Conf. on Decision and Control, San Diego, California, 558-563, 1973.

Rasmussen, N., "An Assessment of Accident Risk in U. S. Commercial Nuclear Plants (WASH-1400)," NUREG-75/014, U. S. Nuclear Regulatory Commission, Washiington, D. C., 1975.

Rosztoczy, Z. R., "Optimal Xenon Shutdown Control of Nuclear Reactors," Trans. Amer. Nucl. Soc., 1967.