12-31-1991

Design of an advanced control algorithm for a nuclear power plant feedwater control system

Randall C. Ezzo
New Jersey Institute of Technology

ABSTRACT

Design of an
Advanced Control Algorithm
for a Nuclear Power Plant Feedwater Control System

by
Randall C. Ezzo

Electric power companies with nuclear power plants have been aggressively looking for ways to increase plant safety and availability by increasing the reliability and performance of the plant's control systems. The performance of the Oyster Creek Nuclear Station's feedwater control system can be significantly improved by replacing the proportional-integral compensator with a compensator based on modern control concepts. In response to a -10 inch reactor water level setpoint change, settling time is improved from 107 seconds with 3% overshoot to 34 seconds with no overshoot.

Design of an Advanced Control Algorithm For a Nuclear Power Plant Feedwater Control System

by

Randall C. Ezzo

A Thesis

Submitted to the Faculty of the Graduate Division of the

New Jersey Institute of Technology

In Partial Fulfillment of the Requirements for the Degree of

Master of Science

Department of Electrical and Computer Engineering

December 1991

APPROVAL PAGE

Design of an Advanced Control Algorithm

For a Nuclear Power Plant

Feedwater Control System

by

Randall C. Ezzo

12/20/91

Dr. Bernard Friedland, Thesis Adviser
Distinguished Professor of Electrical Engineering, NJIT

Dr. Timothy N. Chang, Committee Member
Assistant Professor of Electrical Engineering, NJIT

12/20/91

Dr. Andrew U. Meyer, Committee Member
Professor of Electrical Engineering, NJIT

12/20/91

Dr. Edwin R. Cohen, Committee Member
Professor of Electrical Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Randall C. Ezzo, PE

Degree: Master of Science

Date: December, 1991

Date of Birth:

Place of Birth:

Professional Positions: Engineer, GPU Nuclear Corp., 1982

Undergraduate and Graduate Education:

- Master of Science with major in Electrical Engineering, New Jersey Institute of Technology, Newark, NJ, 1991.
- Bachelor of Science with major in Electrical Design Engineering Technology, Pennsylvania State University, Middletown, PA 1982.
- Associate in Applied Science with major in Electrical Technology, Niagara County Community College, Sanborn, NY, 1977.

Major: Electrical Engineering

Publications: IEEE Standard 741-1990, "Standard for Protection of Power Systems in Nuclear Power Plants" working group member for 1990 revision.

Professional: NJ Professional Engineers License March, 1987

This thesis is dedicated to Faith,
my wife, for her support
and patience.

ACKNOWLEDGEMENT

The author wishes to express his sincere gratitude to his thesis advisor, Distinguished Professor Bernard Friedland, for his guidance and moral support.

Special thanks to Professors Timothy Chang, Andrew Meyer, and Edwin Cohen for serving as members of the committee.

The author is grateful to work associates, Dr. L. C. Po, for the use of his boiling water reactor plant computer model PCTRANB, E. J. Schuler for his help in power plant modeling, and T. Akos for guidance in use of personal computers.

Special thanks to C. M. Bruck for her patience and expert word processing.

Thanks to Dr. R. A. Smoak of Tennessee Technical University and T. L. Artim of Integrated Systems, Inc. for their help in system identification.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

A	plant dynamics matrix
B	control input coefficient matrix
C	observation output coefficient matrix
D	observation/control coefficient matrix
E	exogenous input (to plant) matrix
F	disturbance coefficient matrix
G	linear, quadratic gain matrix
H	Kalman filter cost evaluation function
I	identity matrix
J	linear, quadratic cost evaluation function
K	Kalman filter gain matrix
M	solution matrix of linear, quadratic algebraic Riccati equation
P	solution matrix of Kalman filter algebraic Riccati equation
Q	state performance weighting matrix
R	control performance weighting matrix
V	process noise spectral density matrix
W	observation noise spectral density matrix
e	state error vector
$\hat{e}$	state error estimate vector
$\dot{\hat{e}}$	state error derivative estimate vector
g	linear, quadratic gain
g_o	exogenous input (to plant) gain
k	Kalman filter gain
k_i	integral gain
k_l	flow to level conversion factor
k_m	flow mismatch gain
k_o	Kalman filter exogenous gain
k_p	proportional gain
k_v	level to flow conversion factor
u	control input to plant

v	process white noise vector
w	observation white noise vector
x	state vector
$\dot{x}$	state derivative vector
$\hat{x}$	state estimate vector
y	observation plant output vector
$\hat{y}$	observation estimate vector

CHAPTER 1
INTRODUCTION

First, a linear plant design model is developed with the help of system identification techniques to identify the significant plant parameters. Second, a compensator is designed by calculating the linear-quadratic gains using full state feedback. Third, a Kalman filter observer/estimator is designed to estimate the significant plant states, since many of the states are not measured. Finally, the compensator is inserted into a plant "truth" model which includes many of the non-linearities found in a boiling water nuclear power plant and a comparison is made with the proportional-integral compensator.

The feedwater control system for many boiling water reactor nuclear power plants performs the following functions: density compensation for measured variables, feedwater pump run out protection by transferring from level control to flow control at a maximum flow, automatic setpoint reduction when low sensed level and reactor trip occurs, and the ability to control both high flow (greater than 15% of rated) and low flow (less than 15% of rated) valves. Of the functions mentioned above, this thesis concentrates on only the control algorithm for the high flow regulating valves.

1.1 Plant Description

The Oyster Creek Nuclear Generating Station located in Forked River, New Jersey is the plant studied in this thesis. The nuclear systems were designed by General Electric Company and the support systems were designed by Burns and Roe Corporation. The plant's electric output is rated at approximately 640 megawatts and went into commercial operation in 1969.

Figure 1 shows a simplified diagram of the plant which is a boiling water reactor type. The steam mass flow exits the reactor vessel, flows through the steam piping, enters the turbine via the turbine control valves, and is then condensed to water in the condenser. The water flows out of the condenser, through the feedwater heaters (not shown) to the feedwater pumps, passes through the feedwater control valve, and then enters the reactor vessel annulus region via the feedwater sparger (ring inside reactor vessel which distributes feedwater evenly throughout the annulus region). A certain amount of water accumulates in the annulus and exits the annulus via the recirculation pumps (not shown). When the water leaves the recirculation pumps, the water is forced up through the reactor core, becomes a two-phase fluid, and then flows up to the steam separator. The steam separator and dryer produce a dry steam. The steam flows from the steam separator and dryer out through the reactor vessel where it starts the journey again.

The primary objective of the feedwater control system is to control the water level in the reactor vessel during all modes of plant operation. The feedwater control valves are the final control elements for the feedwater control system. If steam flow is held constant as the feedwater control system valves are opened, reactor water level will increase due to the increase in water mass with respect to the steam mass. If the feedwater control system valves are then closed, reactor water level will decrease. Since the steam mass flow of the reactor vessel is kept relatively constant by a separate control system (the turbine pressure regulator), reactor level can be adjusted by the feedwater mass flow into the reactor vessel. The turbine pressure regulator controls the flow of steam out of the reactor vessel via the turbine control valves.

There are, in actuality, two different water levels in the reactor vessel: the level in the annulus or "downcomer" and the level inside the steam dryer skirt, which is the core water level. The two levels will in general move up or down together with an offset between them that is dependent on the differential pressure across the steam dryer. This steady state offset is zero at low core flows (low power levels) increasing to several inches at high core flows (high power levels). Only one level is measured, the level in the annulus. The level is measured by a differential pressure instrument which is compensated electronically for density changes with a reactor pressure instrument.

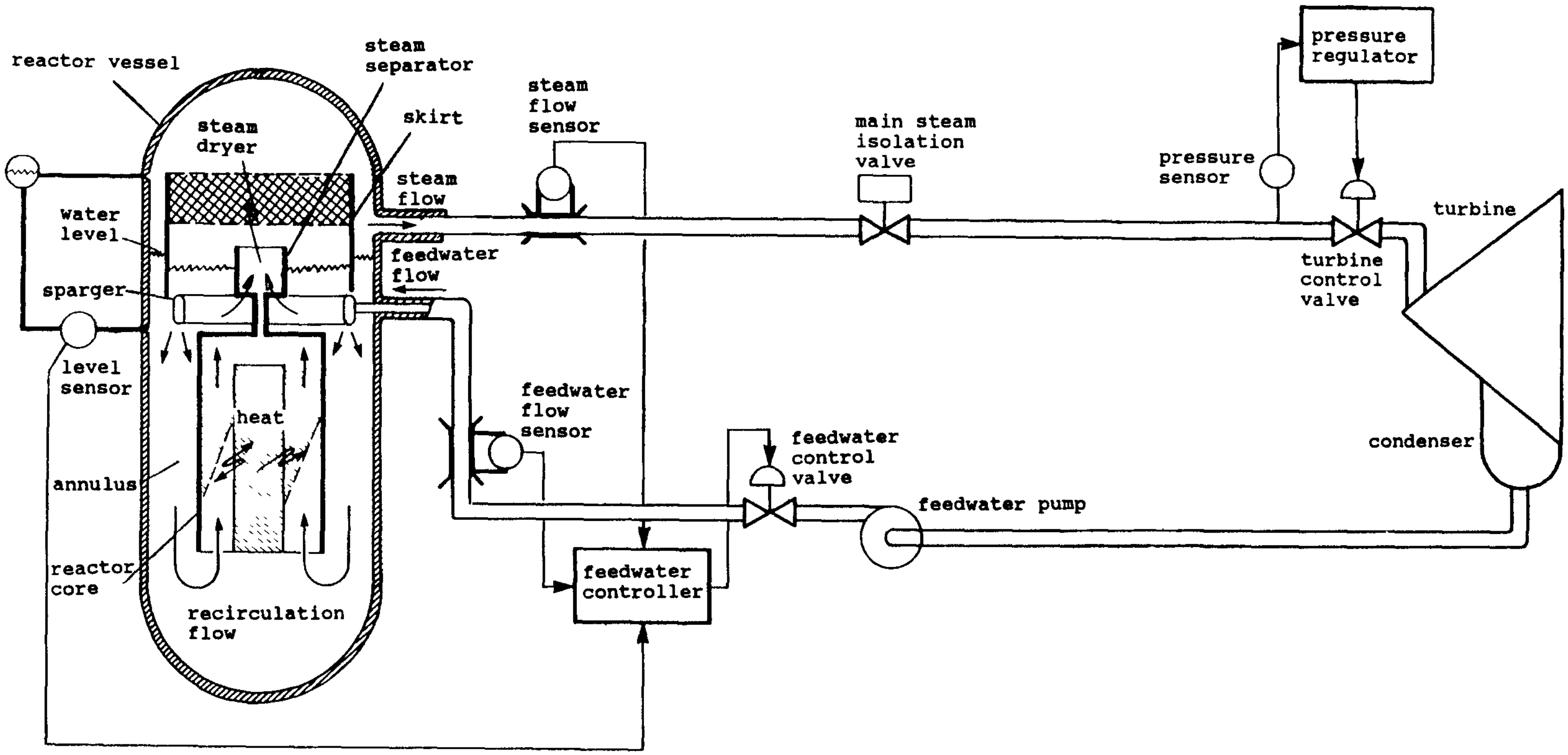
In the actual plant there are two steam lines and three feedwater lines. In this thesis the two steam lines are aggregated into one line. Also, the three feedwater lines and associated feedwater control valves are aggregated into one line and one valve.

1.2 Reactor Water Level Control

The water level in the reactor vessel is difficult to control because the water in the annulus is slightly subcooled and will shrink or swell (decrease or increase in volume) due to changes in pressure and temperature. Changes in pressure will occur with changes in steam flow (for example turbine valve closure, main steam isolation valve closure, and abrupt changes in electrical load). Changes in temperature will occur due to abrupt feedwater flow changes and reactor trips. The shrink and swell phenomenon of reactor water level has a non-minimum phase characteristic (right-half plane zeros).

The existing system requires a -24 inch automatic level setpoint change upon low sensed level and a reactor trip. This feature is necessary because post reactor trip void collapse causes a low sensed reactor water level without any change in vessel inventory (water mass). The existing

Figure 1 Simplified Diagram of a Boiling Water Reactor Plant



proportional-integral compensator responds by increasing the feedwater flow rate. However, this is an undesirable response because post reactor trip depressurization causes the voids (steam bubbles in water) to restablish. The reestablishment of the voids coupled with the initial increased feedwater demand can result in undesirably high water level and flooding of the emergency condenser lines.

Also, the required range of level for normal operation and for plant transients that do not result in a reactor trip is narrow: approximately 137 inches (reactor trip setpoint) to approximately 175 inches (turbine trip setpoint) with normal level setpoint at 160 inches from the top of active fuel. The total height of water in the annulus is about 316 inches. Therefore, the margin to turbine trip setpoint is 15 inches or about 5% of total and the margin to reactor trip is 23 inches or 7% of total. For transients that result in a reactor trip, it is a design goal to maintain level between 86 inches (emergency systems actuation setpoint) and 183 inches (emergency condenser piping).

1.3 Control Principle, Control Variable, and Observations

There are many different control structures that have been implemented for both fossil-fired and nuclear plant feedwater control systems (ANSI/ISA Standard S77.42-1987 1987; Dukelow 1979) and have been developed from years of experience by the "structuring masters" of the power industry. The existing control system uses a proportional-integral control principle. Integral control is chosen to drive the offset error to zero. The control variable is feedwater flow via feedwater valve position. The observations or measured variables are reactor water level, feedwater flow, and steam flow. The measured level is compared against the level setpoint for "single element" control. "Three element" control refers to the use of three variables for control: reactor water level, steam flow, and feedwater flow. Since steam mass flow minus feedwater mass flow is representative of the change in mass inventory in the reactor, the mass inventory term can be used to correct the control signal simultaneously with actual changes in steam flow or feedwater flow. Some plants use the mass inventory term as a feed forward signal. The existing system at Oyster Creek uses the mass inventory term as a feedback signal. By using the same control input and observations a fair comparison of the modern control compensator studied in this thesis to the proportional-integral compensator can be made.

Some plants control feedwater flow using variable speed pumps in addition to or instead of feedwater control valves. At the Oyster Creek Nuclear Station only the feedwater control valves are used to control feedwater flow.

1.4 Summary of Measured Plant Test Data

Figure 2 shows the response to a -10 inch level setpoint step input taken after tuning the feedwater control system proportional-integral parameters for 20% overshoot (May 1970). The 20% figure was chosen as a compromise between fast response and adequate stability margins. A delay of about six seconds in the level response is apparent. This delay is probably due to a delay in reactor inlet enthalpy causing a momentary increase in steam voids (Woods 1968). The momentary increase in voids temporarily offsets the decrease in water mass. As feedwater increases back to its initial state the temperature returns and the momentary increase in voids disappears. Water mass continues to drop until level settles at the new setpoint. Note that steam flow is held very steady by the turbine pressure regulator.

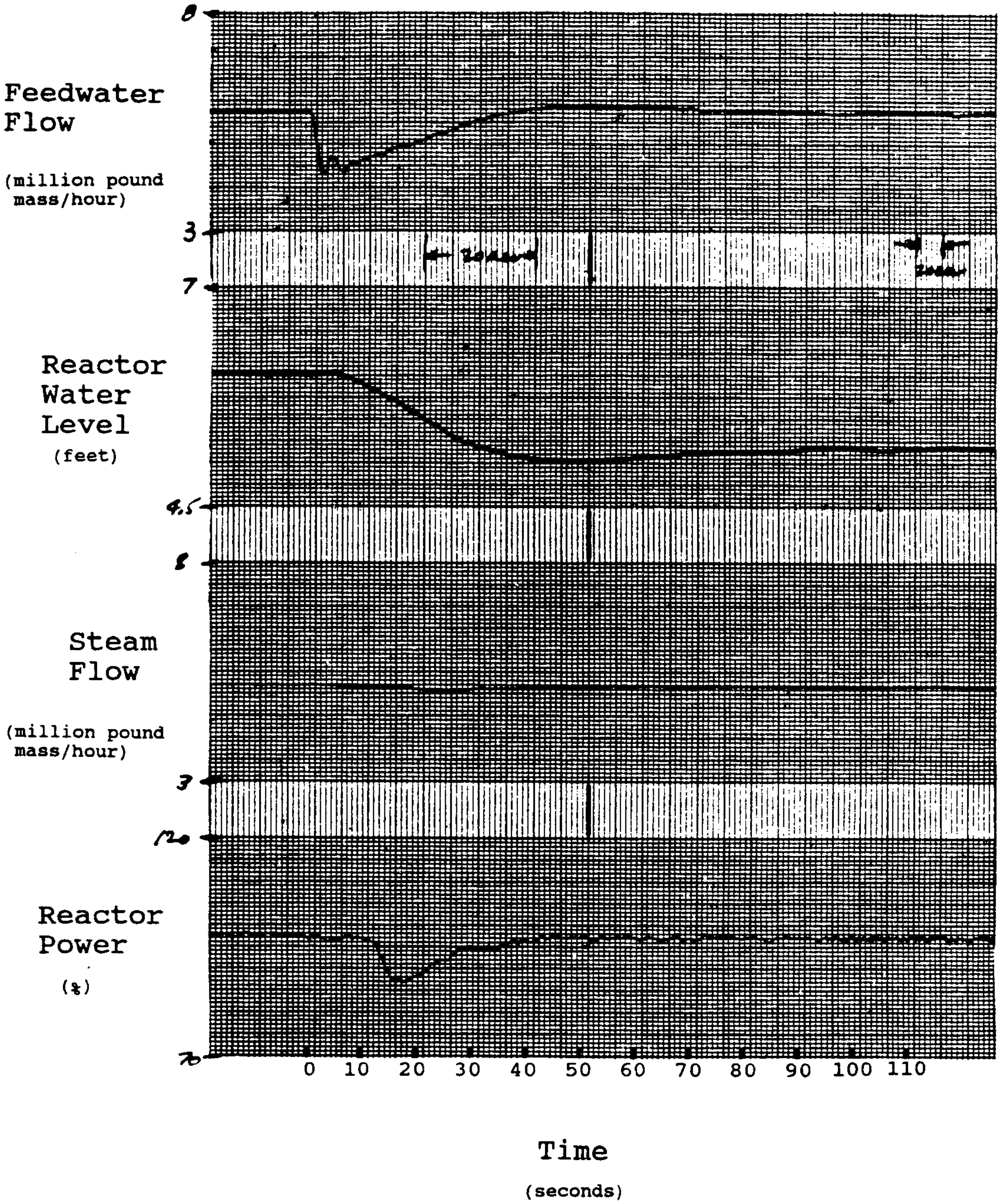
The time constant of the level response is slow-about 25 seconds compared to the time constant of feedwater flow (1.5 seconds). The settling time (time to come within ±0.2 inches of the final value) for the level response is 91 seconds. Note that reactor power dips to about 90% due to the negative void reactivity coefficient after about a 16 second delay. The momentary increase in steam voids (due to momentary decrease in feedwater) results in a momentary decrease in void reactivity and reactor power. The negative void coefficient is an inherent safety feature in United States designed plants--unlike the Chernobyl reactor design which has a positive void coefficient.

1.5 The Challenge

The challenge to the modern control design is to exceed the performance of the test data shown in Figure 2 with higher stability margins. Specifically, it is desired that the settling time (time to come within ±0.2 inches of the final value) of 91 seconds is to be improved with little or no overshoot. More importantly, the modern control compensator should perform better than the proportional-integral compensator for large plant transients (such as main steam isolation) by controlling level closer to the setpoint.

Since the modern control compensator cannot be tested on the actual plant, this thesis utilizes a computer model of the plant to compare the performance of the proportional-integral compensator with that of the modern control compensator. The challenge for modern control in this thesis is for the modern control compensator to perform better than the proportional integral controller on a computer model of the plant. Inherent in this is the challenge to the modern control design to control a complex non-linear plant.

Figure 2 Summary of Measured Test Data



1.6 Modern Control Theory

The theory and notation in this section and thesis follows the text *Control System Design* (Friedland 1986).

1.6.1 Linear, Quadratic Optimal Control The plant can be described by the vector-matrix differential equation:

$$\dot{x} = Ax + Bu \qquad (1.1)$$

where x is the plant state vector, u is the control input and A and B are known matrices.

The control law to be applied is linear:

$$u = -Gx \qquad (1.2)$$

where G is the gain matrix.

For linear, quadratic optimum control, a gain matrix is found to minimize a "cost evaluation" function J:

$$J = \int_{t}^{\infty} (x'Qx + u'Ru)dt \qquad (1.3)$$

where Q is the state weighting matrix and R is the control weighting matrix. It should be noted that Q and R are symmetric matrices and that x'Qx and u'Ru are quadratic forms. The quadratic form x'Qx represents a penalty on the deviation of the state x from the origin. The quadratic form u'Ru represents the cost of control.

Matrix M satisfies the algebraic Riccati equation:

$$0 = MA + A'M - MBR^{-1}B'M + Q \qquad (1.4)$$

The optimum gain matrix in the steady state is given by:

$$G = R^{-1}B'M \tag{1.5}$$

1.6.2 Exogenous Variables In addition to the linear, quadratic gains computed by (1.5) necessary for each state variable that is considered, it is also necessary to compute gains for external references or disturbances (exogenous variables) which are necessarily uncontrollable. The gains computed by (1.5) will control the system error for initial disturbances. The gains for the exogenous variables will control the system error for constant disturbances (x_d) and allow tracking of reference inputs (x_r). Equation (1.1) is augmented to include exogenous variables:

$$\dot{x} = Ax + Bu + Ex_o \tag{1.6}$$

where x_o is the exogenous input vector $[x_r \mid x_d]'$. The control law (1.2) is augmented to include gain for exogenous inputs G_o:

$$u = -Gx - G_o x_o \tag{1.7}$$

For zero steady state error in the output, G_o is given by:

$$G_o = \left[C(A - BG)^{-1} B\right]^{-1} C(A - BG)^{-1} E \tag{1.8}$$

1.6.3 Kalman Filter Since many of the states required by the linear, quadratic regulator are not measured, they must be estimated.

In design of the Kalman filter, we move from deterministic system analysis to statistical system analysis where random processes (white noise) are added to the process and observations. The Kalman Filter is an "optimum" estimator (or observer) provided the random processes are white and gaussian. The intent of the inclusion of noise in Kalman filter design is to make the variable estimation more realistic since random processes such as noise are always present.

A power plant is not unlike a white noise generator. Power plant noise generally has a high bandwidth compared to the bandwidth of reactor level control. Therefore, white noise appears to be a good approximation for power plant noise. Mathematically, it is desired to control the following system:

$$\dot{x} = Ax + Bu + Fv \tag{1.9}$$

$$y = Cx + w \tag{1.10}$$

where v is process white noise with spectral density matrix V and w is observation white noise with spectral density matrix W. The optimum gain matrix, K, is given by:

$$K = PC_k'W^{-1} \tag{1.11}$$

Matrix P satisfies of the matrix quadratic equation also known as an algebraic Riccati equation:

$$O = A_k P + P A_k' - P C_k' W^{-1} C_k P + V \tag{1.12}$$

where:

$$A_k = \begin{bmatrix} A & \vdots & E \\ \hline O & \vdots & A_o \end{bmatrix} \qquad B_k = \begin{bmatrix} B & \vdots & O \\ \hline O & \vdots & I \end{bmatrix} \tag{1.13}$$

where:

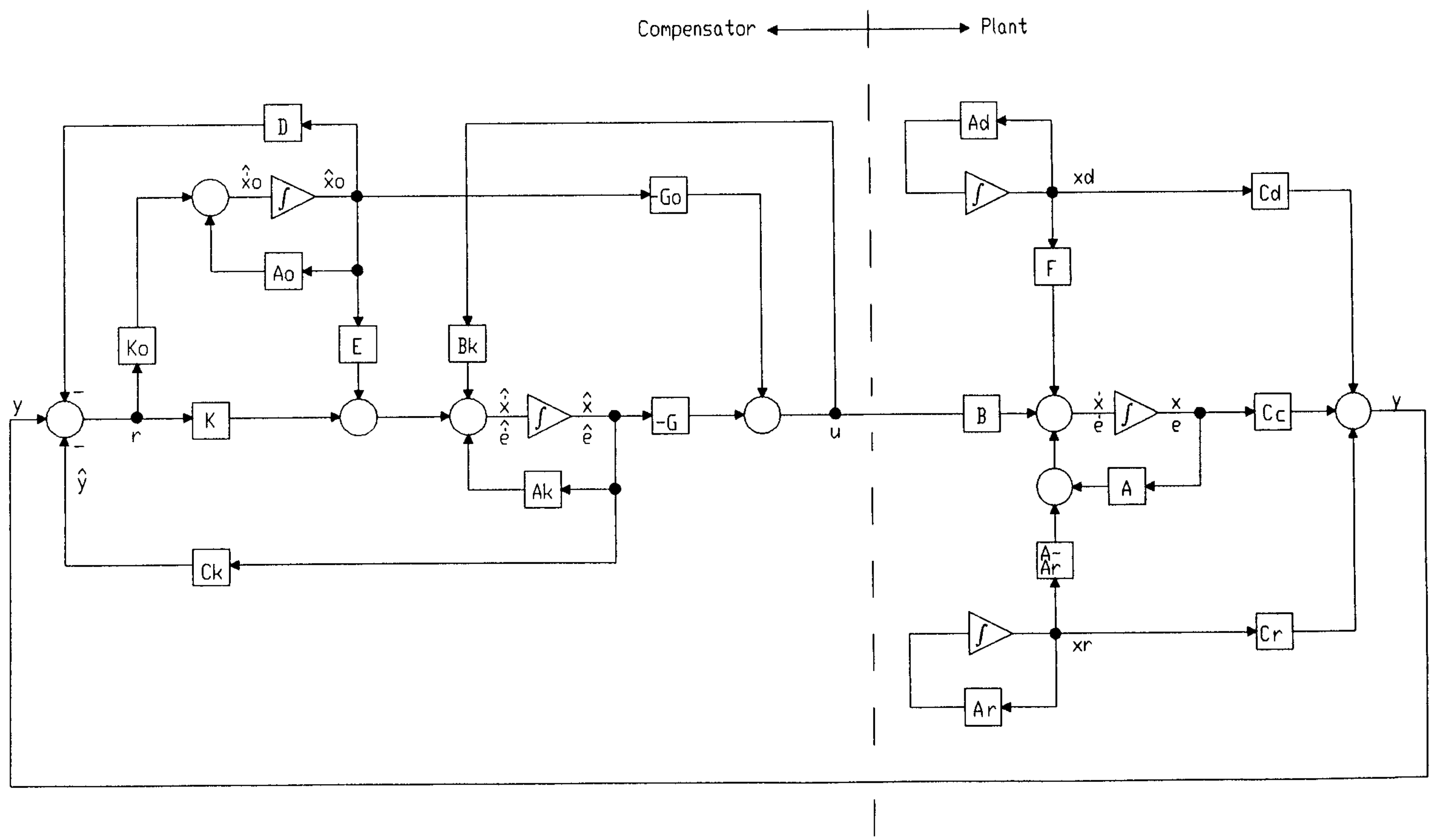
$$A_o = \begin{bmatrix} A_r & \vdots & O \\ \hline O & \vdots & A_d \end{bmatrix} \qquad I = \text{identity matrix} \tag{1.14}$$

If x_r and x_d are modeled by integrators, then A_r and A_d represent the feedback around the integrators.

1.6.4 Generalized Optimal Control Structure

Figure 3 shows the generalized structure of the compensator and plant with exogenous inputs. This structure is edited for the specific application involved (see Figure 7). In this thesis it is found that the following are not required to estimate exogenous inputs: C_d, A_d, A_r, and D and therefore these matrices are set to zero. Matrix C_d is zero because output y does not depend directly on X_d. Matrices A_d and A_r are zero as X_d and X_r are adequately modeled without A_d and A_r. Matrix D is zero as residual r does not depend directly on $\hat{x}_o$.

Figure 3 Generalized Control Structure With Exogenous Inputs



CHAPTER 2

REVIEW OF LITERATURE

2.1 Overview

Linear quadratic/Kalman filter optimal control was first studied for application to nuclear power plants in the early 1970's (Karpeta, Stirsky, Volf, and Roubal 1973). There are several reasons why modern control has not been wide spread in the nuclear industry. One reason is the United States utilities have not ordered a new nuclear plant since the late 1970's. The control system design for existing plants is mostly from the 1950's and 1960's with traditional proportional-integral-derivative controllers. Because of the complete stoppage of nuclear industry growth in the United States, there has not been any motivation to use optimal control--until recently. Nuclear plant utilities have been aggressively searching for ways to prevent unnecessary plant trips by applying the latest technology in control systems replacements. Many utilities have decided to increase reliability by retrofitting existing control systems with dual or triple redundant digital computer-controlled systems. Another not so obvious solution is to increase the transient performance of the control system. If transient performance can be significantly improved (holding reactor water level closer to setpoint), more plant trips can be avoided, and plant availability increased. More importantly, safety is enhanced by a reduction in safety system actuations. Less safety system actuations shows that important plant process variables (water level) are staying within the normal limits. The probability of an accident is thereby reduced and the probability of malfunction of safety equipment is also reduced.

2.2 Literature Search

A literature search was conducted to find books and journal articles that show how linear quadratic/Kalman filter control has been/could be applied to nuclear power plants. A brief summary of these documents is given below.

2.3 Electric Power Research Institute Reports

2.3.1 NP-4769-SR Toshiba Incorporated of Japan, describes the implementation a linear quadratic regulator summed with a simple integrator to achieve a measure of optimal control in a boiling water reactor nuclear power plant feedwater control system (Seiskiro, Makino, Okutani, and Hirayama 1986). Only two linear-quadratic feed-forward gains are computed for directly measured states: one for level and one for a linear combination of steam minus feedwater flow. The paper contends that applying more gains with an observer would not result in significant improvement in control and would require more computation time. Therefore, an observer was not applied.

2.3.2 NP-4919-LD Westinghouse studied optimal control for a Westinghouse pressurized water reactor nuclear power plant feedwater control system (Eastman, Gaydos, Graham, Lipner, Mueller, Nasrallah, Negrus, Paris, Schaefer, Waclo, and Woods 1989). The problems encountered included:

- The reduced-order linear observer could not adequately estimate the plant non-linearities.
- The tuning of the system with linear-quadratic gains was beyond the technical capability of the plant technicians.

Because of these problems linear-quadratic/observer control was not implemented.

The first problem is avoided in this thesis by not attempting to model the shrink and swell reactor water level behavior. In addition, "hard" non-linearities such as compensator and valve minimum and maximum limits are modeled and included in the Kalman filter observer. The second problem mentioned above could be ameliorated by computer-aided calibration and tuning.

CHAPTER 3

CONTROL ALGORITHM DESIGN

3.1 Overview

Linear-quadratic control has been criticized in the past for requiring an accurate complex full order model of the plant (Astrom and Wittenmark 1990, p. 359). However, there are two approaches that can be taken to overcome the difficulty. The first is to generate a reduced-order observer from the full order observer by capturing only the states which define the control. The second, used in this thesis, is to apply system identification techniques to develop a simplified plant model that captures the dominant states without prior knowledge of the full order model.

The simplified plant model or plant "design" model is then used to build the linear-quadratic/Kalman filter compensator. The compensator is then tested using a more accurate and complex non-linear plant model called the "truth" model.

All calculational work is performed on a personal computer and all software packages used are supported by personal computer platforms.

3.2 Development of Plant Design Model

Figure 4 shows a simplified plant model with a proportional-integral controller. This model has been used in the past

(Goto 1973) without the delay for selecting tuning parameters: proportional gain, k_p, integral gain, k_i, and steam minus feedwater flow mismatch gain, k_m. Because this model did not contain a delay, one was added to account for the delay in level response shown in Figure 2. The delay is also apparent at lower power levels.

The model includes a tank as the simplest approximation to the reactor vessel with a conversion factor, k_l, to convert percent flow error (feedwater flow minus steam flow) to inches of water level. The delay in level response is due to the delay in feedwater enthalpy in the annulus. A second order Padé approximation is used to approximate the delay with td as the delay time. The regulating valve receives the control input, u, and is modeled as a second order system with time constant, t_v, damping ratio, zeta (z), and a conversion factor, k_v, to convert the units of the control signal, inches of level, to percent feedwater flow. The level, feedwater flow, and steam flow sensors are modeled as first order lags with time constants t_l, t_f, and t_s respectively. The equations of the plant design model are given below.

$$\dot{x}_1 = -(2z/t_v)x_1 - (1/t_v^2)x_2 + (k_v/t_v^2)u \qquad (3.1)$$

$$\dot{x}_2 = x_1 \qquad (3.2)$$

$$\dot{x}_3 = x_2 - (6/t_d)x_3 - (12/t_d^2)x_4 + x_{o1} \qquad (3.3)$$

$$\dot{x}_4 = x_3 \qquad (3.4)$$

$$\dot{x}_5 = -(6k_1/t_d)x_3 + (12k_1/t_d^2)x_4 - (k_1)x_{o2} \quad (3.5)$$

$$\dot{x}_6 = (1/t_1)x_5 - (1/t_1)x_6 \quad (3.6)$$

$$\dot{x}_7 = (1/t_f)x_2 - (1/t_f)x_7 + (1/t_f)x_{o1} \quad (3.7)$$

$$\dot{x}_8 = -(1/t_s)x_8 + (1/t_s)x_{o2} \quad (3.8)$$

$$\dot{x}_9 = -(1/t_h)x_9 + (1/t_h)x_{o3} \quad (3.9)$$

The equations of the original plant feedwater control system proportional-integral compensator are:

$$\dot{x}_i = -k_i x_6 - k_i k_m x_7 + k_i k_m x_8 + k_i x_9 \quad (3.10)$$

and the control law:

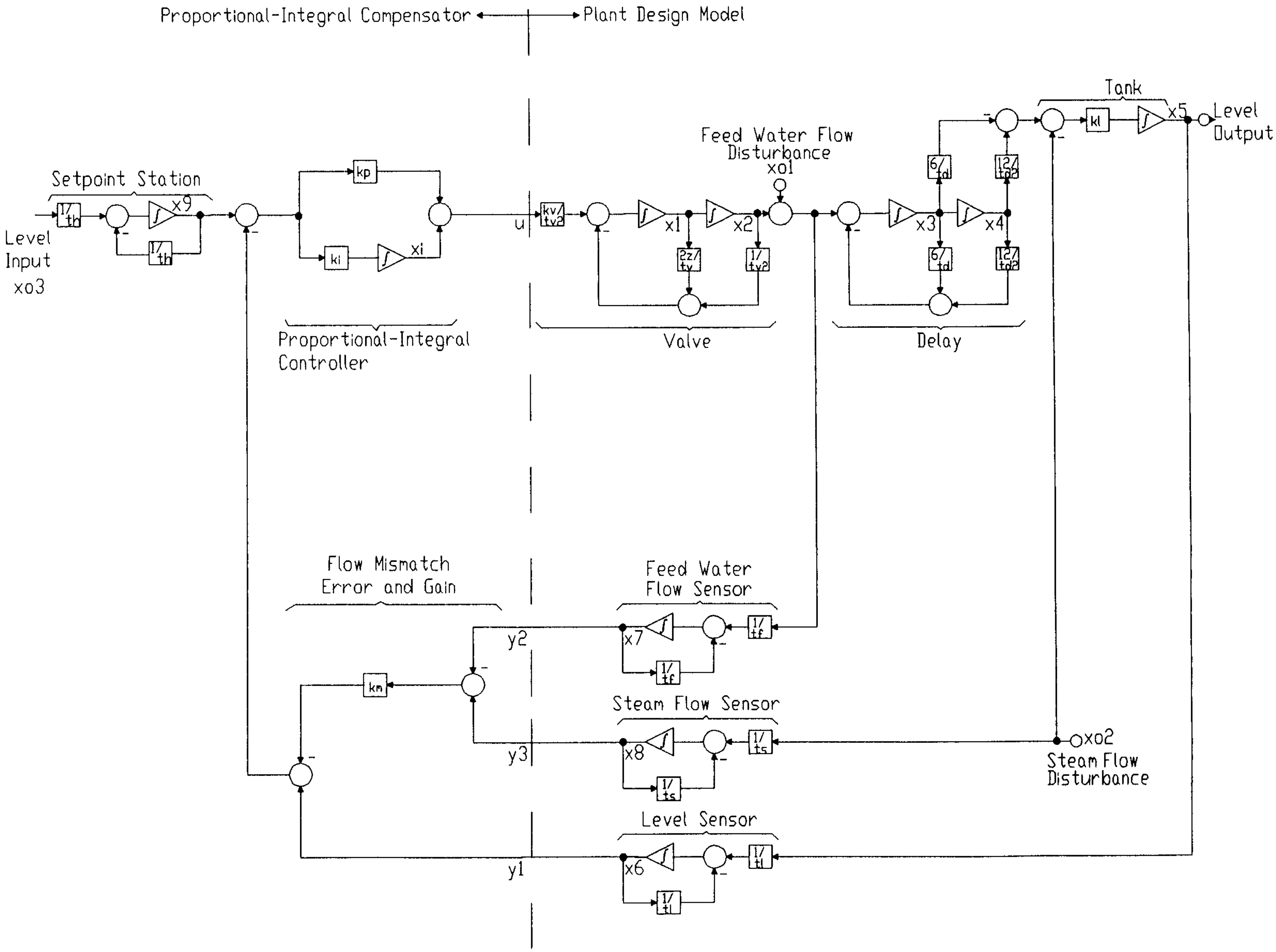
$$u = k_p x_6 - k_p k_m x_7 + k_p k_m x_9 + x_i \quad (3.11)$$

The design model in Figure 4 is simple but captures the salient components of the level response to a step input. One could say that the model is not detailed enough to capture the shrink and swell (water volume increase and decrease) phenomenon discussed earlier in the introduction. It is realized that better control could be obtained if the shrink and swell behavior of reactor water level were adequately modeled. However, Eastman, Gaydos, Graham, Lipner, Nueller, Narsralla, Negrus, Paris, Schaefer, Waclo, and Woods (1989) found that the shrink and swell behavior could not be adequately modeled with a reduced order (low order) observer. This suggests that to adequately model shrink and swell behavior a high order observer is necessary.

This thesis explores the possibility of modern control applied without modeling the shrink and swell behavior. Even though the approach taken is simplified, the claim is made that the compensator to be designed will perform better than the existing proportional-integral compensator and the performance of the design will be proven on a "truth" model that includes shrink and swell phenomenon. Steam flow is modeled as a disturbance (x_{o2}) because it is not controlled by the feedwater control system. A disturbance in feedwater flow (for example a feedwater pump trip) is modeled by x_{o1}.

It should be noted that the current Oyster Creek Plant is different now than that tested in 1969. The thermal power rating of the reactor was upgraded from 1600 to 1930 megawatts, the feedwater control system gains have been changed, a lead/lag filter added, and the feedwater control valve has been modified. Because test data do not exist for the latest plant configuration, this thesis will use the old plant test data to compare the performance of the modern control compensator to the proportional-integral compensator. The comparison with the old plant is a fairer comparison because the existing system has not been tuned for a long time and has a much slower response. The data shown in Figure 2 was taken after the feedwater control system proportional-integral controller was tuned with the parameters that gave the fastest response.

Figure 4 Plant Design Model



3.3 Parameter Identification

Table 2 summarizes the attempts made to define the parameters that match the design model in Figure 5 to the plant test data in Figure 2. The reference values are obtained from reference documents or calculated as follows. The test performed in 1969 showed the proportional gain, k_p, and integral gain, k_i, to be 1.25 and 0.0083 seconds^{-1} respectively. The steam flow minus feedwater flow mismatch gain, k_m, was not listed for the 100% power test. However, k_m is listed as 0.36 inches/% flow for tests at lower power levels and therefore is presumed to be 0.36 inches/% flow. Time constants t_l, t_f, and t_s are listed as 1.00, 0.25, and 0.25 seconds respectively in the final transient analysis for Oyster Creek. The time constant for the setpoint station, t_h, is chosen to be consistent with the flow sensors' time constants at 0.25 seconds. The setpoint is manually changed by the operator. This leaves k_l, t_d, k_v, z, and t_v as parameters which need to be established. The reference value for the percent-flow-to-inches-of-level conversion factor, k_l, was initially taken to be 0.05 inches/% flow as given by an analysis performed by General Electric Company in March, 1973 for another plant. However, experience with the Oyster Creek licensed plant model, RETRAN04 MARKII, showed k_l to be smaller than 0.05 inches/% flow and closer to 0.02 inches/% flow. Therefore, 0.02 is used as the reference value. The reference value for the valve time constant, t_v, should be in the range of 0.1 to 0.5 seconds, per original General Electric Company specifications for the valves.

A value for zeta (z), the damping ratio is initially guessed at 1.0. The inches of level to percent flow value conversion factor, k_v, is calculated as $k_v = (k_a)(k_b) = 3.33$ % flow/inch where k_a = (40 mA signal/96 inches of level) (100% stroke/40 mA signal) = 3.20 %flow/ %stroke. Time delay, t_d, is taken from the data in Figure 2 as 6 seconds. This delay ranged between five and six seconds for similar tests at 25% and 75% power. Six seconds is used as it is the longer, more conservative reference value.

3.4 Parameter Estimation

The vessel design model is loaded into a software package called MATRIX X/PC (Matrix X/PC User's Guide 1990) in traditional input-output form (see Figure 5) and a maximum likelihood routine is used to estimate parameters t_d and k_l. The feedwater flow and reactor level data shown in Figure 2 was tabulated and loaded into two vectors in the MATRIX X software. The delay time, t_d, and the level gain, k_l, are estimated by exciting the vessel model shown in Figure 5, with the feedwater flow data and directing the iterative maximum likelihood routine to match the simulated vessel

Table 1 Summary of Plant Design Model Parameters

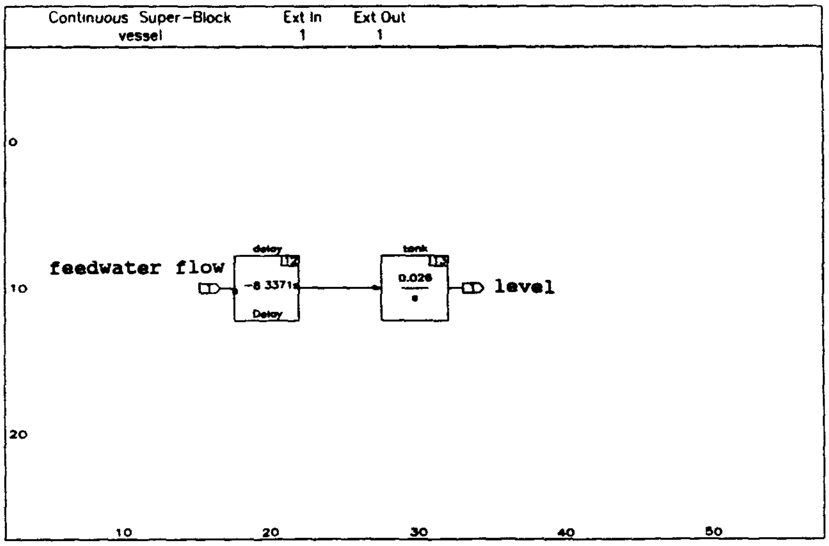
PARAMETER NAME	REFERENCE VALUE	VESSEL MODEL	VALVE MODEL 1	VALVE MODEL 2	FINAL
k_p(unitless)	1.250				1.250
k_i(seconds^{-1})	0.0083				0.0083
k_m(inches/% flow)	0.3600				0.3600
k_l(inches/% flow)	0.0200	0.0260			0.0260
k_v(%flow/inch)	3.330		5.810	5.546	5.546
z(unitless)	1.000		0.4173	0.6952	0.6952
t_v(seconds)	0.5000		0.0818	0.2292	0.2292
t_l(seconds)	1.000				1.000
t_f(seconds)	0.2500				0.2500
t_s(seconds)	0.2500				0.2500
t_d(seconds)	6.000	8.337			8.337
t_h(seconds)	0.2500				0.2500

output with the actual level data. The "vessel model" t_d and k_l parameter values in Table 2 calculated by the maximum likelihood routine are 8.337 seconds and 0.0260 inches level/% flow. The t_d and k_l values were obtained when the "maxlike" routine converged after seven iterations with initial guesses of 6 and 0.035 respectively. Figure 5 also shows the approximated level with t_d and k_l parameters calculated by the maximum likelihood routine.

"Valve model 1" values in Table 2 are obtained by starting with the input-output blocks shown in Figure 6 along with the vessel model shown in Figure 5. The advantage of the small signal (perturbation) model in Figure 6 is that steam flow change is zero about an operating point thus simplifying the analysis. This time k_v, z, and w_n (defined to be $1/t_v$) are allowed to vary. The model is excited with a -10 inch step an the "maxlike" routine (Matrix X/PC User's Guide 1990) is directed to match the valve output to feed-water flow tabulated data. The "valve model 1" parameters k_v, z, and t_v values in Table 2 are calculated by the maximum likelihood routine as 5.810, 0.4173, and 0.0818 ($1/w_n$) after fourteen iterations with initial guesses of 3.2, 1.0, and 0.06 respectively.

Because t_v (calculated with valve model 1) is outside the range of the normal value of 0.1 to 0.5 seconds, a second

Figure 5 Simplified Vessel Model



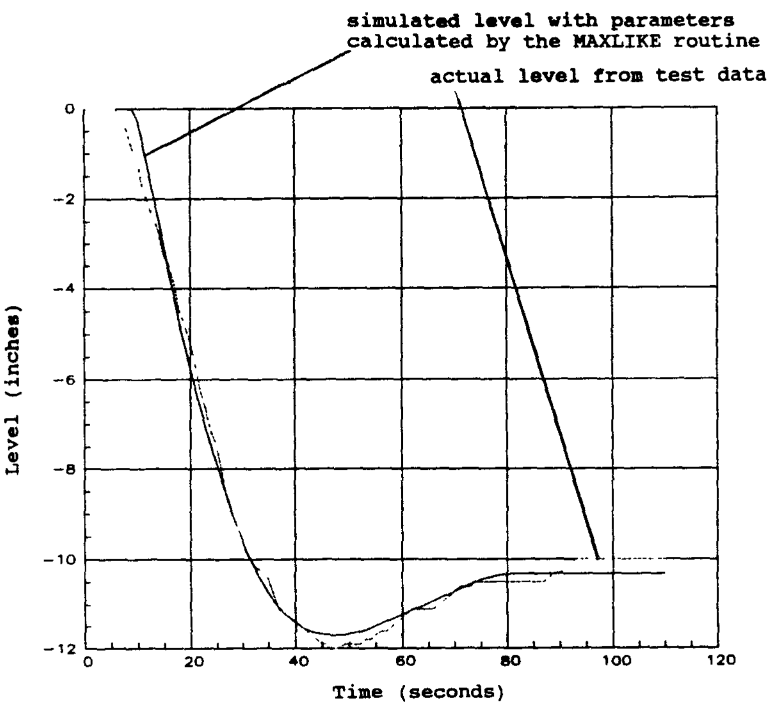
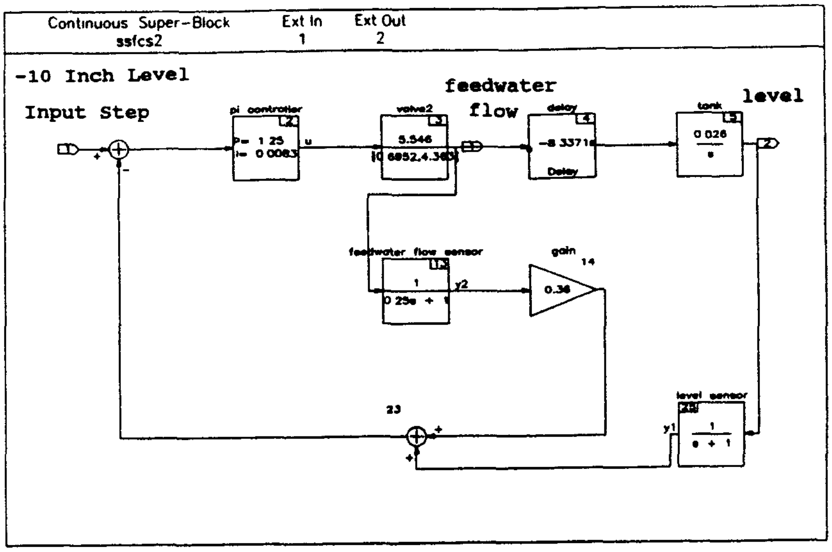
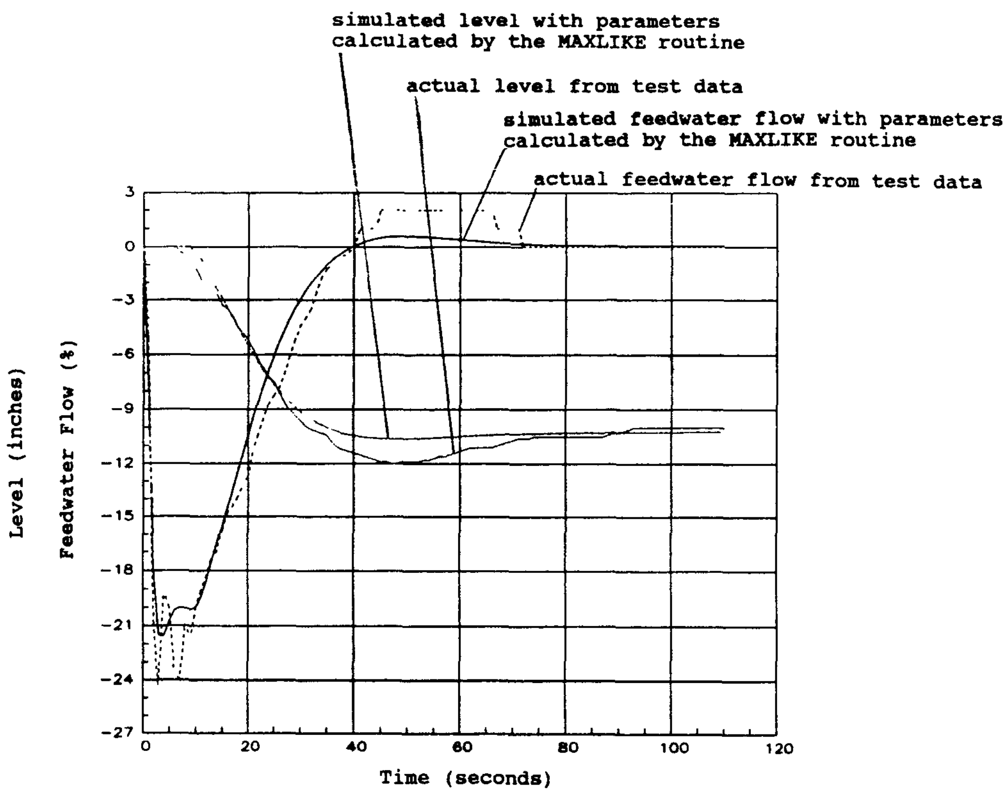


Figure 6 Small Signal Plant Design Model





trial is initiated: valve model 2. Valve model 2 started with initial guesses 3.2, 1.0, 0.5 for k_v, z, and t_v respectively and the "maxlike" routine diverged. Next, t_v was changed to 0.167 and the "maxlike" routine converged after 4 iterations to 5.546, 0.6952, and 0.2292 for k_v, z, and t_v respectively. Since these values are more in line with the reference values, the "valve model 2" values are taken as final. The fact that k_v is almost a factor of two larger than the calculated value maybe accounted for because of other conversion factors that are not represented by the simplified design model. Figure 6 shows the graphical results of the parameter estimation with the final parameters. A reasonable match to the plant test data validates the plant design model. The final values used for the plant design model parameters are given in Table 2.

3.5 Development of Control Gains

A control systems software package provided by New Jersey Institute of Technology includes a program called CSD (CSD User's Guide 1985) to calculate the linear quadratic control gains, Kalman filter gains and exogenous gains. The appropriate matrices A, B, C, E, Q, R, V, and W are loaded into the CSD program and the appropriate gain matrices G and K are calculated by solving the respective algebraic Riccati equation. The algebraic Riccati equation is solved by Laub's algorithm.

The choice of the state performance weighting matrix, Q, began with unity in the Q(5,5) position since the Q(5,5) position represents the most important state: the level. The choice of the control performance weighting matrix began with unity.

The initial choice for process noise spectral density matrix, V, was unity in the V(1,1), V(10,10) V(11,11), and V(12,12) positions to inject artificial white noise into the control input and the two disturbance inputs and reference input respectively. Unity was also chosen for observation noise spectral density matrix, W, positions W(1,1), W(2,2), and W(3,3) to inject noise into the sensor observations. Since the sensors are all differential pressure measurement devices of comparable quality, W(1,1), W(2,2) = W(3,3).

Figure 7 shows a realization of the general control structure shown in Figure 3. The plant design model is essentially duplicated in the Kalman filter. Observation estimates $\hat{y}_1$, $\hat{y}_2$, and $\hat{y}_3$ are subtracted from observations y_1, y_2, and y_3 respectively to develop residuals r_1, r_2, and r_3 respectively. The residuals are modified by the Kalman filter gains k_{11}, k_{12}, . . ., k_{o22}, summed together, and then input to each state variable derivative. The estimated states $\hat{x}_1$, $\hat{x}_2$, . . ., $\hat{x}_{o3}$ are modified by the linear quadratic gains, summed together, and provide the control input.

The equation for the control input is:

$$u = -g_1\hat{x}_1-g_2\hat{x}_2-g_3\hat{x}_3-g_4\hat{x}_4-g_5\hat{x}_5-g_{o1}\hat{x}_{o1}-g_{o2}\hat{x}_{o2}-g_{o3}\hat{x}_{o3} \qquad (3.12)$$

where the g values are the linear quadratic gains. It was found that the gains for x_6, x_7, x_8, and x_9 states were calculated to be zero showing that control is possible without modeling the sensors and setpoint station. The equations for the Kalman filter are:

$$r_1 = y_1 - \hat{x}_6 - \hat{x}_9 \quad (3.13)$$

$$r_2 = y_2 - \hat{x}_7 \quad (3.14)$$

$$r_3 = y_3 - \hat{x}_8 \quad (3.15)$$

$$\dot{\hat{x}}_1 = -(2z/t_v)\hat{x}_1 - (1/t_v^2)\hat{x}_2 + (k_v/t_v^2)u + k_{11}r_1 + k_{12}r_2 + k_{13}r_3 \quad (3.16)$$

$$\dot{\hat{x}}_2 = \hat{x}_1 + k_{21}r_1 + k_{22}r_2 + k_{23}r_3 \quad (3.17)$$

$$\dot{\hat{x}}_3 = \hat{x}_2 - (6/t_d)\hat{x}_3 - (12/t_d^2)\hat{x}_4 + \hat{x}_{o1} + k_{31}r_1 + k_{32}r_2 + k_{33}r_3 \quad (3.18)$$

$$\dot{\hat{x}}_4 = \hat{x}_3 + k_{41}r_1 + k_{42}r_2 + k_{43}r_3 \quad (3.19)$$

$$\dot{\hat{x}}_5 = -(6k_1/t_d)\hat{x}_3 + (12k_1/t_d^2)\hat{x}_4 - (k_1)\hat{x}_{o2} + k_{51}r_1 + k_{52}r_2 + k_{53}r_3 \quad (3.20)$$

$$\dot{\hat{x}}_6 = (1/t_1)\hat{x}_5 - (1/t_1)\hat{x}_6 + k_{61}r_1 + k_{62}r_2 + k_{63}r_3 \quad (3.21)$$

$$\dot{\hat{x}}_7 = (1/t_f)\hat{x}_2 - (1/t_f)\hat{x}_7 + (1/t_f)\hat{x}_{o1} + k_{71}r_1 + k_{72}r_2 + k_{73}r_3 \quad (3.22)$$

$$\dot{\hat{x}}_8 = -(1/t_s)\hat{x}_8 + (1/t_s)\hat{x}_{o2} + (k_{81})r_1 + k_{82}r_2 + k_{83}r_3 \quad (3.23)$$

$$\dot{\hat{x}}_9 = -(1/t_h)\hat{x}_9 + (1/t_h)\hat{x}_{o3} + k_{91}r_1 + k_{92}r_2 + k_{93}r_3 \quad (3.24)$$

$$\dot{\hat{x}}_{o1} = k_{o11}r_1 + k_{o12}r_2 + k_{o13}r_3 \quad (3.25)$$

$$\dot{\hat{x}}_{o2} = k_{o21}r_1 + k_{o27}r_2 + k_{o23}r_3 \quad (3.26)$$

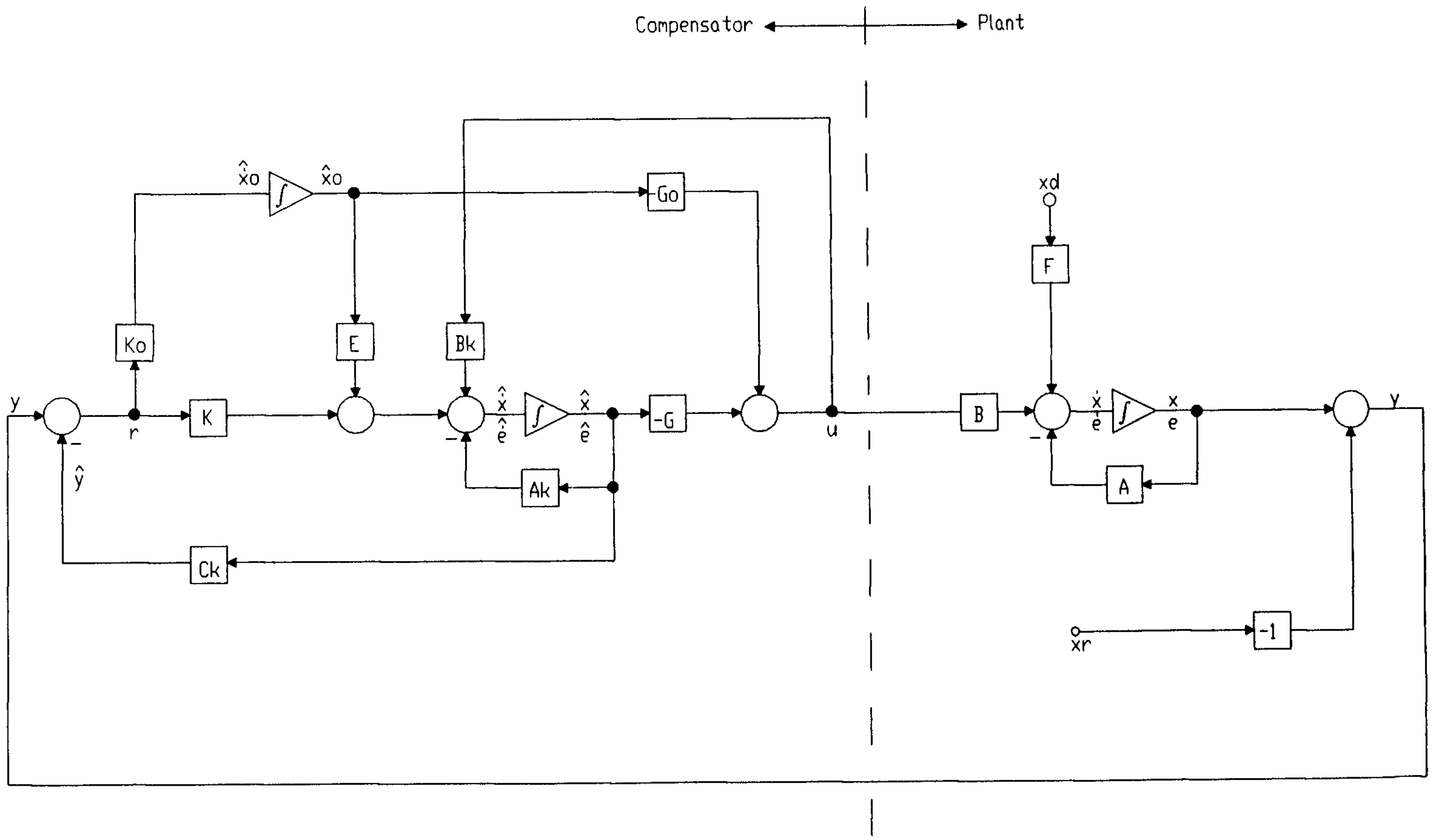
$$\hat{x}_{o3} = k_{o31}r_1 + k_{o32}r_2 + k_{o33}r_3 \quad (3.27)$$

3.6 Check of Control Gains

The control systems software package provided by New Jersey Institute of Technology, in addition to the control gain calculation, CSD, includes a good simulation program called ALSIM (ALSIM User's Guide 1990). ALSIM uses a fourth order Runge-Kutta routine with a variable step size for solving the system first order differential equations. The design model including the modern control compensator is loaded into the ALSIM program and plots are obtained. The listing is shown in the Appendix.

First the linear, quadratic gains are checked using full-state feedback. After several iterations of simulating, choosing different state weighting (Q) and control weighting (R) matrices, recalculating the control gain matrix (G), and then simulating again, the "best" G matrix is chosen. The "best" G matrix gives the fastest level performance while maintaining a reasonable control magnitude (say twice the normal 100% flow value) when the system is excited with 100% step inputs for x_{o1}, x_{o2}, and x_{o3} separately. It should be noted that limits will be installed on the control input and valve in the truth model giving a realistic response.

Figure 7 Modern Control Structure



Secondly, the Kalman filter is inserted into the compensator such that no plant states are directly fed back to the linear, quadratic regulator. The "best" Kalman filter gain matrix (K) is found after several iterations of simulating, choosing different process noise (V) and observation noise (W) spectral density matrices, recalculating the Kalman filter gain matrix (K), and simulating again. The best K matrix gives the best level and other state estimates which can be seen by overlaying the actual and estimated values.

Figure 8 shows the response to a -10 inch step with the best G and K matrices. The best matrices were found to be Q(5,5) = 1, R = 0.100, V(1,1) = 10000, V(10,10) = V(11,11) = V(12,12) = 1, and W(1,1) = W(2,2) = W(3,3) = 1. The initial "bump" in the level response is due to the second order Padè approximation for the delay. The bump will not be seen in the actual plant because a pure process delay exists, not a second order approximation. Also, the control signal output from the compensator is filtered by the feedwater system and by the time it gets translated into a level change the "bumps" are ironed out.

Figure 8 shows that significant settling time improvement is possible. Later, the modern control compensator is incorporated in the truth model and give a more realistic test of the compensator.

Figure 8 also shows that the Kalman filter estimates are right on top of the "actual" response curves. Also, a comparison of full state feedback to Kalman filter feedback shows only a small degradation in performance.

The main purpose of the design model is to provide a starting point to apply the G and K gain matrices.

3.7 Stability of Design Model

The linear plant design model is loaded into MATRIX X in matrix format (Matrix X/PC User's Guide 1990). A "system" matrix is constructed of the plant:

$$S_p = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \tag{3.25}$$

and compensator:

$$S_c = \left[\begin{array}{c|c} A-K*C-(B-K*D) & K \\ \hline G & O \end{array}\right] \tag{3.26}$$

The S_c matrix is formed using the "LQCOMP" MATRIX X command. The MATRIX X "AFEEDB" command is then used to connect S_p and S_c in a feedback configuration (see Figure 9) thus forming one matrix, S_{cl} which represents the closed loop system. In this form, the MATRIX X "EIG" and "ZEROS" commands are executed on S_{cl} to obtain the closed loop eigenvalues and transmission zeros respectively. The open loop eigenvalues are also obtained using the "EIG" command. Figure 9 shows the configuration of the S_p and S_c matrices, the eigenvalues and the transmission zeros. All eigenvalues are in the right-half plane indicating closed loop stability.

A Bode plot is obtained by connecting the S_p and S_c matrices in series using the "SERIES" command and is shown in Figure 10. The MATRIX X "BODE" command is used to obtain the plots. The gain and phase stability margins are calculated by MATRIX at 20.21 decibels (at 5.342 radians/second) and (at 0.6429 radians/ second) 65.95 degrees respectively.

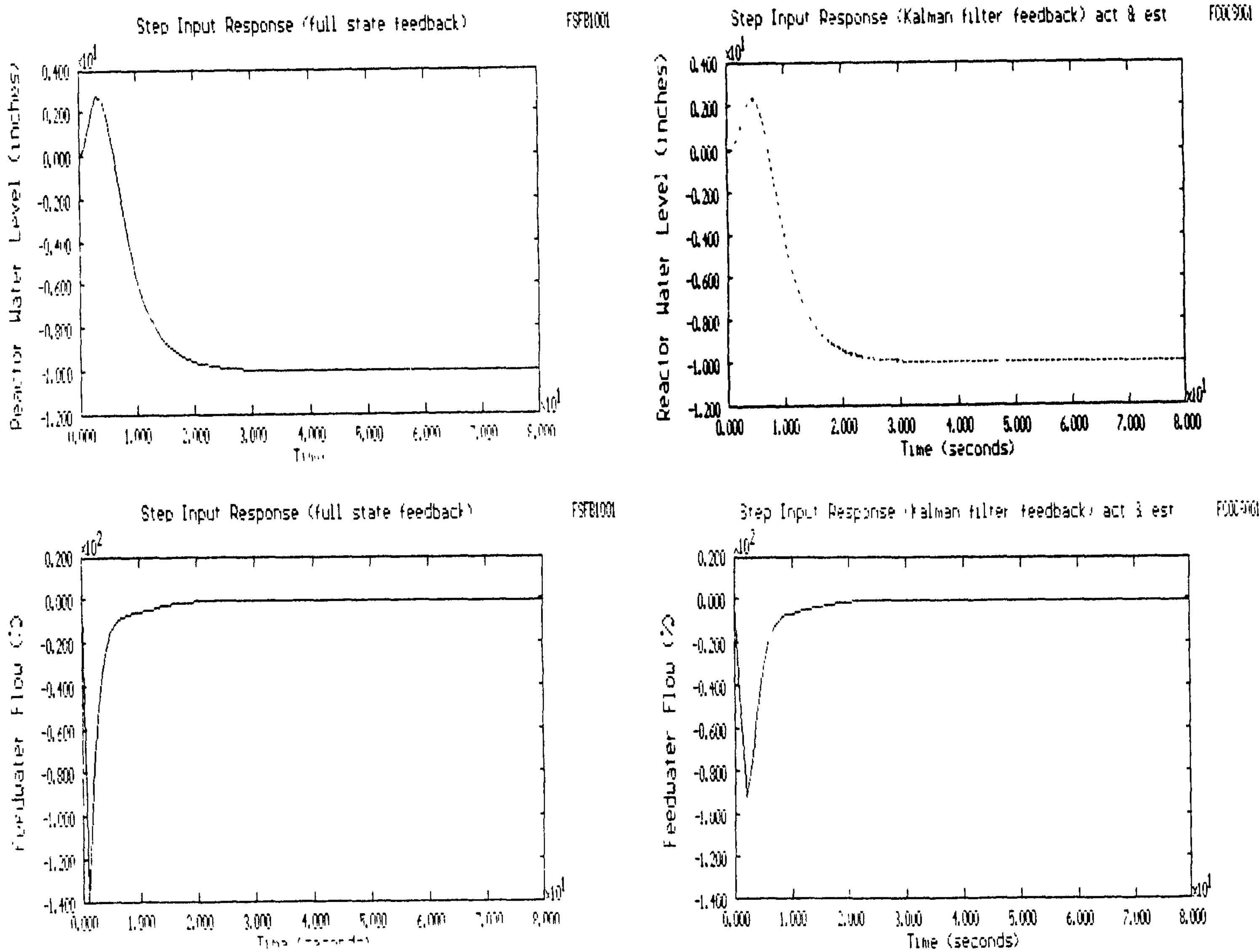
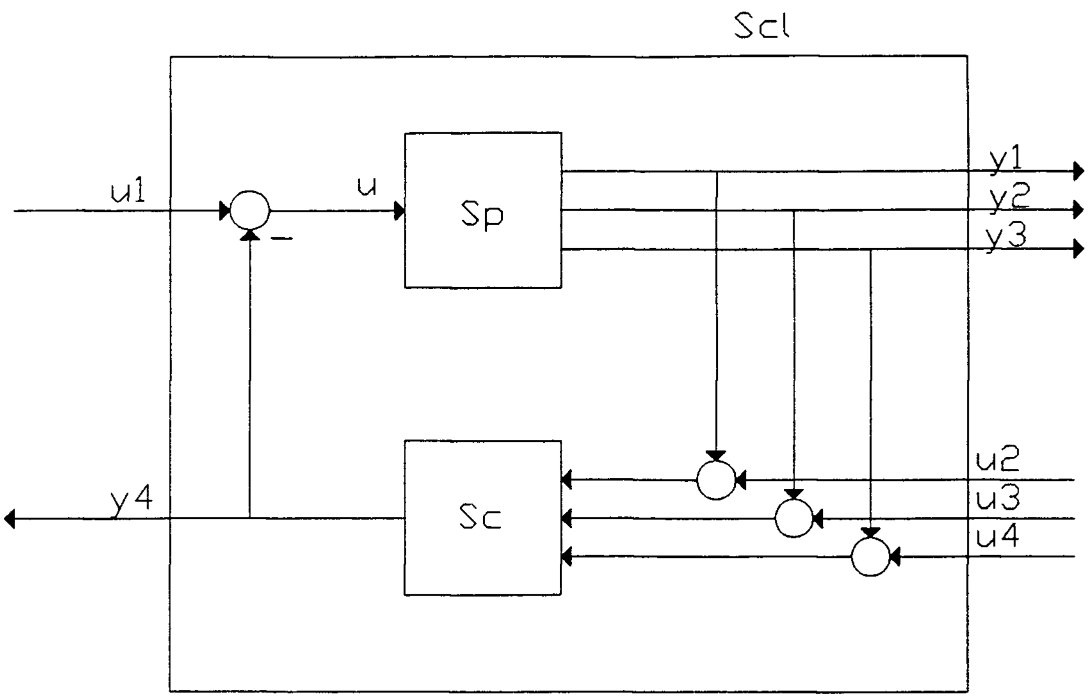


Figure 8 Check of Control Gains

Figure 9 Eigenvalues and Transmission Zeros of Design Model



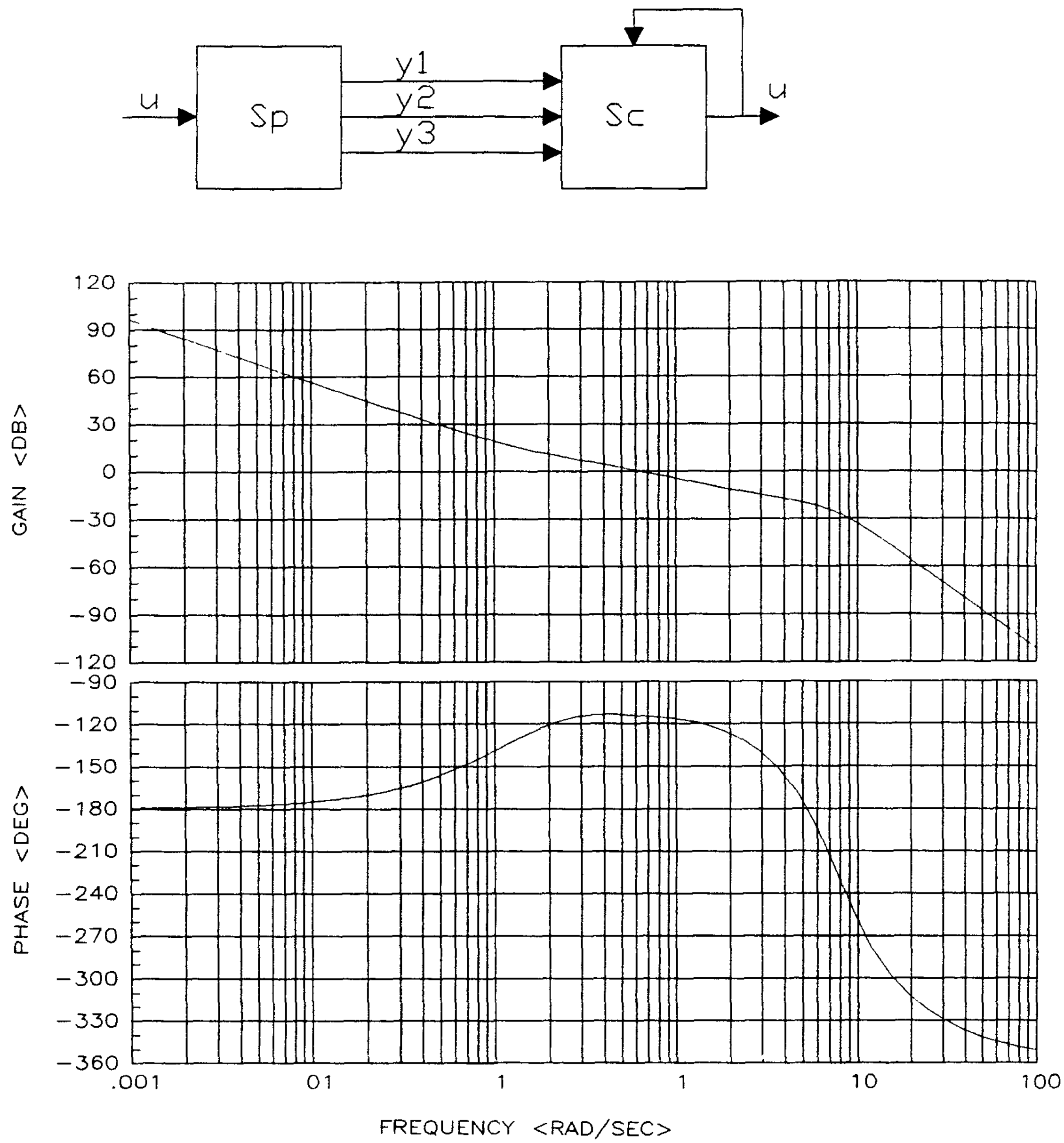
<u>Eigenvalues</u>

open loop (Sp)	closed loop (Sc)
0	-0.1870
-0.3599+j0.2076	-0.4606+j0.3865
-0.3599-j0.2076	-0.4606-j0.3865
-1.000	-1.059
-4.000	-3.864
-4.000	-7.611
-4.000	-4.000
-3.033+j3.135	-3.993+j6.128
-3.033-j3.135	-3.993-j6.128

<u>Transmission Zeros</u>

-4.000

Figure 10 Bode Plot of Design Model



	Margin	Frequency
Gain	20.21 decibels	5.342 radians/second
Phase	65.95 degrees	0.6429 radians/second

CHAPTER 4

PERFORMANCE OF ADVANCED CONTROL ALGORITHM

4.1 Description of Plant Truth Model

The truth model is a personal-computer based generic boiling water reactor simulation program called PCTRANB developed by Dr. L. C. Po and written in FORTRAN language (Po 1989). Most of the effort in the development of PCTRANB went into the reactor vessel model and emergency systems. The 20 state truth model includes a non-linear vessel model and non-linear valve and controller saturation functions. This is in contrast to the 5 state (neglecting sensor states) linear design model.

Additions are made to the feedwater control system portion including a second order valve model with saturation limits, sensor models, and a fourth order Padé approximation for the feedwater enthalpy delay. The feedwater control system proportional-integral controller is removed and a proportional-integral controller based on state space methods is installed. The reactor pressure regulator is replaced with a proportional-only controller with a lead/lag filter as this is what Oyster Creek and several other boiling water reactors have. The lead/lag filter is also realized using state space methods. The feedwater control system and reactor pressure controllers both have an upper flow of 120% and a lower limit of 0%. The listings for modifications to PCTRANB for this thesis are listed in the Appendix.

The integration routine for solving the plant first order linear and non-linear differential equations is the Euler method with constant time step. For the reactor vessel non-linear first order differential equations, PCTRANB checks second order numerical stability. If the Euler convergence criterion is met, the calculation proceeds, otherwise PCTRANB introduces a correction to the slope.

In order to compare the PCTRANB values to the tabulated test data, a program was written (PCTRAN5.FOR) to load the appropriate values at one second intervals into plot variables. The names of the plot variables in PCTRAN1.FOR were then changed so that plots of PCTRANB variables would be overlayed with the measured data shown in Figure 2.

PCTRANB includes a mimic of the reactor vessel with several important variables and is shown in Figure 11. The mimic shown in Figure 12 is a snapshot of the major plant paramaters at the instant after a main steam isolation valve closure. The mimic is representative of Oyster Creek with the exception that Oyster Creek has five recirculation pumps as opposed to two. From the panel on the right hand side, setpoint changes (including level) can be made interactively while model is running. Figure 12 shows the comparison between the plant truth model and actual plant response to a

-10 inch level input step change. The settling time for the truth model is about 107 seconds with an overshoot of 3%. This is in contrast to the actual plant response of 20% overshoot with a settling time of 91 seconds. A comparison of feedwater flow (actual and modeled) shows why this is so. From 0 to 30 seconds; actual feedwater flow envelopes the modeled feedwater flow and therefore less area is integrated by the vessel--thus less overshoot is apparent in the modeled level. Also, from 30 seconds to 70 seconds actual feedwater flow rises higher than modeled feedwater flow--thus actual level returns to the setpoint faster allowing faster settling time. Steam flow response remains fairly constant during the transient thus validating the assumption for the small signal model used to calculate parameters in Section 3.4. The modeled reactor power does not dip as deep as actual reactor power. This is again because of the modeled feedwater flow response. Less feedwater results in less temperature decrease which in turn causes less of a power reduction.

Overall, it should be noted that the truth model is more "well-behaved" than the actual plant. Therefore, the truth model would have to be refined further prior to an implementation of the modern control compensator. For purposes used in this thesis the truth model serves only as a common ground with which to compare the proportional-integral compensator with the modern control compensator.

Time was available to make one refinement to the valve model. A backlash "block" was added at the input, $\dot{x}2$ (velocity), such that if the velocity input changes within the backlash width, the output velocity is zero, otherwise velocity out equals velocity in. The result is shown in Figure 13 and tabulated output in the Appendix. The modeled level has an overshoot of 6% and settling time of 180 seconds. The backlash makes the valve more sluggish thus causing a longer settling time. The addition of the backlash allows closer agreement to the actual plant in overshoot but more disagreement in valve settling time. It is interesting to note that the addition of valve backlash significantly degrades the performance of the proportional-integral compensator.

4.2 Incorporation of Compensator in Plant Truth Model

Because values in the Kalman filter are calculated in percent flow, steam flow and feedwater flow observations are scaled from pound mass/second to percent flow. The control input is then rescaled from percent flow to pound mass/second.

The Kalman filter first order differential equations are solved using the same fixed time step used throughout the program with the Euler method and no slope correction. Because the state space level input, x_{o3}, is referenced to the origin, a model of the setpoint station (first order lag) is added to the plant model to make the level track the actual

Figure 11 PCTRANB Reactor Vessel Mimic

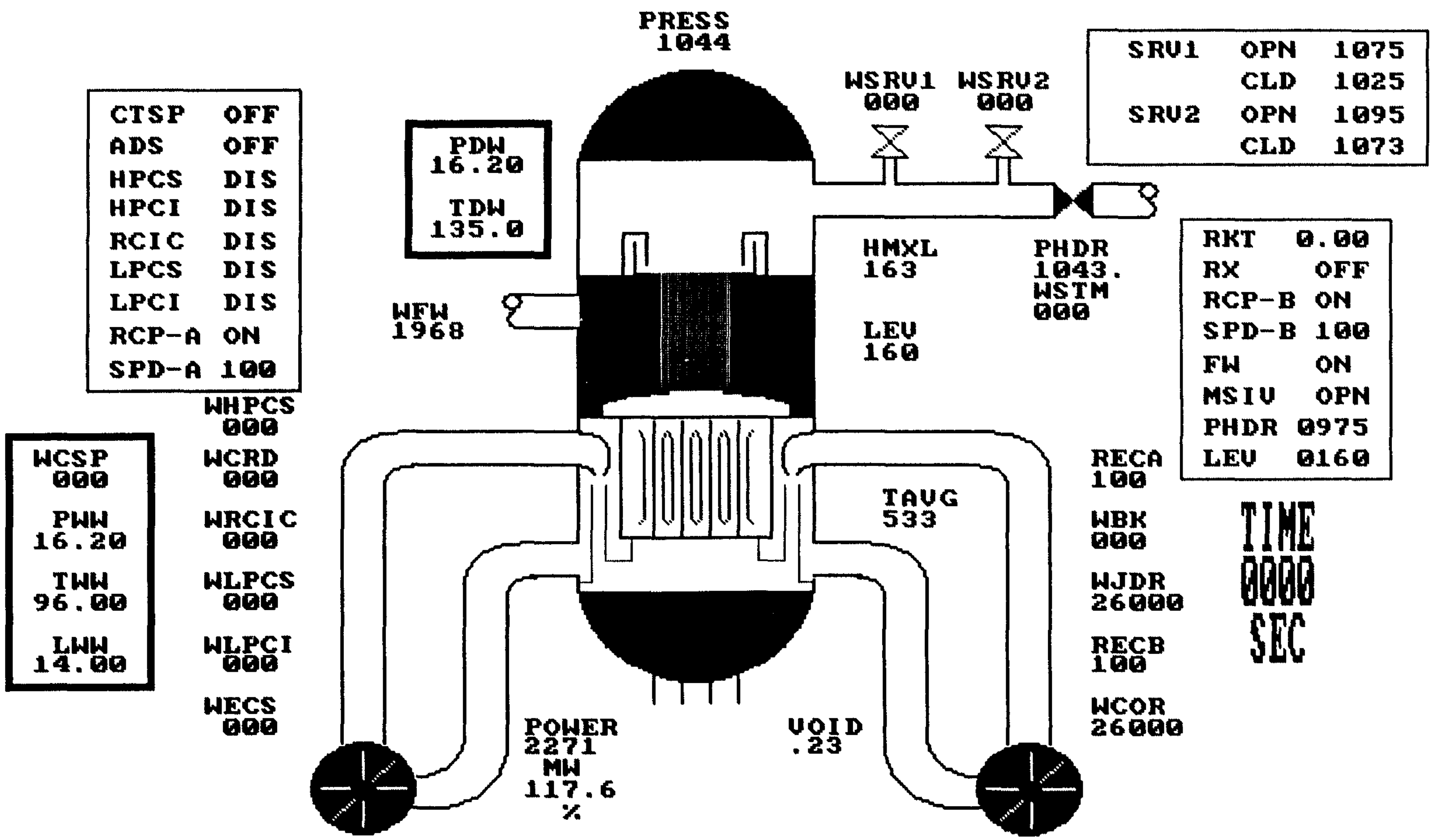


Figure 12 Comparison of Plant Truth Model to Actual Plant

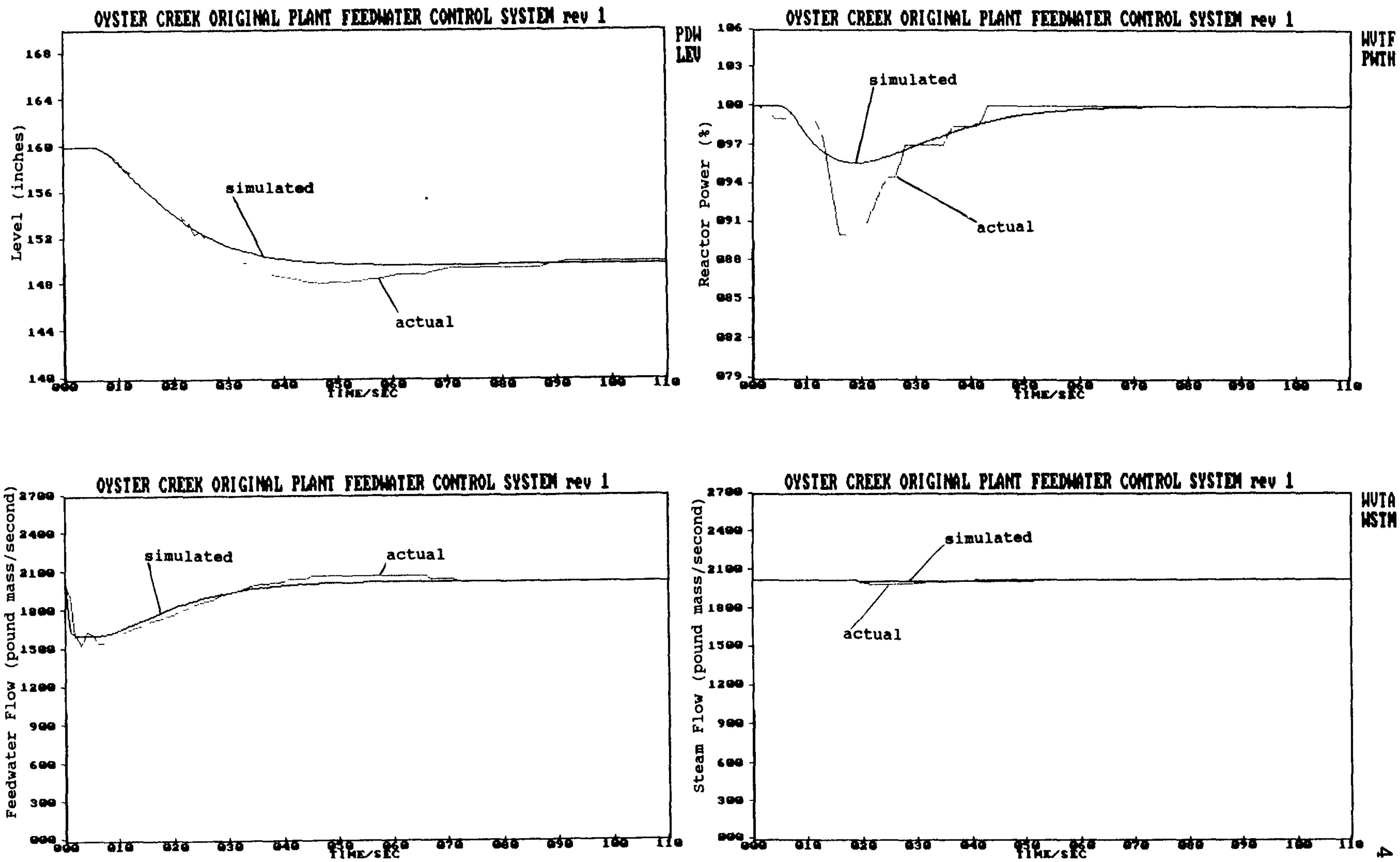
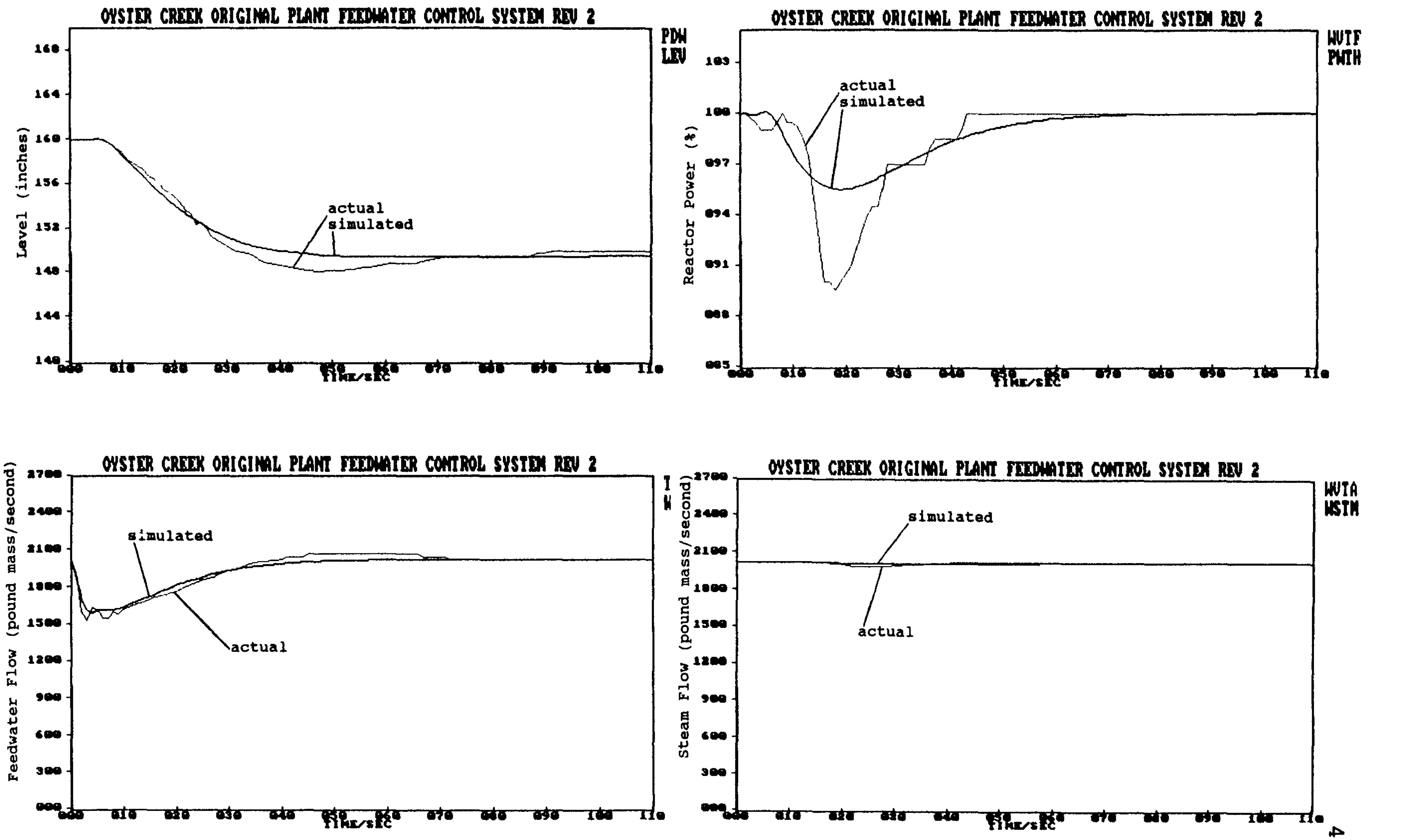


Figure 13 Comparison of Plant Truth Model (with backlash) to Actual Plant



setpoint of 160 inches (not zero). The following equation is added to the plant "truth" model (as in the "design" model):

$$\dot{x}_9 = -(1/t_h)x_9 + (1/t_h)x_{o3} \qquad (4.1)$$

The second-order Padè approximation for the delay is maintained in the Kalman filter and is used to approximate the fourth-order delay incorporated in the plant truth model. It was found that a better approximation for the delay is obtained by using:

$$e^{-std} \approx \frac{1 - (td/2)s + (td^2/8)s^2}{1 + (td/2)s + (td^2/8)s^2} \qquad (4.5)$$

as opposed to:

$$e^{-std} \approx \frac{1 - (td/2)s + (td^2/12)s^2}{1 + (td/2)s + (td^2/12)s^2} \qquad (4.6)$$

Equation 4.5 gives a smaller magnitude initial "bump" that is apparent in second order approximations for time delay and therefore was used in the Kalman filter.

Many nuclear plant simulators use the Euler method for integration and use 0.250 seconds for the time step. It was found that to work well, the compensator required a smaller time step (0.01 seconds). The valve was being controlled in a "jittery" fashion with the 0.1 second time step. For time steps greater than 0.1 second, the compensator would not work at all. this is not surprising since the valve time constant is 0.2292 seconds, and in order for the integration to work properly, the time step should be several times smaller than the smallest time constant (0.2292).

The upper and lower limits of 120% and 0% were maintained for both the compensator output control signal and the valve output signal (flow). The same non-linear saturation limits were incorporated into the Kalman filter making the Kalman filter essentially a non-linear filter when the respective signals go into saturation. Saturation is not expected for the -10 inch level step input but is expected for large transients such as main steam isolation.

4.3 Comparison of Estimated States to Actual States

In order to see how well the Kalman filter is estimating the states, several plot variables were changed in PCTRAN1.FOR so that estimated states could be overlayed with actual plant states calculated by PCTRANB.

Figure 14 compares plots of Kalman filter estimated states with actual plant states (for Kalman filter feedback). For a -10 inch level input step the states are estimated well.

4.4 Comparison of Kalman Filter Feedback to Full State Feedback

To get an idea of the performance, were the Kalman filter to perfectly estimate the states, all states were fed back directly to the linear, quadratic regulator thereby bypassing the Kalman filter. In order to accomplish full state feedback it is necessary to remove the fourth order Padé approximation and install a second order Padé approximation for the delay in the plant truth model. This is because the linear, quadratic gains were calculated with for the states of a second order delay which are not compatible with the states of the fourth order delay. Figure 14 also compares the response to a -10 inch level step input for the Kalman filter feedback to that of the full state feedback. It is seen that the artificial improvement in performance afforded by full state feedback is small.

A number of experiments were conducted to find the cause of the periodic disturbance in feedwater flow during steady state. First, full state feedback shows that the cycling of ±5% is present without the Kalman filter. Therefore, the linear, quadratic regulator is suspect since the proportional integral compensator does not experience the valve cycling. However, the cycling is not evident in the ASLIM full state feedback simulation - even with a larger time step of 0.1 seconds. Therefore, the problem lies either in the integration method (ALSIM uses a much more accurate method) or the fidelity of the linear, quadratic regulator and the plant truth model.

Secondly, different time steps were applied and it was found that by increasing the time step the magnitude and frequency of the disturbance would change. This suggests there is some aliasing of the sample period into the control signal. The disturbance was distinguished from the signal by making the time step small: 0.0001 seconds. The disturbance went completely away when the reactor vessel equations' integration routine was changed to the Euler method with no slope correction to be consistent with the routine used for the other equations in the truth model. This leads to a conclusion that an accurate, routine is required for testing and should be used globally. There was only enough time to obtain one plot with 0.0001 second time step and the Euler method used globally and this is shown in Figure 16.

4.5 Comparison of Linear, Quadratic/Kalman Filter Compensator the Proportional-Integral Compensator

Figure 15 shows a comparison of the response of the linear, quadratic/ Kalman filter compensator to that of the proportional-integral compensator. It can be seen that settling time is improved from 107 seconds with 3% overshoot to 34 seconds (see also Figure 16) with no overshoot. By examination of the feedwater flow response it can be seen why there is an improvement. The modern control compensator moves the valve much further than the proportional-integral compensator thus making level more responsive. In order for the proportional-integral compensator to move the valve the same amount as the modern control compensator, the proportional gain would have to be increased--possibly causing instability in the control loop.

Figure 17 shows a compensation of the two compensators with valve backlash installed. Significant improvement is evident. Settling time is improved from 180 seconds with 6% overshoot to 34 seconds with no overshoot. The settling time for the proportional-integral compensator is increased from 107 seconds to 180 seconds, but settling time for the modern control remains unchanged. The modern control compensator is more robust than the proportional-integral compensator for valve backlash changes. This is because the modern control compensator is getting more information (position and velocity) fed back than the proportional integral compensator.

Figure 18 shows the response for a large transient-main steam isolation (valve closure) at 100% power. Here, the reactor trips on high power due to the closure of the main steam isolation valves. It can be seen that even though level is not estimated well, actual level responds very well. This is because the steam flow and feedwater flow observations compensate for the inaccurate level estimate in the Kalman filter. Feedwater flow is estimated well and initially follows steam flow but rapidly increases when low level is sensed. It can be seen that there is no delay in level response due to the rapidity of the transient.

Figure 19 shows a dramatic improvement in response to a main steam isolation. Here, level is maintained by the modern control compensator at 160 inches and does not go as high as the proportional-integral compensator. Note that because of feedwater valve leakage level will continue to rise until the feedwater pumps are tripped - even though the feedwater control valve is completely closed.

Figure 20 shows the response for another large transient--a reactor trip from 100% power. Here, the reactor is manually tripped causing a void (steam bubble) collapse resulting in an uncontrollable decrease in level or "shrink" (this is also evident in the main steam isolation transient). Again, as shown in Figure 19, level is not estimated well because of the uncontrolled, large "shrink" in level that is not modeled in the Kalman filter. However, again the feedwater flow and steam flow measurements compensate and level recovers to the normal setpoint after the initial shrink. Figure 21 shows significant improvement in level recovery due to the modern control compensator. The proportional integral compensator does not decrease feedwater fast enough causing level to increase way beyond the setpoint of 160 inches and remain at about 174 inches for over a minute. Figure 21 is more evidence of improvement in transient response due to the modern control compensator.

Figure 14 Comparison of Estimated States to Actual States and Full State Feedback

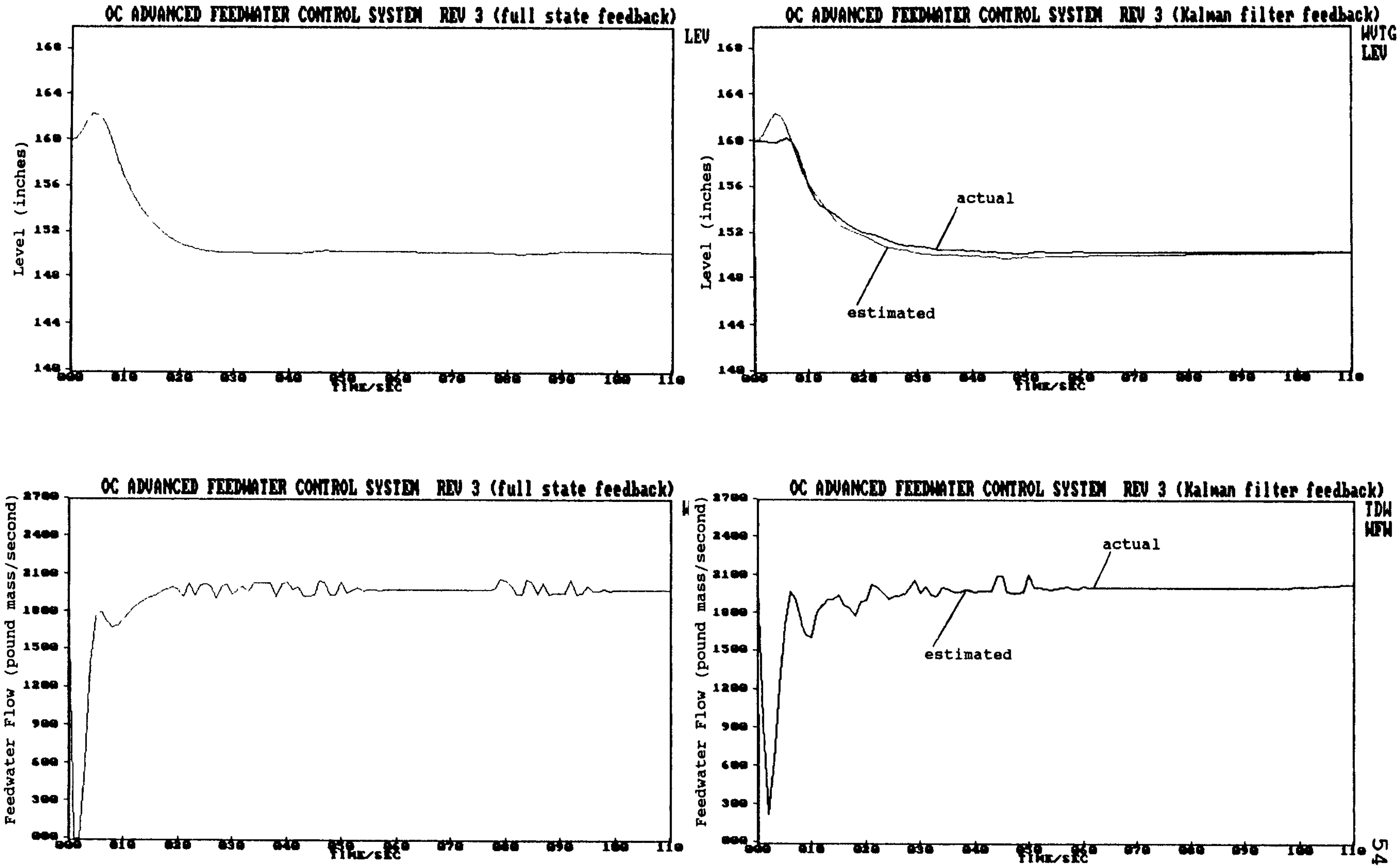
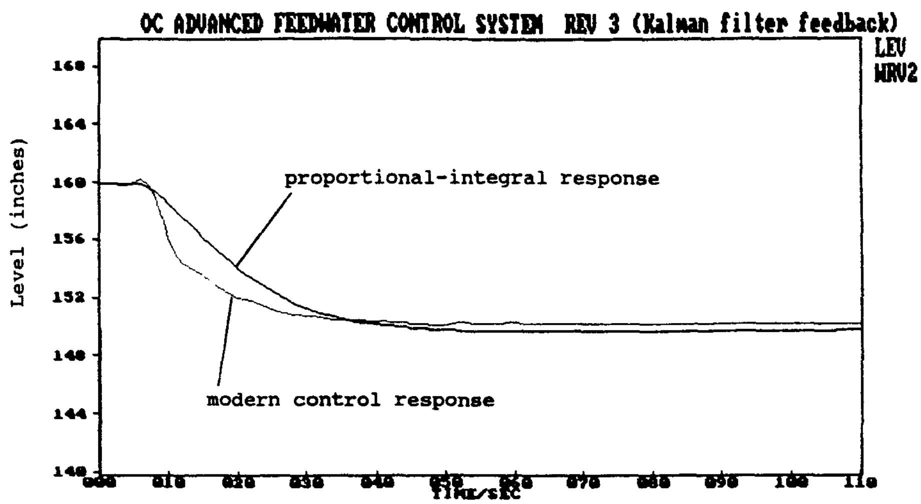


Figure 15 Comparison of Response to a -10 Inch Level Input Step



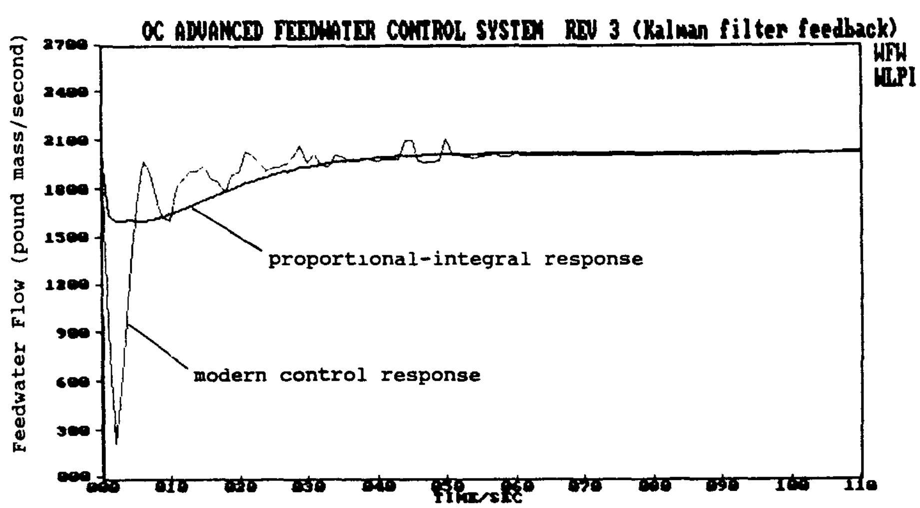
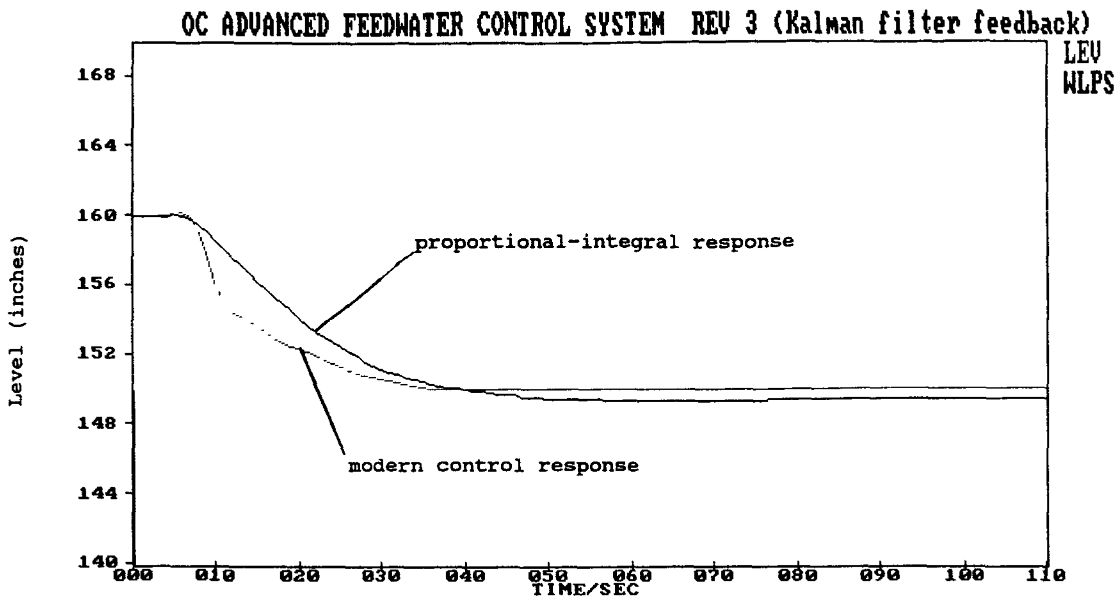


Figure 16 Comparison of Response With Time Step at 0.0001 Seconds Using Euler Method Globally



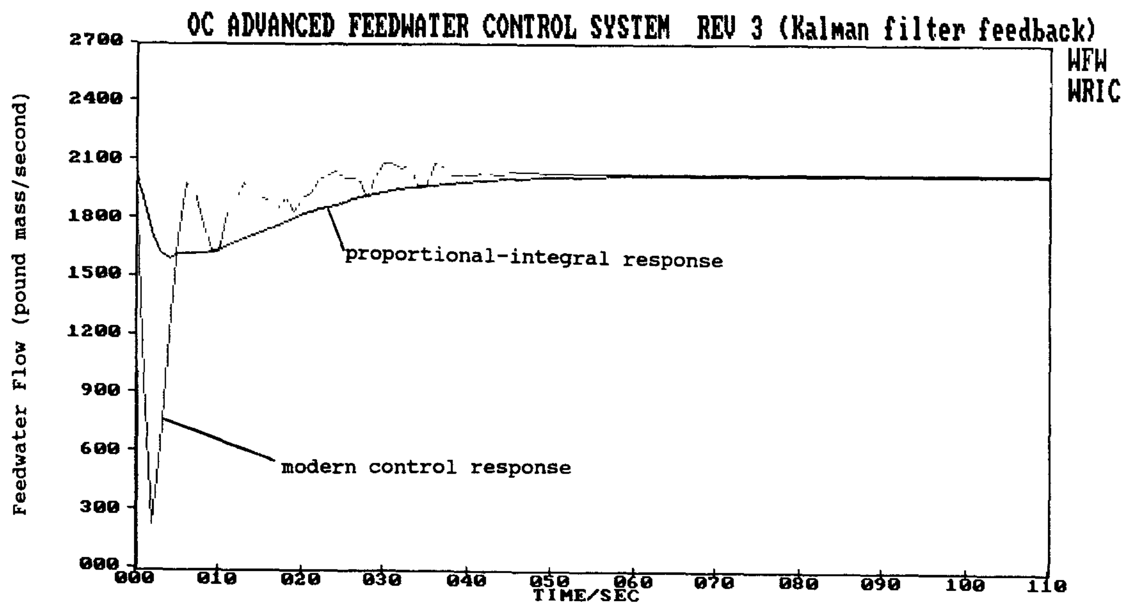
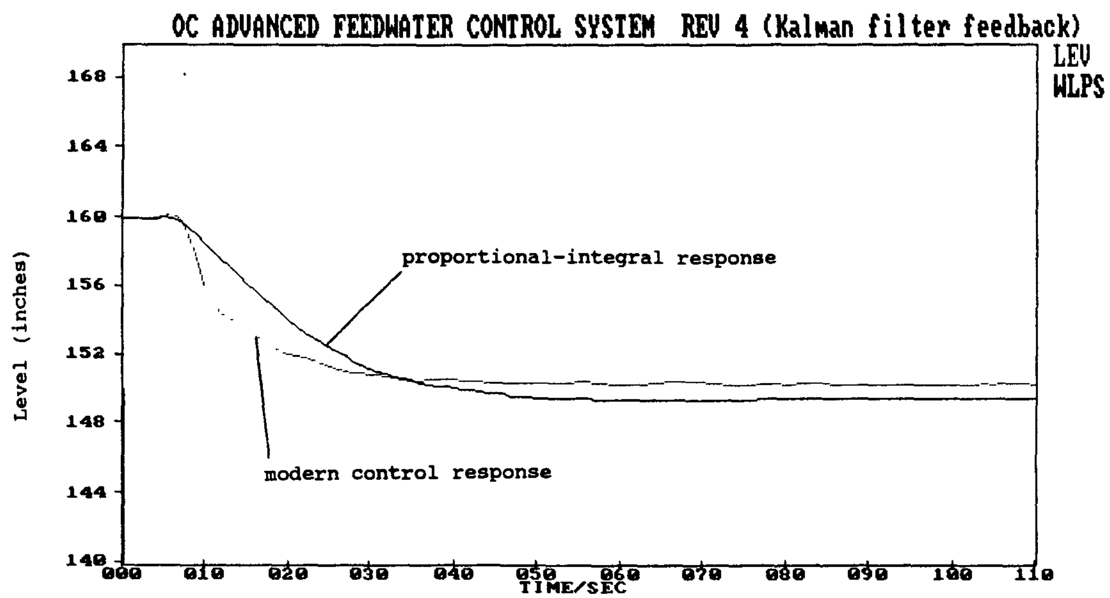


Figure 17 Comparison of Response With Backlash Added to Valve



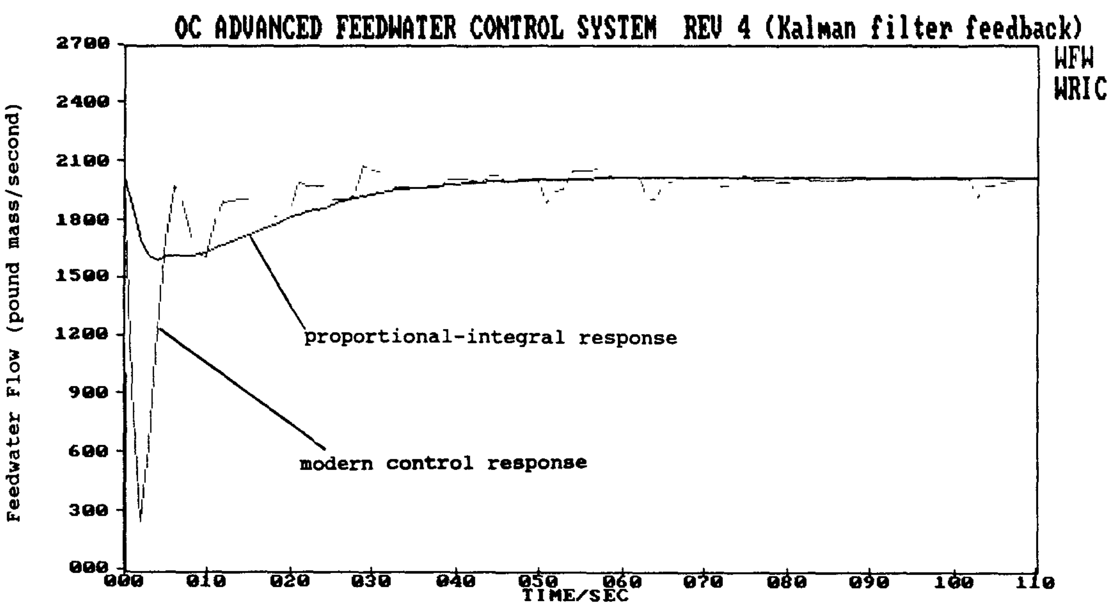
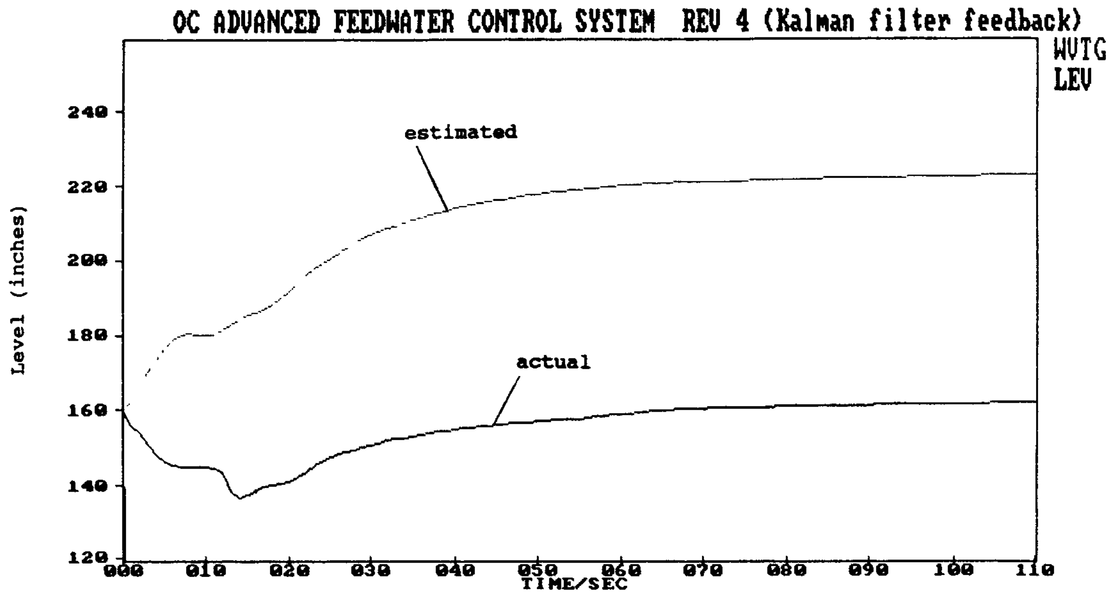


Figure 18 Main Steam Isolation-Comparison of Estimated States to Actual States



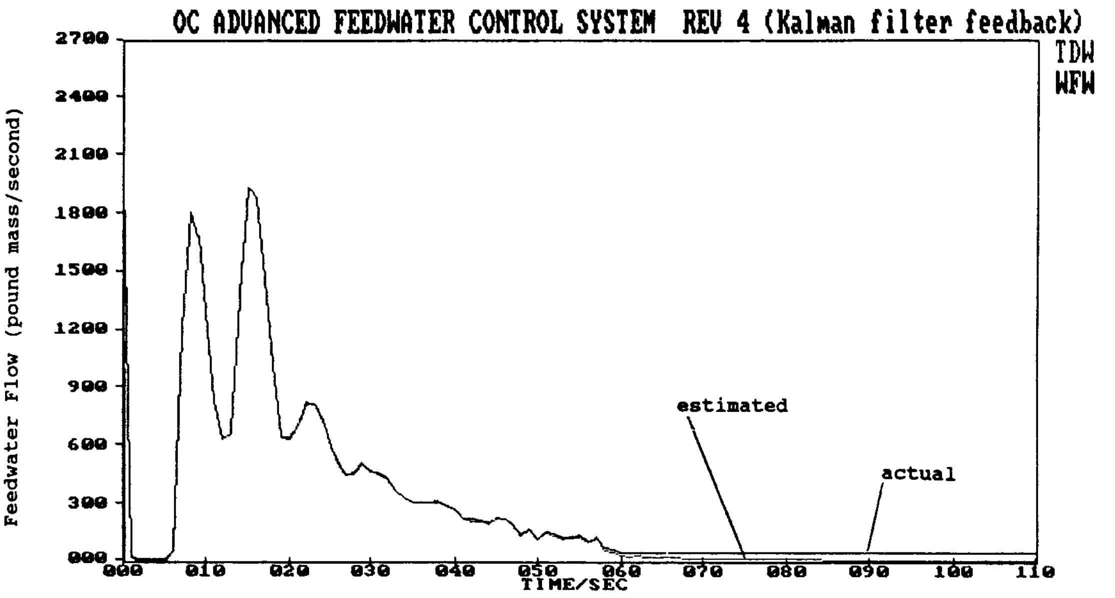
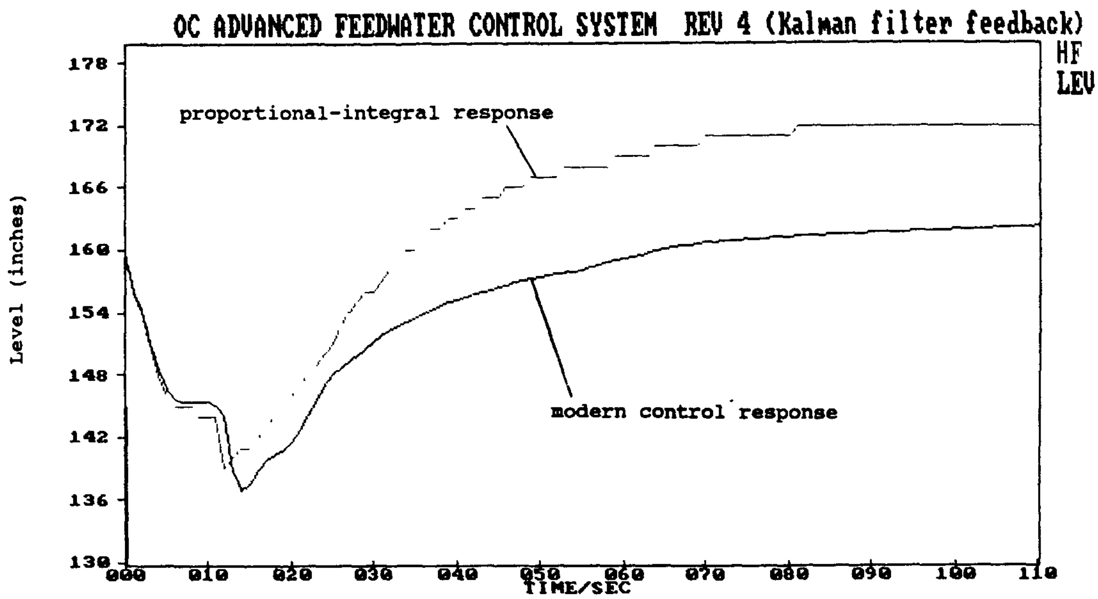


Figure 19 Main Steam Isolation-Comparison of Response



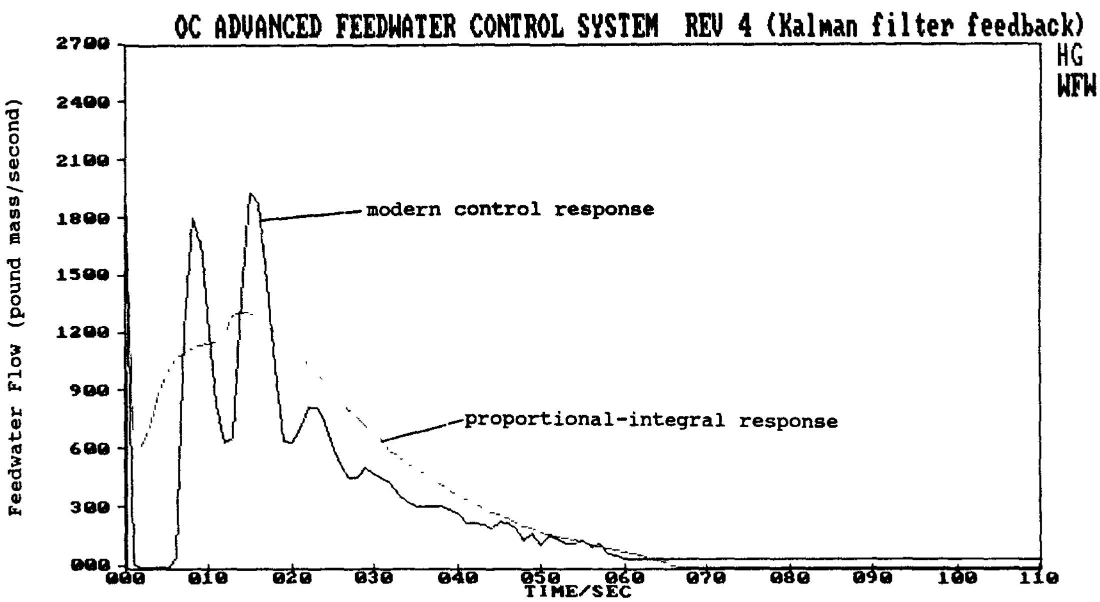
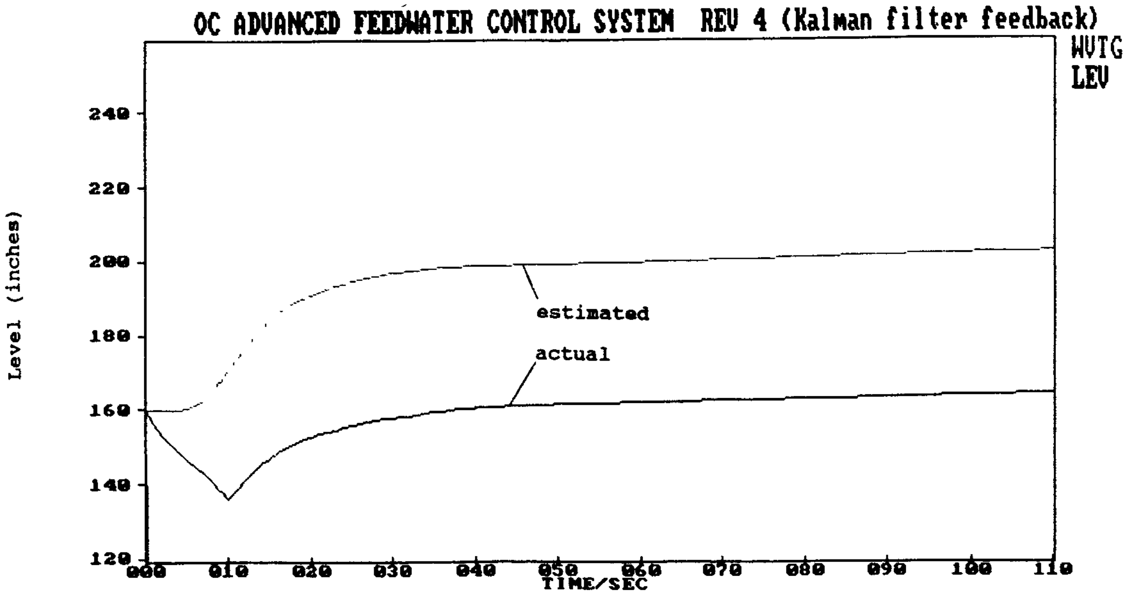


Figure 20 Reactor Trip-Comparison of Estimated States to Actual States



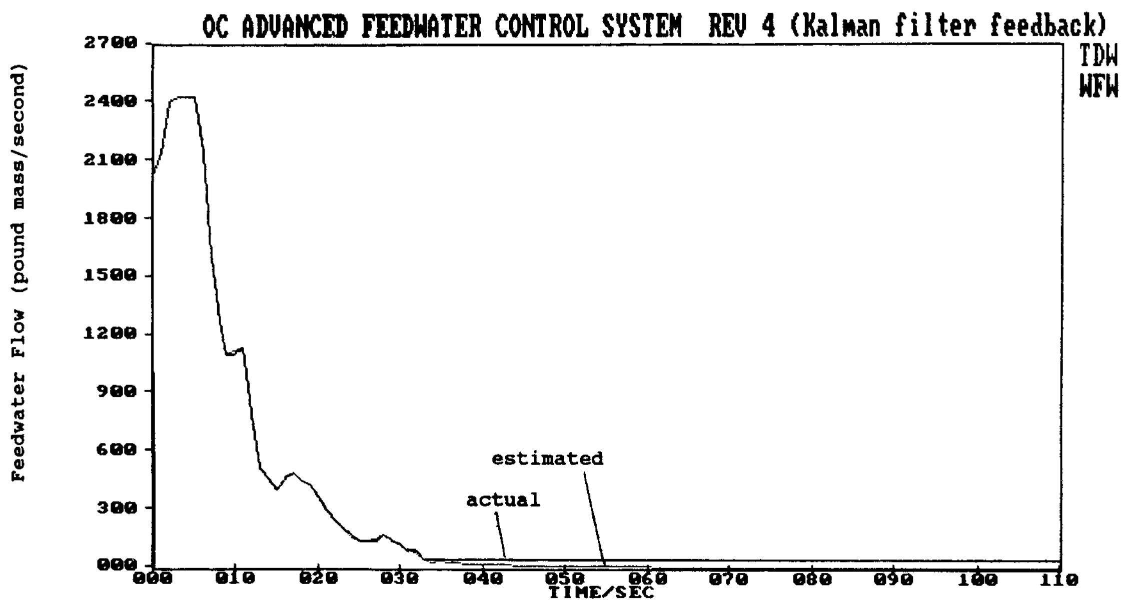
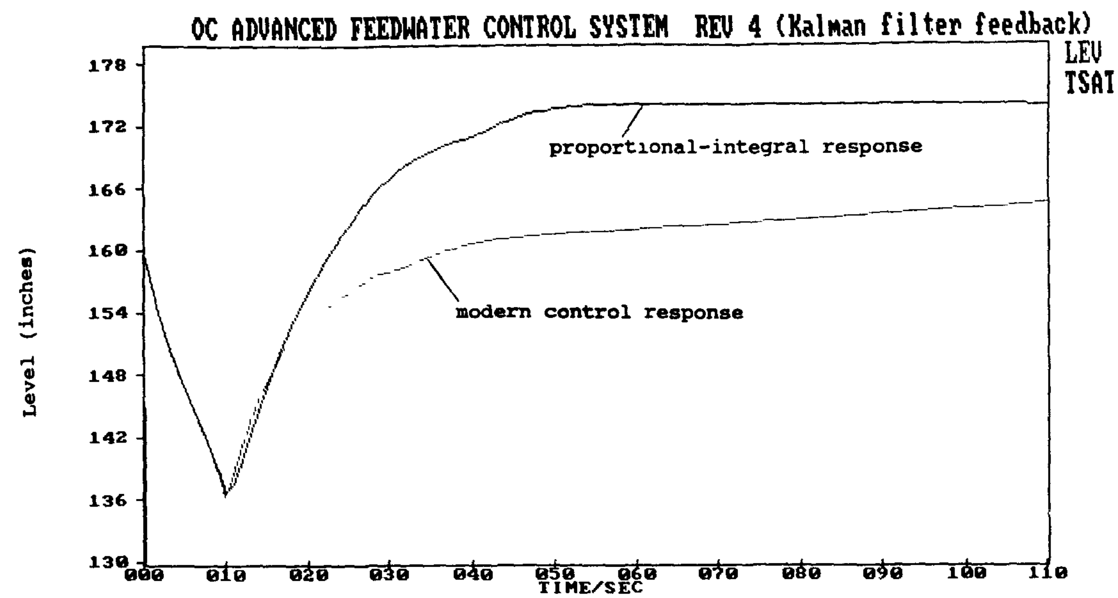
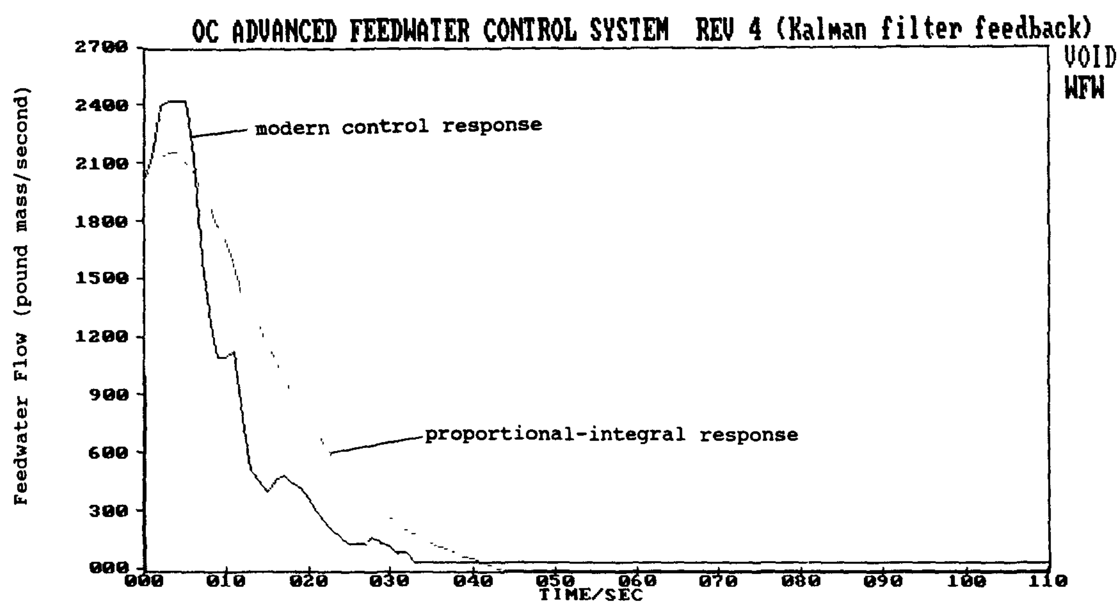


Figure 21 Reactor Trip-Comparison of Response





CHAPTER 5

CONCLUSIONS AND FOLLOW-ON WORK

5.1 Conclusions

This thesis shows that significant improvement is possible in performance by replacing a proportional-integral compensator with a modern control compensator for small (-10 inch step) and large (main steam isolation) transients. Implementation of the modern control compensator would therefore be likely to increase safety and plant availability. The following specific conclusions are drawn:

1. During a main steam isolation transient level is maintained closer to the setpoint of 160 inches by the modern control compensator than the proportional-integral compensator by as much as 10 inches. The same is true for a reactor trip.
2. During a -10 inch level setpoint step change the level response is significantly improved by the modern control compensator. Settling time is improved from 180 seconds with 6% overshoot to 34 seconds with no overshoot.
3. The addition of backlash to the valve results in significant degradation of perfromance of the proportional-integral compensator. Settling time is increased from 107 seconds with 3% overshoot to 180 seconds with 6% overshoot. However, degradation of the settling time for the modern control compensator is not observed and remains the same at 34 seconds. This shows

that the modern control compensator is more robust than the proportional-integral compensator for valve backlash variations.

4. Selecting parameters and gains for a small step transient, such as -10 inch level setpoint change, gives adequate response for large transients, and as a main steam isolation from 100% power, as well.

5. The use of the non-linear valve and control saturation limits in the linear Kalman filter works well as shown in the main steam isolation transinet where the valve output (feedwater flow) actual and estimate states are limited at zero flow. The same is also true at the high flow (120%) end for a reactor trip.

6. Even though the shrink and swell behavior of reactor water level is not modeled in the Kalman filter, the feedwater flow and steam flow observations compensate when estimated level is different from actual level.

7. The use of an accurate integration method is required for testing the modern control compensator. Also, the same method should be used globally.

8. The sensors and the setpoint station could be neglected thereby reducing the order of the Kalman filter from 12 to 8.

9. Using the MAXLIKE parameter estimation routine allows a more accurate design than would be achieved without parameter estimation.

5.2 Follow-On Work

Prior to implementation, the modern control compensator should be tuned for optimal response for all transients with the latest plant parameters. This would include making sure the latest valve parameters were incorporated. Also, the Q, R, V, and W matrices and the time step should be experimented with further to give the best response for all plant transients. An H-infinity formulation should be used to select the optimum Q and R matrices. Also, because PCTRANB has not been rigorously validated for Oyster Creek, the modern control compensator would be required to be tested for a wide range of transients using the basic principles simulator and/or the replica simulator which have been validated for Oyster Creek.

APPENDIX A

ALSIM Listings

```
#include "\ALSIM\ALSIM.H"

/*
** Advanced Feedwater Control Design Model (full state feedback rev4)
*/

#define g1      fpar[1]
#define g2      fpar[2]
#define g3      fpar[3]
#define g4      fpar[4]
#define g5      fpar[5]
#define go1     fpar[6]
#define go2     fpar[7]
#define go3     fpar[18]

#define kl      fpar[8]
#define kv      fpar[9]
#define tf      fpar[10]
#define tl      fpar[11]
#define ts      fpar[12]
#define tv      fpar[13]
#define td      fpar[14]
#define th      fpar[19]
#define zeta    fpar[20]
#define stlv    fpar[15]
#define xo1     fpar[16]
#define xo2     fpar[17]

#define x1dot  dxdt[1]
#define wfwdot dxdt[2]
#define xd1dot dxdt[3]
#define xd2dot dxdt[4]
#define levdot dxdt[5]
#define lvsnsdot  dxdt[6]
#define wfsnsdot  dxdt[7]
#define wssnsdot  dxdt[8]
#define stpntdot  dxdt[9]
derv(t, x, dxdt)
double t, *x, *dxdt;
{
/* Control Law */

    u[1] = -g1*x[1] - g2*x[2] - g3*x[3] - g4*x[4] - g5*x[5]
            - go1*xo1 - go2*xo2 - go3*stlv;

/* Plant Dynamics */

    x1dot = -((2*zeta)/tv)*x[1] - (1/(tv*tv))*x[2] + (kv/(tv*tv))*u[1];
    wfwdot = x[1];
    xd1dot = x[2] - (6/td)*x[3] - (12/(td*td))*x[4] + xo1;
    xd2dot = x[3];
    levdot = -((6*kl)/td)*x[3] + ((12*kl)/(td*td))*x[4] - kl*xo2;
    lvsnsdot = (1/tl)*x[5] - (1/tl)*x[6];
    wfsnsdot = (1/tf)*x[2] - (1/tf)*x[7] + (1/tf)*xo1;
    wssnsdot = -(1/ts)*x[8] + (1/ts)*xo2;
    stpntdot = -(1/th)*x[9] + (1/th)*stlv;
}

#include "\ALSIM\ALSIM.H"

/*
** Advanced Feedwater Control Design Model (full state feedback rev4)
*/

#define g1      fpar[1]
#define g2      fpar[2]
```

```
#define g3      fpar[3]
#define g4      fpar[4]
#define g5      fpar[5]
#define go1     fpar[6]
#define go2     fpar[7]
#define go3     fpar[18]

#define kl      fpar[8]
#define kv      fpar[9]
#define tf      fpar[10]
#define tl      fpar[11]
#define ts      fpar[12]
#define tv      fpar[13]
#define td      fpar[14]
#define th      fpar[19]
#define zeta    fpar[20]
#define stlv    fpar[15]
#define xo1     fpar[16]
#define xo2     fpar[17]

#define x1dot  dxdt[1]
#define wfwdot dxdt[2]
#define xd1dot dxdt[3]
#define xd2dot dxdt[4]
#define levdot dxdt[5]
#define lvsnsdot  dxdt[6]
#define wfsnsdot  dxdt[7]
#define wssnsdot  dxdt[8]
#define stpntdot  dxdt[9]
derv(t, x, dxdt)
double t, *x, *dxdt;
{
/* Control Law */

    u[1] = -g1*x[1] - g2*x[2] - g3*x[3] - g4*x[4] - g5*x[5]
           - go1*xo1 - go2*xo2 - go3*stlv;

/* Plant Dynamics */

    x1dot = -((2*zeta)/tv)*x[1] - (1/(tv*tv))*x[2] + (kv/(tv*tv))*u[1];
    wfwdot = x[1];
    xd1dot = x[2] - (6/td)*x[3] - (12/(td*td))*x[4] + xo1;
    xd2dot = x[3];
    levdot = -((6*kl)/td)*x[3] + ((12*kl)/(td*td))*x[4] - kl*xo2;
    lvsnsdot = (1/tl)*x[5] - (1/tl)*x[6];
    wfsnsdot = (1/tf)*x[2] - (1/tf)*x[7] + (1/tf)*xo1;
    wssnsdot = -(1/ts)*x[8] + (1/ts)*xo2;
    stpntdot = -(1/th)*x[9] + (1/th)*stlv;
}
```

```
0.0      ;initial time
100.0    ;final time
.100     ;maximum stepsize
.100     ;minimum stepsize
0.001    ;fractional error criterion

200      ;multiple of maximum stepsize for print output
10       ;multiple of maximum stepsize for plot output

9        ;number of plant states
1        ;number of plant inputs
0        ;number of plant outputs
0        ;number of controller states

0        ;size of user defined plot vector
0        ;size of user common area
0        ;size of gaussian random number vector
         ;vector multiplied by sqrt(hmax) to provide approx. uniform
         ;variance for variable stepsize
318      ;random number seed
272      ;random number seed
190      ;random number seed

0        ;number of user defined integer input parameters
0,0      ;end integer input parameters

20       ;number of user defined floating point input parameters
1,.0040       ;g1    R=0.1  Q(5,5)=1
2,.0250       ;g2
3,.0802       ;g3
4,.1297       ;g4
5,3.163       ;g5
6,.2052       ;go1
7,-.9563      ;go2
8,0.0260      ;kl
9,5.546       ;kv
10,0.2500     ;tf
11,1.000      ;tl
12,0.2500     ;ts
13,0.2292     ;tv
14,8.337      ;td
15,-10        ;stlv
16,0          ;xo1
17,0          ;xo2
18,-3.163     ;go3
19,.25        ;th
20,.6952      ;zeta
0,0      ;end floating point input parameters

0,0      ;end plant initial conditions

0,0      ;end controller initial conditions
```

```
#include "\ALSIM\ALSIM.H"

/*
** Advanced Feedwater Control Design Model (rev 4)
*/

#define g1      fpar[1]
#define g2      fpar[2]
#define g3      fpar[3]
#define g4      fpar[4]
#define g5      fpar[5]
#define go1     fpar[6]
#define go2     fpar[7]
#define go3     fpar[48]

#define kl      fpar[8]
#define kv      fpar[9]
#define tf      fpar[10]
#define tl      fpar[11]
#define ts      fpar[12]
#define tv      fpar[13]
#define td      fpar[14]
#define th      fpar[49]
#define zeta    fpar[56]

#define k11     fpar[15]
#define k12     fpar[16]
#define k13     fpar[17]
#define k21     fpar[18]
#define k22     fpar[19]
#define k23     fpar[20]
#define k31     fpar[21]
#define k32     fpar[22]
#define k33     fpar[23]
#define k41     fpar[24]
#define k42     fpar[25]
#define k43     fpar[26]
#define k51     fpar[27]
#define k52     fpar[28]
#define k53     fpar[29]
#define k61     fpar[30]
#define k62     fpar[31]
#define k63     fpar[32]
#define k71     fpar[33]
#define k72     fpar[34]
#define k73     fpar[35]
#define k81     fpar[36]
#define k82     fpar[37]
#define k83     fpar[38]
#define k91     fpar[50]
#define k92     fpar[51]
#define k93     fpar[52]
#define ko11    fpar[39]
#define ko12    fpar[40]
#define ko13    fpar[41]
#define ko21    fpar[42]
#define ko22    fpar[43]
#define ko23    fpar[44]
#define ko31    fpar[53]
#define ko32    fpar[54]
#define ko33    fpar[55]

#define stlv    fpar[45]
#define xo1     fpar[46]
#define xo2     fpar[47]
```

```
#define x1dot  dxdt[1]
#define x2dot  dxdt[2]
#define x3dot  dxdt[3]
#define x4dot  dxdt[4]
#define x5dot  dxdt[5]
#define x6dot  dxdt[6]
#define x7dot  dxdt[7]
#define x8dot  dxdt[8]
#define x9dot  dxdt[9]

#define x1hat  x[10]
#define x2hat  x[11]
#define x3hat  x[12]
#define x4hat  x[13]
#define x5hat  x[14]
#define x6hat  x[15]
#define x7hat  x[16]
#define x8hat  x[17]
#define x9hat  x[18]
#define xo1hat x[19]
#define xo2hat x[20]
#define xo3hat x[21]

#define x1dothat  dxdt[10]
#define x2dothat  dxdt[11]
#define x3dothat  dxdt[12]
#define x4dothat  dxdt[13]
#define x5dothat  dxdt[14]
#define x6dothat  dxdt[15]
#define x7dothat  dxdt[16]
#define x8dothat  dxdt[17]
#define x9dothat  dxdt[18]
#define xo1dothat dxdt[19]
#define xo2dothat dxdt[20]
#define xo3dothat dxdt[21]

derv(t, x, dxdt)
double t, *x, *dxdt;
{
double r1, r2, r3;


/* Control Law */

    u[1] = -g1*x1hat - g2*x2hat - g3*x3hat - g4*x4hat - g5*x5hat
           - go1*xo1hat - go2*xo2hat - go3*xo3hat;


/* Plant Dynamics */

    x1dot = -((2*zeta)/tv)*x[1] - (1/(tv*tv))*x[2] + (kv/(tv*tv))*u[1];
    x2dot = x[1];
    x3dot = x[2] - (6/td)*x[3] - (12/(td*td))*x[4] + xo1;
    x4dot = x[3];
    x5dot = -((6*kl)/td)*x[3] + ((12*kl)/(td*td))*x[4] - kl*xo2;
    x6dot = (1/tl)*x[5] - (1/tl)*x[6];
    x7dot = (1/tf)*x[2] - (1/tf)*x[7] + (1/tf)*xo1;
    x8dot = -(1/ts)*x[8] + (1/ts)*xo2;
    x9dot = -(1/th)*x[9] + (1/th)*stlv;

/* Observations */

    y[1] = x[6] - x[9];
    y[2] = x[7];
    y[3] = x[8];
```

```
/* Kalman Filter */

r1 = y[1] - x6hat + x9hat;
r2 = y[2] - x7hat;
r3 = y[3] - x8hat;

x1dothat = -((2*zeta)/tv)*x1hat - (1/(tv*tv))*x2hat + (kv/(tv*tv))*u[1]
            + k13*r1 + k12*r2 + k13*r3;
x2dothat = x1hat + k21*r1 + k22*r2 + k23*r3;
x3dothat = x2hat - (6/td)*x3hat - (12/(td*td))*x4hat + xo1hat + k31*r1
            + k32*r2 + k33*r3;
x4dothat = x3hat + k41*r1 + k42*r2 + k43*r3;
x5dothat = -((6*kl)/td)*x3hat + (12*kl/(td*td))*x4hat - kl*xo2hat + k51*r1
            + k52*r2 + k53*r3;
x6dothat = (1/tl)*x5hat - (1/tl)*x6hat + k61*r1 + k62*r2 + k63*r3;
x7dothat = (1/tf)*x2hat - (1/tf)*x7hat + (1/tf)*xo1hat + k71*r1
            + k72*r2 + k73*r3;
x8dothat = -(1/ts)*x8hat + (1/ts)*xo2hat + k81*r1 + k82*r2 + k83*r3;
x9dothat = -(1/th)*x9hat + (1/th)*xo3hat + k91*r1 + k92*r2 + k93*r3;
xo1dothat= ko11*r1 + ko12*r2 + ko13*r3;
xo2dothat= ko21*r1 + ko22*r2 + ko23*r3;
xo3dothat= ko31*r1 + ko32*r2 + ko33*r3;

}
```

```
0.0      ;initial time
100.0    ;final time
.100     ;maximum stepsize
.100     ;minimum stepsize
0.001    ;fractional error criterion

200      ;multiple of maximum stepsize for print output
10       ;multiple of maximum stepsize for plot output

21       ;number of plant states
1        ;number of plant inputs
4        ;number of plant outputs
0        ;number of controller states

0        ;size of user defined plot vector
0        ;size of user common area
0        ;size of gaussian random number vector
         ;vector multiplied by sqrt(hmax) to provide approx. uniform
         ;variance for variable stepsize
318      ;random number seed
272      ;random number seed
190      ;random number seed

0        ;number of user defined integer input parameters
0,0      ;end integer input parameters

56       ;number of user defined floating point input parameters
1,.0040       ;g1    R=0.1  Q(5,5)=1
2,.0250       ;g2
3,.0802       ;g3
4,.1297       ;g4
5,3.163       ;g5
6,.2052       ;go1
7,-.9563      ;go2
8,.0260       ;kl
9,5.546       ;kv
10,0.2500     ;tf
11,1.000      ;tl
12,0.2500     ;ts
13,0.2292     ;tv
14,8.337      ;td
15,.0037      ;k11   V(1,1)=10000  V(10,10)=V(11,11)=V(12,12)=1  W=identity
16,53.80      ;k12
17,.0002      ;k13
18,.0002      ;k21
19,20.21      ;k22
20,-.0002     ;k23
21,-.0053     ;k31
22,3.183      ;k32
23,0.         ;k33
24,.0040      ;k41
25,.2648      ;k42
26,0.         ;k43
27,.0016      ;k51
28,-.0080     ;k52
29,-.0322     ;k53
30,.0012      ;k61
31,-.0010     ;k62
32,-.0167     ;k63
33,0.         ;k71
34,9.263      ;k72
35,0.         ;k73
36,-.0082     ;k81
37,-.0002     ;k82
38,.8990      ;k83
39,.0009      ;ko11
```

```
40,.9900      ;ko12
41,.0002      ;ko13
42,-.0082     ;ko21
43,-.0001     ;ko22
44,1.000      ;ko23
45,-10        ;stlv
46,0          ;xo1
47,0          ;xo2
48,-3.163     ;go3
49,.25        ;th
50,-.7991     ;k91
51,-.0028     ;k92
52,-.0082     ;k93
53,-.9991     ;ko31
54,-.0027     ;ko32
55,-.0085     ;ko33
56,.6952      ;zeta
0,0      ;end floating point input parameters

0,0      ;end plant initial conditions

0,0      ;end controller initial conditions


0.0      ;initial time
100.0    ;final time
.100     ;maximum stepsize
.100     ;minimum stepsize
0.001    ;fractional error criterion

200      ;multiple of maximum stepsize for print output
10       ;multiple of maximum stepsize for plot output

21       ;number of plant states
1        ;number of plant inputs
4        ;number of plant outputs
0        ;number of controller states

0        ;size of user defined plot vector
0        ;size of user common area
0        ;size of gaussian random number vector
         ;vector multiplied by sqrt(hmax) to provide approx. uniform
         ;variance for variable stepsize
318      ;random number seed
272      ;random number seed
190      ;random number seed

0        ;number of user defined integer input parameters
0,0      ;end integer input parameters

56       ;number of user defined floating point input parameters
1,.0040       ;g1    R=0.1  Q(5,5)=1
2,.0250       ;g2
3,.0802       ;g3
4,.1297       ;g4
5,3.163       ;g5
6,.2052       ;go1
7,-.9563      ;go2
8,.0260       ;kl
9,5.546       ;kv
10,0.2500     ;tf
11,1.000      ;tl
12,0.2500     ;ts
13,0.2292     ;tv
14,8.337      ;td
15,.0037      ;k11   V(1,1)=10000  V(10,10)=V(11,11)=V(12,12)=1  W=identity
```

```
16,53.80      ;k12
17,.0002      ;k13
18,.0002      ;k21
19,20.21      ;k22
20,-.0002     ;k23
21,-.0053     ;k31
22,3.183      ;k32
23,0.         ;k33
24,.0040      ;k41
25,.2648      ;k42
26,0.         ;k43
27,.0016      ;k51
28,-.0080     ;k52
29,-.0322     ;k53
30,.0012      ;k61
31,-.0010     ;k62
32,-.0167     ;k63
33,0.         ;k71
34,9.263      ;k72
35,0.         ;k73
36,-.0082     ;k81
37,-.0002     ;k82
38,.8990      ;k83
39,.0009      ;ko11
40,.9900      ;ko12
41,.0002      ;ko13
42,-.0082     ;ko21
43,-.0001     ;ko22
44,1.000      ;ko23
45,-10        ;stlv
46,0          ;xo1
47,0          ;xo2
48,-3.163     ;go3
49,.25        ;th
50,-.7991     ;k91
51,-.0028     ;k92
52,-.0082     ;k93
53,-.9991     ;ko31
54,-.0027     ;ko32
55,-.0085     ;ko33
56,.6952      ;zeta
0,0     ;end floating point input parameters

0,0     ;end plant initial conditions

0,0     ;end controller initial conditions
```

APPENDIX B

PCTRANB Original Plant Feedwater Control System Listing

```
C---INITIALIZE FEEDWATER CONTROL SYSTEM VARIABLES--REV2------------
         WFW0=POWER*3.413E6/3600./(HG-HFW)
         WFW=WFW0*PF
         WSTM=WFW
         CINT=1./(GFW*5.546)
         XX1=0.
         XX2=WFW0
         XD1=0.
         XD2=0.
         XD3=0.
         XD4=WFW0/.6482
         XX6=LEV0
         XX7=WFW0
         XX8=WSTM
C INITIALIZE MEASURED TEST DATA
         RXLVL=160.0
         FDFLO=1.00
         SMFLO=1.00
         RXPWR=100.0
C INITIALIZE HEADER PRESSURE REGULATOR
         PDOT1=0.
         PDOT2=.5000
C-------------------------------------------------------------------
```

```
C ----------------------------------------------------------------
C OYSTER CREEK ORIGINAL PLANT FEEDWATER CONTROL SYSTEM REV 2
C ----------------------------------------------------------------
C       LEVEL SETPOINT
              STLV=LEV0+(XLV-LEV0)*STEP(TIME-TXLV)
C       STEAM-FEED MISMATCH
              BLEV=GMSMH*(XX7-XX8)/WFW0*SPLV+XX6
C       PI CONTROLLER
              ERR=(STLV-BLEV)/SPLV
C REV2     ERROR LIMIT (18%)
              IF(ERR.LE.-.18)THEN
                   ERR=-.18
              ELSEIF(ERR.GT..18)THEN
                   ERR=.18
              ENDIF
              CINT=CINT+DT*(ERR/TIFW)
              CNTL=GFW*WFW0*(CINT+ERR)
C       MAIN FEEDWATER REGULATING VALVE (kv=5.546 zeta=.6952 tv=.2292)
              VLVERR=105.6*CNTL-6.066*XX1-19.03*WFW
              XX1=XX1+DT*VLVERR
C REV2 ADD BACKLASH (WIDTH= .6)
              IF(ABS(XX1).LE.150)THEN
                   BCKLSV=0.
              ELSEIF(ABS(XX1).GT.150)THEN
                   BCKLSV=XX1
              ENDIF
              WFW=WFW+DT*BCKLSV
C       VALVE SATURATION
              IF(WFW.LT.0)THEN
                   WFW=0.
                   CNTL=0.
              ELSEIF(WFW.GT.FWMX*WFW0)THEN
                   WFW=FWMX*WFW0
              ENDIF
C       DELAY FOR MASS TRANSPORT (FOURTH ORDER PADE APPROXIMATION:td=6sec)
              XD1=XD1+DT*(-2.667*XD1-3.333*XD2-2.222*XD3-.64815*XD4+WFW)
              XD2=XD2+DT*XD1
              XD3=XD3+DT*XD2
              XD4=XD4+DT*XD3
              WFWDLY=(-.6667*XD1+1.667*XD2-1.667*XD3+.64815*XD4)
C       LEVEL SENSOR
              XX6=XX6+DT*(LEV-XX6)
C       FEEDWATER FLOW SENSOR
              XX7=XX7+DT*(WFW-XX7)/.25
C       STEAM FLOW SENSOR
              XX8=XX8+DT*(WSTM-XX8)/.25
```

```
C      PRESSURE REGULATOR (STATE SPACE MODEL) 10
               PERR=(PHDR-PST)/30.000+1.000
               PDOT1=PDOT1+DT*(.3922*PERR-6.7843*PDOT1-.7843*PDOT2)
               PDOT2=PDOT2+DT*PDOT1
               WSTM=WFW0*(PDOT1+2.00*PDOT2)
               IF(WSTM.LT.0)THEN
                    WSTM=0.
               ELSEIF(WSTM.GT.FWMX*WFW0)THEN
                    WSTM=FWMX*WFW0
                ENDIF
```

```
\OYSTER CREEK ORIGINAL PLANT FEEDWATER CONTROL SYSTEM REV 2 \
POWER     VSAT      VRCS      SSMS      UO2MS     HCOR      WCORO     ALDC
 1930.000 10400.000 3050.000   400.000   100.000    12.000 26000.000    10.000
    PHIGH      LLOW      VSTM        L2        L8        L1      ASHD      HMDW
 1061.000   137.000  7794.000    86.000   175.000    56.000   220.000     1.000
     SIZE       LBK       CD1       CD2       TBV        CO       VGJ      HMWW
  0.0E+0     60.000     1.000     1.000     0.400     1.000     1.000     1.000
     LEVO        PF       HFW     TECCS       ANS       TFW        PO      PHDR
  160.000     1.000   286.900    96.000     1.000  9999.000  1035.000   975.000
     SRV1      SRV2     PSRV1    PSRV1P     PSRV2    PSRV2P      PISO      TLAG
    0.253     0.429  1075.000  1025.000  1095.000  1073.000   856.700     1.000
     IRUN      TRUN        DT      TINT      TPRT       TLD       TLG
        0  5000.000     0.010     0.000     1.000     1.000     1.000
     TADS     THPCS     TLPCI     TLPCS     TRCIC     TLOFW     TMSIV
99999.000 99999.000 99999.000  9999.000 99999.000 99999.000 99999.000
     IADS      ISTM       IFW     IRCIC     IHPCS     ILPCI     ILPCS
        0         0         0         2         2         2         2
        J         K     ICORE      ISAT      IRCA     IHPCI     ITBTP
        0         0         0         0         0         2         0
    TRCPA       CDF     TRCPB      FCRC      PWNC      FCNC    TSCRAM
99999.000     0.950 99999.000     1.000     0.440     0.280  9999.000
    GMSMH      FWMX      RCIC      SPDA      SPDB      TDLY      TADY
    0.375     1.200    69.000   100.000   100.000     2.000   120.000
     RCLP     THPCI       CVF       CLV     SPDSP      WCRF      HIFX
    5.000 99999.999     -.500     0.100    25.000     1.000     1.180
     ALDA       BOL       DNT      FLFR      TBTP      TRCT      RHEX
   76.700   150.000     0.001     0.300 99999.000     0.000     0.000
   PHI(1)    PHI(2)    PHI(3)    PHI(4)    PHI(5)    PHI(6)    PHI(7)
   14.700   214.700  1175.000  1178.000  1300.000  1400.000  1500.000
   WHI(1)    WHI(2)    WHI(3)    WHI(4)    WHI(5)    WHI(6)    WHI(7)
    5.000     5.000     5.000     5.000     5.000     0.000     0.000
   PHS(1)    PHS(2)    PHS(3)    PHS(4)    PHS(5)    PHS(6)    PHS(7)
   14.700   177.400   448.700   701.200  1143.200  1186.700  1208.300
   WHS(1)    WHS(2)    WHS(3)    WHS(4)    WHS(5)    WHS(6)    WHS(7)
  750.300   680.900   555.800   416.900   201.500    76.400     0.000
   PLS(1)    PLS(2)    PLS(3)    PLS(4)    PLS(5)    PLS(6)    PLS(7)
   14.700    66.800   131.900   188.300   249.100   305.500   361.900
   WLS(1)    WLS(2)    WLS(3)    WLS(4)    WLS(5)    WLS(6)    WLS(7)
  833.700   694.800   555.800   416.800   277.900   139.000     0.000
   PLI(1)    PLI(2)    PLI(3)    PLI(4)    PLI(5)    PLI(6)    PLI(7)
   14.700    32.100    79.800   121.900   162.300   205.700   249.100
   WLI(1)    WLI(2)    WLI(3)    WLI(4)    WLI(5)    WLI(6)    WLI(7)
  764.200   694.800   555.800   416.800   277.900     0.000     0.000
    QI(1)     QI(2)     QI(3)     QI(4)     QI(5)     QI(6)     QI(7)
    0.000     0.020     0.080     0.200     0.300     0.440     1.001
    WI(1)     WI(2)     WI(3)     WI(4)     WI(5)     WI(6)     WI(7)
    0.000     0.100     0.200     0.240     0.250     0.270     1.001
     ISLB      ICSP      IRCB      PCSP      FCSP      GCSP      TCSP
        0         0         0    16.900     0.950  3000.000 99999.000
     TDWO      PDWO      LDWO       ADW       VDW      PDSN      PFCL
  135.000    16.200     3.300  2000.000169000.000    60.000   999.700
     TFCL      QCLO      TDSN      RLKO      DTVT      TWWO      PWWO
99999.000     5.000   350.000     0.500     8.000    96.000    16.200
     LWWO       AWW       VWW      DPVB      DTVB      USTC       TFO
   14.000  8714.000260000.000     0.100    40.000     0.050  1200.000
      GFW      GSTM      TIFW      TIST      PTBV      SPLV      PSPN
    1.250     3.300   150.000 99999.000  1015.000    96.000   100.000
       P1       TP1      XLV1      TXLV       HW1      THW1      AKDP
  975.000     0.000   160.000     0.000   338.000 99999.000    -1.E-3
     CITS      BRTH       XOQ       EPS      AISP      ARBN       FAS
    0.200   3.47E-4   8.58E-4     0.000     3.270     0.250   444.000
     I131      I132      I133      I135     KR85M      KR85      KR87
   86.640   126.620   181.180   171.070    22.160     0.990    42.540
     KR88    XE133M     XE133     XE135     CS137
   60.260     7.570   182.030    23.540    10.930
        X       WCU         P         V        VR        MT     WJDRA
```

TM	WCOR	WFWO	WJDRB	LEV	TFSB	TFPK	
QN	WSTM	WVP	RH	SUM	VOIDO	WFW	
PDW	TDW	LDW	PWW	TWW	LWW	TPCT	
ADWF	ADWG	ADWA	UF	TF	TLK	PERRO	DWFWO

APPENDIX C

PCTRANB Advanced Feedwater Control System Listings

```
C---INITIALIZE ADVANCED FEEDWATER CONTROL SYSTEM VARIABLES REV4-
        WFW0=POWER*3.413E6/3600./(HG-HFW)
        WFW=WFW0*PF
        WSTM=WFW
C   ADD BACKLASH REV4
        BCKLSH=WFW0/5.546
        XX1=0.
        XD1=0.
        XD2=0.
        XD3=0.
        XD4=WFW0/.6482
        XX6=LEV0
        XX7=WFW0
        XX8=WFW0
        XX9=LEV0
        X1HAT=0.
        X2HAT=100.
        X3HAT=0.
        X4HAT=100./.1726
        X5HAT=LEV0
        X6HAT=LEV0
        X7HAT=100.
        X8HAT=100.
        X9HAT=LEV0
        XO1HAT=0.
        XO2HAT=100.
        XO3HAT=LEV0
C INITIALIZE HEADER PRESSURE REGULATOR
        PDOT1=0.
        PDOT2=.5000
C INITIALIZE MEASURED TEST DATA
        RXLVL=160.0
        FDFLO=1.00
        SMFLO=1.00
        RXPWR=100.0
C INITIALIZE PI COMPENSATOR DATA
        PILVL=160.0
        PIFLO=2023.
C INITIALIZE PI COMPENSATOR(W/BACKLASH) DATA
        BILVL=160.0
        BIFLO=2022.
```

```
C-----------------------------------------------------------------------
C OYSTER CREEK ADVANCED FEEDWATER CONTROL SYSTEM REV 4  R=.1 V(5,5)=0
C-----------------------------------------------------------------------
C       LEVEL SETPOINT
               STLV=LEV0+(XLV-LEV0)*STEP(TIME-TXLV)
C----------------------
C       CONTROL LAW
C----------------------
               LQF=-.0250*X2HAT-.1297*X4HAT
               LQD=-.0040*X1HAT-.0802*X3HAT
               LQL=-3.163*X5HAT
               EXG=-.2052*XO1HAT+.9563*XO2HAT+3.163*XO3HAT
               CNTL=(LQF+LQD+LQL+EXG)*WFW0/100.
C       CONTROLLER SATURATION
               IF(CNTL.LT.0.)THEN
                    CNTL=0.
               ELSEIF(CNTL.GT.FWMX*WFW0)THEN
                    CNTL=FWMX*WFW0
               ENDIF
C       MAIN FEEDWATER REGULATING VALVE (kv=5.546 zeta=.6952 tv=.2292)
               XX1=XX1+DT*(105.6*CNTL-6.066*XX1-19.03*WFW)
C REV4   ADD BACKLASH (WIDTH=.6)
               IF(ABS(XX1).LE.150)THEN
                    BCKLSV=0.
               ELSEIF(ABS(XX1).GT.150)THEN
                    BCKLSV=XX1
               ENDIF
               WFW=WFW+DT*BCKLSV
C       VALVE SATURATION
               IF(WFW.LT.0.)THEN
                    WFW=0.
               ELSEIF(WFW.GT.FWMX*WFW0)THEN
                    WFW=FWMX*WFW0
               ENDIF
C       DELAY FOR FW ENTHALPY (FOURTH ORDER PADE APPROXIMATION:td=6sec)
               XD1=XD1+DT*(-2.667*XD1-3.333*XD2-2.222*XD3-.6482*XD4+WFW)
               XD2=XD2+DT*XD1
               XD3=XD3+DT*XD2
               XD4=XD4+DT*XD3
               WFWDLY=(-.6667*XD1+1.667*XD2-1.667*XD3+.6482*XD4)
C       LEVEL SENSOR
               XX6=XX6+DT*(LEV-XX6)
C       FEEDWATER FLOW SENSOR
               XX7=XX7+DT*(WFW-XX7)/.25
C       STEAM FLOW SENSOR
               XX8=XX8+DT*(WSTM-XX8)/.25
C       LEVEL SETPOINT STATION
               XX9=XX9+DT*(STLV-XX9)/.25
C------OBSERVATIONS
               Y1=XX6-XX9
               Y2=XX7*100./WFW0
               Y3=XX8*100./WFW0
```

```
C-----------------------------------------------------------------|
C     KALMAN FILTER INCLUDING NON-LINEAR VALVE SATURATION          |
C-----------------------------------------------------------------|
          R1=Y1-X6HAT+X9HAT
          R2=Y2-X7HAT
          R3=Y3-X8HAT
          CTRL=CNTL*100./WFW0
C
          RES1=.0037*R1+53.80*R2+.0002*R3
          X1HAT=X1HAT+DT*(-6.660*X1HAT-19.03*X2HAT+105.6*CTRL)
          RES2=.0002*R1+20.21*R2-.0002*R3
          X2HAT=X2HAT+DT*(X1HAT+RES2)
C
C       VALVE SATURATION
          IF(X2HAT.LT.0.)THEN
              X2HAT=0.
          ELSEIF(X2HAT.GT.FWMX*100.)THEN
              X2HAT=FWMX*100.
          ENDIF
C
          RES3=-.0053*R1+3.183*R2
          X3HAT=X3HAT+DT*(X2HAT-.7197*X3HAT-.1726*X4HAT+XO1HAT+RES3)
          RES4=.0040*R1+.2648*R2
          X4HAT=X4HAT+DT*(X3HAT+RES4)
          RES5=.0016*R1-.0080*R2-.0322*R3
          X5HATA=-.0187*X3HAT+.0045*X4HAT-.0260*XO2HAT
          X5HAT=X5HAT+DT*(X5HATA+RES5)
          RES6=.0012*R1-.0010*R2-.0167*R3
          X6HAT=X6HAT+DT*(X5HAT-X6HAT+RES6)
          RES7=9.623*R2
          X7HAT=X7HAT+DT*(4*(X2HAT-X7HAT+XO1HAT)+RES7)
          RES8=-.0082*R1-.0002*R2+.8990*R3
          X8HAT=X8HAT+DT*(4*(-X8HAT+XO2HAT)+RES8)
          RES9=-.7991*R1-.0028*R2-.0082*R3
          X9HAT=X9HAT+DT*(4*(-X9HAT+XO3HAT)+RES9)
          XO1HAT=XO1HAT+DT*(.0009*R1+.9900*R2+.0002*R3)
          XO2HAT=XO2HAT+DT*(-.0082*R1-.0001*R2+1.000*R3)
          XO3HAT=XO3HAT+DT*(-.9991*R1-.0027*R2-.0085*R3)
```

\OC ADVANCED FEEDWATER CONTROL SYSTEM REV 3 (Kalman filter feedback)\

POWER	VSAT	VRCS	SSMS	UO2MS	HCOR	WCORO	ALDC
1930.000	10400.000	3050.000	400.000	100.000	12.000	26000.000	10.000
PHIGH	LLOW	VSTM	L2	L8	L1	ASHD	HMDW
1061.000	137.000	7794.000	86.000	175.000	56.000	220.000	1.000
SIZE	LBK	CD1	CD2	TBV	CO	VGJ	HMWW
0.0E+0	60.000	1.000	1.000	0.400	1.000	1.000	1.000
LEVO	PF	HFW	TECCS	ANS	TFW	PO	PHDR
160.000	1.000	286.900	96.000	1.000	9999.000	1035.000	975.000
SRV1	SRV2	PSRV1	PSRV1P	PSRV2	PSRV2P	PISO	TLAG
0.253	0.429	1075.000	1025.000	1095.000	1073.000	856.700	1.000
IRUN	TRUN	DT	TINT	TPRT	TLD	TLG	
0	5000.000	1.0E-4	0.000	1.000	1.000	1.000	
TADS	THPCS	TLPCI	TLPCS	TRCIC	TLOFW	TMSIV	
99999.000	99999.000	99999.000	9999.000	99999.000	99999.000	99999.000	
IADS	ISTM	IFW	IRCIC	IHPCS	ILPCI	ILPCS	
0	0	0	2	2	2	2	
J	K	ICORE	ISAT	IRCA	IHPCI	ITBTP	
0	0	0	0	0	2	0	
TRCPA	CDF	TRCPB	FCRC	PWNC	FCNC	TSCRAM	
99999.000	0.950	99999.000	1.000	0.440	0.280	9999.000	
GMSMH	FWMX	RCIC	SPDA	SPDB	TDLY	TADY	
0.375	1.200	69.000	100.000	100.000	2.000	120.000	
RCLP	THPCI	CVF	CLV	SPDSP	WCRF	HIFX	
5.000	99999.999	-0.500	0.100	25.000	1.000	1.180	
ALDA	BOL	DNT	FLFR	TBTP	TRCT	RHEX	
76.700	150.000	0.001	0.300	99999.000	0.000	0.000	
PHI(1)	PHI(2)	PHI(3)	PHI(4)	PHI(5)	PHI(6)	PHI(7)	
14.700	214.700	1175.000	1178.000	1300.000	1400.000	1500.000	
WHI(1)	WHI(2)	WHI(3)	WHI(4)	WHI(5)	WHI(6)	WHI(7)	
5.000	5.000	5.000	5.000	5.000	0.000	0.000	
PHS(1)	PHS(2)	PHS(3)	PHS(4)	PHS(5)	PHS(6)	PHS(7)	
14.700	177.400	448.700	701.200	1143.200	1186.700	1208.300	
WHS(1)	WHS(2)	WHS(3)	WHS(4)	WHS(5)	WHS(6)	WHS(7)	
750.300	680.900	555.800	416.900	201.500	76.400	0.000	
PLS(1)	PLS(2)	PLS(3)	PLS(4)	PLS(5)	PLS(6)	PLS(7)	
14.700	66.800	131.900	188.300	249.100	305.500	361.900	
WLS(1)	WLS(2)	WLS(3)	WLS(4)	WLS(5)	WLS(6)	WLS(7)	
833.700	694.800	555.800	416.800	277.900	139.000	0.000	
PLI(1)	PLI(2)	PLI(3)	PLI(4)	PLI(5)	PLI(6)	PLI(7)	
14.700	32.100	79.800	121.900	162.300	205.700	249.100	
WLI(1)	WLI(2)	WLI(3)	WLI(4)	WLI(5)	WLI(6)	WLI(7)	
764.200	694.800	555.800	416.800	277.900	0.000	0.000	
QI(1)	QI(2)	QI(3)	QI(4)	QI(5)	QI(6)	QI(7)	
0.000	0.020	0.080	0.200	0.300	0.440	1.001	
WI(1)	WI(2)	WI(3)	WI(4)	WI(5)	WI(6)	WI(7)	
0.000	0.100	0.200	0.240	0.250	0.270	1.001	
ISLB	ICSP	IRCB	PCSP	FCSP	GCSP	TCSP	
0	0	0	16.900	0.950	3000.000	99999.000	
TDWO	PDWO	LDWO	ADW	VDW	PDSN	PFCL	
135.000	16.200	3.300	2000.000	169000.000	60.000	999.700	
TFCL	QCLO	TDSN	RLKO	DTVT	TWWO	PWWO	
99999.000	5.000	350.000	0.500	8.000	96.000	16.200	
LWWO	AWW	VWW	DPVB	DTVB	USTC	TFO	
14.000	8714.000	260000.000	0.100	40.000	0.050	1200.000	
GFW	GSTM	TIFW	TIST	PTBV	SPLV	PSPN	
1.250	3.300	150.000	99999.000	1015.000	96.000	100.000	
P1	TP1	XLV1	TXLV	HW1	THW1	AKDP	
975.000	0.000	160.000	0.000	338.000	99999.000	-1.E-3	
CITS	BRTH	XOQ	EPS	AISP	ARBN	FAS	
0.200	3.47E-4	8.58E-4	0.000	3.270	0.250	444.000	
I131	I132	I133	I135	KR85M	KR85	KR87	
86.640	126.620	181.180	171.070	22.160	0.990	42.540	
KR88	XE133M	XE133	XE135	CS137			
60.260	7.570	182.030	23.540	10.930			
X	WCU	P	V	VR	MT	WJDRA	

TM	WCOR	WFWO	WJDRB	LEV	TFSB	TFPK	
QN	WSTM	WVP	RH	SUM	VOIDO	WFW	
PDW	TDW	LDW	PWW	TWW	LWW	TPCT	
ADWF	ADWG	ADWA	UF	TF	TLK	PERRO	DWFWO

```
$STORAGE:2
$NOFLOATCALLS
C PCTRAN5.FOR
C  MEASURED TEST DATA: REACTOR WATER LEVEL(INCHES TAF)
      SUBROUTINE MSRDT1(TIME,RXLVL)
         IF(TIME.LE.6)THEN
              RXLVL=160.0
         ELSEIF(TIME.GT.6.AND.TIME.LE.7)THEN
              RXLVL=159.7
         ELSEIF(TIME.GT.7.AND.TIME.LE.8)THEN
              RXLVL=159.52
         ELSEIF(TIME.GT.8.AND.TIME.LE.9)THEN
              RXLVL=159.1
         ELSEIF(TIME.GT.9.AND.TIME.LE.10)THEN
              RXLVL=158.8
         ELSEIF(TIME.GT.10.AND.TIME.LE.11)THEN
              RXLVL=158.2
         ELSEIF(TIME.GT.11.AND.TIME.LE.12)THEN
              RXLVL=157.9
         ELSEIF(TIME.GT.12.AND.TIME.LE.13)THEN
              RXLVL=157.6
         ELSEIF(TIME.GT.13.AND.TIME.LE.14)THEN
              RXLVL=157.3
         ELSEIF(TIME.GT.14.AND.TIME.LE.15)THEN
              RXLVL=156.7
         ELSEIF(TIME.GT.15.AND.TIME.LE.16)THEN
              RXLVL=156.5
         ELSEIF(TIME.GT.16.AND.TIME.LE.17)THEN
              RXLVL=156.1
         ELSEIF(TIME.GT.17.AND.TIME.LE.18)THEN
              RXLVL=155.5
         ELSEIF(TIME.GT.18.AND.TIME.LE.19)THEN
              RXLVL=155.2
         ELSEIF(TIME.GT.19.AND.TIME.LE.20)THEN
              RXLVL=154.8
         ELSEIF(TIME.GT.20.AND.TIME.LE.21)THEN
              RXLVL=154.3
         ELSEIF(TIME.GT.21.AND.TIME.LE.22)THEN
              RXLVL=153.7
         ELSEIF(TIME.GT.22.AND.TIME.LE.23)THEN
              RXLVL=153.3
         ELSEIF(TIME.GT.23.AND.TIME.LE.24)THEN
              RXLVL=152.3
         ELSEIF(TIME.GT.24.AND.TIME.LE.25)THEN
              RXLVL=152.5
         ELSEIF(TIME.GT.25.AND.TIME.LE.26)THEN
              RXLVL=151.9
         ELSEIF(TIME.GT.26.AND.TIME.LE.27)THEN
              RXLVL=151.3
         ELSEIF(TIME.GT.27.AND.TIME.LE.28)THEN
              RXLVL=151.0
         ELSEIF(TIME.GT.28.AND.TIME.LE.29)THEN
              RXLVL=150.7
         ELSEIF(TIME.GT.29.AND.TIME.LE.30)THEN
              RXLVL=150.4
         ELSEIF(TIME.GT.30.AND.TIME.LE.31)THEN
              RXLVL=150.1
         ELSEIF((TIME.GT.31.AND.TIME.LE.32).OR.(TIME.GT.91.AND.TIME.LE.
     1 92))THEN
              RXLVL=149.9
         ELSEIF((TIME.GT.32.AND.TIME.LE.33).OR.(TIME.GT.90.AND.TIME.LE.
     1 91))THEN
              RXLVL=149.8
         ELSEIF((TIME.GT.33.AND.TIME.LE.34).OR.(TIME.GT.88.AND.TIME.LE.
     1 90))THEN
              RXLVL=149.7
```

```
         ELSEIF((TIME.GT.34.AND.TIME.LE.35).OR.(TIME.GT.87.AND.TIME.LE.
     1 88))THEN
              RXLVL=149.6
         ELSEIF((TIME.GT.35.AND.TIME.LE.36).OR.(TIME.GT.68.AND.TIME.LE.
     1   69))THEN
              RXLVL=149.2
         ELSEIF(TIME.GT.36.AND.TIME.LE.37)THEN
              RXLVL=148.9
         ELSEIF((TIME.GT.37.AND.TIME.LE.38).OR.(TIME.GT.60.AND.TIME.LE.
     1  61))THEN
              RXLVL=148.8
         ELSEIF((TIME.GT.38.AND.TIME.LE.39).OR.(TIME.GT.59.AND.TIME.LE.
     1   60))THEN
              RXLVL=148.7
         ELSEIF((TIME.GT.39.AND.TIME.LE.40).OR.(TIME.GT.58.AND.TIME.LE.
     1   59))THEN
              RXLVL=148.6
         ELSEIF((TIME.GT.40.AND.TIME.LE.41).OR.(TIME.GT.57.AND.TIME.LE.
     1   58))THEN
              RXLVL=148.5
         ELSEIF((TIME.GT.41.AND.TIME.LE.42).OR.(TIME.GT.55.AND.TIME.LE.
     1   57))THEN
              RXLVL=148.4
         ELSEIF((TIME.GT.42.AND.TIME.LE.43).OR.(TIME.GT.54.AND.TIME.LE.
     1   55))THEN
              RXLVL=148.3
         ELSEIF((TIME.GT.43.AND.TIME.LE.44).OR.(TIME.GT.52.AND.TIME.LE.
     1   54))THEN
              RXLVL=148.2
         ELSEIF((TIME.GT.44.AND.TIME.LE.46).OR.(TIME.GT.48.AND.TIME.LE.
     1   52))THEN
              RXLVL=148.1
         ELSEIF(TIME.GT.46.AND.TIME.LE.48)THEN
              RXLVL=148.0
         ELSEIF(TIME.GT.66.AND.TIME.LE.67)THEN
              RXLVL=149.0
         ELSEIF(TIME.GT.67.AND.TIME.LE.68)THEN
              RXLVL=149.1
         ELSEIF(TIME.GT.69.AND.TIME.LE.70)THEN
              RXLVL=149.3
         ELSEIF(TIME.GT.70.AND.TIME.LE.73)THEN
              RXLVL=149.4
         ENDIF
      END
C
C MEASURED TEST DATA: FEEDWATER FLOW(PER UNIT)
      SUBROUTINE MSRDT2(TIME,FDFLO)
         IF(TIME.LE.0.OR.TIME.GT.71)THEN
              FDFLO=1.00
         ELSEIF(TIME.GT.0.AND.TIME.LE.1)THEN
              FDFLO=.948
         ELSEIF(TIME.GT.1.AND.TIME.LE.2)THEN
              FDFLO=.791
         ELSEIF(TIME.GT.2.AND.TIME.LE.3)THEN
              FDFLO=.757
         ELSEIF(TIME.GT.3.AND.TIME.LE.4)THEN
              FDFLO=.809
         ELSEIF(TIME.GT.4.AND.TIME.LE.5)THEN
              FDFLO=.800
         ELSEIF(TIME.GT.5.AND.TIME.LE.7)THEN
              FDFLO=.765
         ELSEIF(TIME.GT.7.AND.TIME.LE.8)THEN
              FDFLO=.791
         ELSEIF(TIME.GT.8.AND.TIME.LE.9)THEN
              FDFLO=.783
         ELSEIF(TIME.GT.9.AND.TIME.LE.10)THEN
```

```
          FDFLO=.800
      ELSEIF(TIME.GT.10.AND.TIME.LE.11)THEN
          FDFLO=.809
      ELSEIF(TIME.GT.11.AND.TIME.LE.12)THEN
          FDFLO=.817
      ELSEIF(TIME.GT.12.AND.TIME.LE.13)THEN
          FDFLO=.826
      ELSEIF(TIME.GT.13.AND.TIME.LE.14)THEN
          FDFLO=.830
      ELSEIF(TIME.GT.14.AND.TIME.LE.15)THEN
          FDFLO=.843
      ELSEIF(TIME.GT.15.AND.TIME.LE.16)THEN
          FDFLO=.852
      ELSEIF(TIME.GT.16.AND.TIME.LE.17)THEN
          FDFLO=.856
      ELSEIF(TIME.GT.17.AND.TIME.LE.18)THEN
          FDFLO=.861
      ELSEIF(TIME.GT.18.AND.TIME.LE.19)THEN
          FDFLO=.868
      ELSEIF(TIME.GT.19.AND.TIME.LE.20)THEN
          FDFLO=.873
      ELSEIF(TIME.GT.20.AND.TIME.LE.21)THEN
          FDFLO=.887
      ELSEIF(TIME.GT.21.AND.TIME.LE.22)THEN
          FDFLO=.896
      ELSEIF(TIME.GT.22.AND.TIME.LE.23)THEN
          FDFLO=.904
      ELSEIF(TIME.GT.23.AND.TIME.LE.24)THEN
          FDFLO=.913
      ELSEIF(TIME.GT.24.AND.TIME.LE.25)THEN
          FDFLO=.918
      ELSEIF(TIME.GT.25.AND.TIME.LE.26)THEN
          FDFLO=.922
      ELSEIF(TIME.GT.26.AND.TIME.LE.27)THEN
          FDFLO=.930
      ELSEIF(TIME.GT.27.AND.TIME.LE.28)THEN
          FDFLO=.939
      ELSEIF(TIME.GT.28.AND.TIME.LE.29)THEN
          FDFLO=.948
      ELSEIF(TIME.GT.29.AND.TIME.LE.30)THEN
          FDFLO=.956
      ELSEIF(TIME.GT.30.AND.TIME.LE.31)THEN
          FDFLO=.960
      ELSEIF(TIME.GT.31.AND.TIME.LE.32)THEN
          FDFLO=.965
      ELSEIF(TIME.GT.32.AND.TIME.LE.33)THEN
          FDFLO=.974
      ELSEIF(TIME.GT.33.AND.TIME.LE.34)THEN
          FDFLO=.983
      ELSEIF(TIME.GT.34.AND.TIME.LE.35)THEN
          FDFLO=.988
      ELSEIF(TIME.GT.35.AND.TIME.LE.36)THEN
          FDFLO=.991
      ELSEIF(TIME.GT.36.AND.TIME.LE.37)THEN
          FDFLO=.993
      ELSEIF(TIME.GT.37.AND.TIME.LE.38)THEN
          FDFLO=.995
      ELSEIF(TIME.GT.37.AND.TIME.LE.39)THEN
          FDFLO=.997
      ELSEIF(TIME.GT.39.AND.TIME.LE.40)THEN
          FDFLO=1.00
      ELSEIF((TIME.GT.40.AND.TIME.LE.44).OR.(TIME.GT.66.AND.TIME.LE.
     1 71))THEN
          FDFLO=1.01
      ELSEIF(TIME.GT.44.AND.TIME.LE.66)THEN
          FDFLO=1.02
```

```
        ENDIF
      END
C
C MEASURED TEST DATA: STEAM FLOW(PER UNIT)
      SUBROUTINE MSRDT3(TIME,SMFLO)
        IF(TIME.LE.19.OR.TIME.GT.41)THEN
             SMFLO=1.00
        ELSEIF(TIME.GT.19.AND.TIME.LE.20)THEN
             SMFLO=.991
        ELSEIF(TIME.GT.20.AND.TIME.LE.21)THEN
             SMFLO=.988
        ELSEIF(TIME.GT.21.AND.TIME.LE.22)THEN
             SMFLO=.985
        ELSEIF(TIME.GT.22.AND.TIME.LE.23)THEN
             SMFLO=.984
        ELSEIF(TIME.GT.23.AND.TIME.LE.27)THEN
             SMFLO=.982
        ELSEIF(TIME.GT.27.AND.TIME.LE.28)THEN
             SMFLO=.984
        ELSEIF(TIME.GT.28.AND.TIME.LE.29)THEN
             SMFLO=.985
        ELSEIF(TIME.GT.29.AND.TIME.LE.30)THEN
             SMFLO=.988
        ELSEIF(TIME.GT.30.AND.TIME.LE.31)THEN
             SMFLO=.989
        ELSEIF(TIME.GT.31.AND.TIME.LE.32)THEN
             SMFLO=.991
        ELSEIF(TIME.GT.32.AND.TIME.LE.33)THEN
             SMFLO=.992
        ELSEIF(TIME.GT.33.AND.TIME.LE.34)THEN
             SMFLO=.993
        ELSEIF(TIME.GT.34.AND.TIME.LE.35)THEN
             SMFLO=.994
        ELSEIF(TIME.GT.35.AND.TIME.LE.36)THEN
             SMFLO=.995
        ELSEIF(TIME.GT.36.AND.TIME.LE.37)THEN
             SMFLO=.996
        ELSEIF(TIME.GT.37.AND.TIME.LE.38)THEN
             SMFLO=.996
        ELSEIF(TIME.GT.37.AND.TIME.LE.39)THEN
             SMFLO=.997
        ELSEIF(TIME.GT.39.AND.TIME.LE.40)THEN
             SMFLO=.998
        ELSEIF(TIME.GT.40.AND.TIME.LE.41)THEN
             SMFLO=.999
        ENDIF
      END
C
C MEASURED TEST DATA: REACTOR POWER(%)
      SUBROUTINE MSRDT4(TIME,RXPWR)
        IF(TIME.LE.1.OR.TIME.GT.42)THEN
             RXPWR=100.
        ELSEIF(TIME.GT.1.AND.TIME.LE.2)THEN
             RXPWR=99.75
        ELSEIF(TIME.GT.2.AND.TIME.LE.3)THEN
             RXPWR=99.5
        ELSEIF(TIME.GT.3.AND.TIME.LE.6)THEN
             RXPWR=99.0
        ELSEIF(TIME.GT.6.AND.TIME.LE.7)THEN
             RXPWR=99.5
        ELSEIF(TIME.GT.7.AND.TIME.LE.8)THEN
             RXPWR=100.
        ELSEIF(TIME.GT.8.AND.TIME.LE.10)THEN
             RXPWR=99.5
        ELSEIF(TIME.GT.10.AND.TIME.LE.11)THEN
             RXPWR=99.25
```

```
        ELSEIF(TIME.GT.11.AND.TIME.LE.12)THEN
             RXPWR=98.5
        ELSEIF(TIME.GT.12.AND.TIME.LE.13)THEN
             RXPWR=97.5
        ELSEIF(TIME.GT.13.AND.TIME.LE.14)THEN
             RXPWR=95.0
        ELSEIF(TIME.GT.14.AND.TIME.LE.15)THEN
             RXPWR=92.5
        ELSEIF(TIME.GT.15.AND.TIME.LE.17)THEN
             RXPWR=90.0
        ELSEIF(TIME.GT.17.AND.TIME.LE.18)THEN
             RXPWR=89.5
        ELSEIF(TIME.GT.18.AND.TIME.LE.19)THEN
             RXPWR=90.0
        ELSEIF(TIME.GT.19.AND.TIME.LE.20)THEN
             RXPWR=90.5
        ELSEIF(TIME.GT.20.AND.TIME.LE.21)THEN
             RXPWR=91.0
        ELSEIF(TIME.GT.21.AND.TIME.LE.22)THEN
             RXPWR=92.0
        ELSEIF(TIME.GT.22.AND.TIME.LE.23)THEN
             RXPWR=93.0
        ELSEIF(TIME.GT.23.AND.TIME.LE.24)THEN
             RXPWR=94.0
        ELSEIF(TIME.GT.24.AND.TIME.LE.26)THEN
             RXPWR=94.5
        ELSEIF(TIME.GT.26.AND.TIME.LE.27)THEN
             RXPWR=95.5
        ELSEIF(TIME.GT.27.AND.TIME.LE.35)THEN
             RXPWR=97.0
        ELSEIF(TIME.GT.35.AND.TIME.LE.36)THEN
             RXPWR=98.0
        ELSEIF(TIME.GT.36.AND.TIME.LE.41)THEN
             RXPWR=98.5
        ELSEIF(TIME.GT.41.AND.TIME.LE.42)THEN
             RXPWR=99.0
        ENDIF
      END
C  PI COMPENSATOR DATA: REACTOR WATER LEVEL(INCHES TAF)
      SUBROUTINE PICOM1(TIME,PILVL)
        IF(TIME.LE.6)THEN
             PILVL=160.0
        ELSEIF(TIME.GT.6.AND.TIME.LE.7)THEN
             PILVL=159.7
        ELSEIF(TIME.GT.7.AND.TIME.LE.8)THEN
             PILVL=159.4
        ELSEIF(TIME.GT.8.AND.TIME.LE.9)THEN
             PILVL=159.0
        ELSEIF(TIME.GT.9.AND.TIME.LE.10)THEN
             PILVL=158.6
        ELSEIF(TIME.GT.10.AND.TIME.LE.11)THEN
             PILVL=158.1
        ELSEIF(TIME.GT.11.AND.TIME.LE.12)THEN
             PILVL=157.6
        ELSEIF(TIME.GT.12.AND.TIME.LE.13)THEN
             PILVL=157.2
        ELSEIF(TIME.GT.13.AND.TIME.LE.14)THEN
             PILVL=156.7
        ELSEIF(TIME.GT.14.AND.TIME.LE.15)THEN
             PILVL=156.2
        ELSEIF(TIME.GT.15.AND.TIME.LE.16)THEN
             PILVL=155.8
        ELSEIF(TIME.GT.16.AND.TIME.LE.17)THEN
             PILVL=155.3
        ELSEIF(TIME.GT.17.AND.TIME.LE.18)THEN
             PILVL=154.9
```

```
          ELSEIF(TIME.GT.18.AND.TIME.LE.19)THEN
                PILVL=154.5
          ELSEIF(TIME.GT.19.AND.TIME.LE.20)THEN
                PILVL=154.1
          ELSEIF(TIME.GT.20.AND.TIME.LE.21)THEN
                PILVL=153.7
          ELSEIF(TIME.GT.21.AND.TIME.LE.22)THEN
                PILVL=153.4
          ELSEIF(TIME.GT.22.AND.TIME.LE.23)THEN
                PILVL=153.1
          ELSEIF(TIME.GT.23.AND.TIME.LE.24)THEN
                PILVL=152.8
          ELSEIF(TIME.GT.24.AND.TIME.LE.25)THEN
                PILVL=152.5
          ELSEIF(TIME.GT.25.AND.TIME.LE.26)THEN
                PILVL=152.2
          ELSEIF(TIME.GT.26.AND.TIME.LE.27)THEN
                PILVL=152.0
          ELSEIF(TIME.GT.27.AND.TIME.LE.28)THEN
                PILVL=151.8
          ELSEIF(TIME.GT.28.AND.TIME.LE.29)THEN
                PILVL=151.6
          ELSEIF(TIME.GT.29.AND.TIME.LE.30)THEN
                PILVL=151.4
          ELSEIF(TIME.GT.30.AND.TIME.LE.31)THEN
                PILVL=151.2
          ELSEIF(TIME.GT.31.AND.TIME.LE.32)THEN
                PILVL=151.0
          ELSEIF(TIME.GT.32.AND.TIME.LE.33)THEN
                PILVL=150.9
          ELSEIF(TIME.GT.33.AND.TIME.LE.34)THEN
                PILVL=150.8
          ELSEIF(TIME.GT.34.AND.TIME.LE.35)THEN
                PILVL=150.6
          ELSEIF(TIME.GT.35.AND.TIME.LE.36)THEN
                PILVL=150.5
          ELSEIF(TIME.GT.36.AND.TIME.LE.37)THEN
                PILVL=150.4
          ELSEIF(TIME.GT.37.AND.TIME.LE.39)THEN
                PILVL=150.3
          ELSEIF(TIME.GT.39.AND.TIME.LE.40)THEN
                PILVL=150.2
          ELSEIF(TIME.GT.40.AND.TIME.LE.42)THEN
                PILVL=150.1
          ELSEIF(TIME.GT.42.AND.TIME.LE.44)THEN
                PILVL=150.0
          ELSEIF(TIME.GT.44.AND.TIME.LE.47)THEN
                PILVL=149.9
          ELSEIF((TIME.GT.47.AND.TIME.LE.51).OR.(TIME.GT.106))THEN
                PILVL=149.8
          ELSEIF(TIME.GT.51.AND.TIME.LE.106)THEN
                PILVL=149.7
          ENDIF
      END
C
C PI COMPENSATOR DATA: FEEDWATER FLOW
      SUBROUTINE PICOM2(TIME,PIFLO)
          IF(TIME.LE.0)THEN
                PIFLO=2023.
          ELSEIF(TIME.GT.0.AND.TIME.LE.1)THEN
                PIFLO=1641.
          ELSEIF(TIME.GT.1.AND.TIME.LE.2)THEN
                PIFLO=1610.
          ELSEIF(TIME.GT.2.AND.TIME.LE.3)THEN
                PIFLO=1613.
          ELSEIF(TIME.GT.3.AND.TIME.LE.4)THEN
```

```
      PIFLO=1614.
   ELSEIF(TIME.GT.4.AND.TIME.LE.6)THEN
      PIFLO=1612.
   ELSEIF(TIME.GT.6.AND.TIME.LE.7)THEN
      PIFLO=1615.
   ELSEIF(TIME.GT.7.AND.TIME.LE.8)THEN
      PIFLO=1624.
   ELSEIF(TIME.GT.8.AND.TIME.LE.9)THEN
      PIFLO=1636.
   ELSEIF(TIME.GT.9.AND.TIME.LE.10)THEN
      PIFLO=1651.
   ELSEIF(TIME.GT.10.AND.TIME.LE.11)THEN
      PIFLO=1668.
   ELSEIF(TIME.GT.11.AND.TIME.LE.12)THEN
      PIFLO=1686.
   ELSEIF(TIME.GT.12.AND.TIME.LE.13)THEN
      PIFLO=1704.
   ELSEIF(TIME.GT.13.AND.TIME.LE.14)THEN
      PIFLO=1723.
   ELSEIF(TIME.GT.14.AND.TIME.LE.15)THEN
      PIFLO=1740.
   ELSEIF(TIME.GT.15.AND.TIME.LE.16)THEN
      PIFLO=1759.
   ELSEIF(TIME.GT.16.AND.TIME.LE.17)THEN
      PIFLO=1776.
   ELSEIF(TIME.GT.17.AND.TIME.LE.18)THEN
      PIFLO=1793.
   ELSEIF(TIME.GT.18.AND.TIME.LE.19)THEN
      PIFLO=1810.
   ELSEIF(TIME.GT.19.AND.TIME.LE.20)THEN
      PIFLO=1826.
   ELSEIF(TIME.GT.20.AND.TIME.LE.21)THEN
      PIFLO=1841.
   ELSEIF(TIME.GT.21.AND.TIME.LE.22)THEN
      PIFLO=1855.
   ELSEIF(TIME.GT.22.AND.TIME.LE.23)THEN
      PIFLO=1868.
   ELSEIF(TIME.GT.23.AND.TIME.LE.24)THEN
      PIFLO=1881.
   ELSEIF(TIME.GT.24.AND.TIME.LE.25)THEN
      PIFLO=1893.
   ELSEIF(TIME.GT.25.AND.TIME.LE.26)THEN
      PIFLO=1904.
   ELSEIF(TIME.GT.26.AND.TIME.LE.27)THEN
      PIFLO=1914.
   ELSEIF(TIME.GT.27.AND.TIME.LE.28)THEN
      PIFLO=1924.
   ELSEIF(TIME.GT.28.AND.TIME.LE.29)THEN
      PIFLO=1933.
   ELSEIF(TIME.GT.29.AND.TIME.LE.30)THEN
      PIFLO=1941.
   ELSEIF(TIME.GT.30.AND.TIME.LE.31)THEN
      PIFLO=1949.
   ELSEIF(TIME.GT.31.AND.TIME.LE.32)THEN
      PIFLO=1956.
   ELSEIF(TIME.GT.32.AND.TIME.LE.33)THEN
      PIFLO=1962.
   ELSEIF(TIME.GT.33.AND.TIME.LE.34)THEN
      PIFLO=1968.
   ELSEIF(TIME.GT.34.AND.TIME.LE.35)THEN
      PIFLO=1973.
   ELSEIF(TIME.GT.35.AND.TIME.LE.36)THEN
      PIFLO=1978.
   ELSEIF(TIME.GT.36.AND.TIME.LE.37)THEN
      PIFLO=1983.
   ELSEIF(TIME.GT.37.AND.TIME.LE.38)THEN
```

```
          PIFLO=1987.
      ELSEIF(TIME.GT.38.AND.TIME.LE.39)THEN
          PIFLO=1991.
      ELSEIF(TIME.GT.39.AND.TIME.LE.40)THEN
          PIFLO=1994.
      ELSEIF(TIME.GT.40.AND.TIME.LE.41)THEN
          PIFLO=1998.
      ELSEIF(TIME.GT.41.AND.TIME.LE.42)THEN
          PIFLO=2000.
      ELSEIF(TIME.GT.42.AND.TIME.LE.43)THEN
          PIFLO=2003.
      ELSEIF(TIME.GT.43.AND.TIME.LE.44)THEN
          PIFLO=2005.
      ELSEIF(TIME.GT.44.AND.TIME.LE.45)THEN
          PIFLO=2007.
      ELSEIF(TIME.GT.45.AND.TIME.LE.46)THEN
          PIFLO=2009.
      ELSEIF(TIME.GT.46.AND.TIME.LE.47)THEN
          PIFLO=2011.
      ELSEIF(TIME.GT.47.AND.TIME.LE.48)THEN
          PIFLO=2012.
      ELSEIF(TIME.GT.48.AND.TIME.LE.49)THEN
          PIFLO=2014.
      ELSEIF(TIME.GT.49.AND.TIME.LE.50)THEN
          PIFLO=2015.
      ELSEIF(TIME.GT.50.AND.TIME.LE.51)THEN
          PIFLO=2016.
      ELSEIF(TIME.GT.51.AND.TIME.LE.52)THEN
          PIFLO=2017.
      ELSEIF(TIME.GT.52.AND.TIME.LE.53)THEN
          PIFLO=2018.
      ELSEIF(TIME.GT.53.AND.TIME.LE.55)THEN
          PIFLO=2019.
      ELSEIF(TIME.GT.55.AND.TIME.LE.56)THEN
          PIFLO=2020.
      ELSEIF(TIME.GT.56.AND.TIME.LE.58)THEN
          PIFLO=2021.
      ELSEIF(TIME.GT.58.AND.TIME.LE.61)THEN
          PIFLO=2022.
      ELSEIF(TIME.GT.61.AND.TIME.LE.66)THEN
          PIFLO=2023.
      ELSEIF(TIME.GT.66.AND.TIME.LE.83)THEN
          PIFLO=2024.
      ELSEIF(TIME.GT.83)THEN
          PIFLO=2025.
      ENDIF
   END
C  PI COMPENSATOR DATA(W/BACKLASH): REACTOR WATER LEVEL(INCHES TAF)
   SUBROUTINE BICOM1(TIME,BILVL)
      IF(TIME.LE.4)THEN
          BILVL=160.0
      ELSEIF(TIME.GT.4.AND.TIME.LE.5)THEN
          BILVL=160.1
      ELSEIF(TIME.GT.5.AND.TIME.LE.6)THEN
          BILVL=160.0
      ELSEIF(TIME.GT.6.AND.TIME.LE.7)THEN
          BILVL=159.8
      ELSEIF(TIME.GT.7.AND.TIME.LE.8)THEN
          BILVL=159.4
      ELSEIF(TIME.GT.8.AND.TIME.LE.9)THEN
          BILVL=159.0
      ELSEIF(TIME.GT.9.AND.TIME.LE.10)THEN
          BILVL=158.6
      ELSEIF(TIME.GT.10.AND.TIME.LE.11)THEN
          BILVL=158.1
      ELSEIF(TIME.GT.11.AND.TIME.LE.12)THEN
```

```
      BILVL=157.6
   ELSEIF(TIME.GT.12.AND.TIME.LE.13)THEN
      BILVL=157.2
   ELSEIF(TIME.GT.13.AND.TIME.LE.14)THEN
      BILVL=156.7
   ELSEIF(TIME.GT.14.AND.TIME.LE.15)THEN
      BILVL=156.2
   ELSEIF(TIME.GT.15.AND.TIME.LE.16)THEN
      BILVL=155.8
   ELSEIF(TIME.GT.16.AND.TIME.LE.17)THEN
      BILVL=155.3
   ELSEIF(TIME.GT.17.AND.TIME.LE.18)THEN
      BILVL=154.9
   ELSEIF(TIME.GT.18.AND.TIME.LE.19)THEN
      BILVL=154.5
   ELSEIF(TIME.GT.19.AND.TIME.LE.20)THEN
      BILVL=154.1
   ELSEIF(TIME.GT.20.AND.TIME.LE.21)THEN
      BILVL=153.7
   ELSEIF(TIME.GT.21.AND.TIME.LE.22)THEN
      BILVL=153.3
   ELSEIF(TIME.GT.22.AND.TIME.LE.23)THEN
      BILVL=153.0
   ELSEIF(TIME.GT.23.AND.TIME.LE.24)THEN
      BILVL=152.7
   ELSEIF(TIME.GT.24.AND.TIME.LE.25)THEN
      BILVL=152.4
   ELSEIF(TIME.GT.25.AND.TIME.LE.26)THEN
      BILVL=152.1
   ELSEIF(TIME.GT.26.AND.TIME.LE.27)THEN
      BILVL=151.9
   ELSEIF(TIME.GT.27.AND.TIME.LE.28)THEN
      BILVL=151.6
   ELSEIF(TIME.GT.28.AND.TIME.LE.29)THEN
      BILVL=151.4
   ELSEIF(TIME.GT.29.AND.TIME.LE.30)THEN
      BILVL=151.2
   ELSEIF(TIME.GT.30.AND.TIME.LE.31)THEN
      BILVL=151.0
   ELSEIF(TIME.GT.31.AND.TIME.LE.32)THEN
      BILVL=150.9
   ELSEIF(TIME.GT.32.AND.TIME.LE.33)THEN
      BILVL=150.7
   ELSEIF(TIME.GT.33.AND.TIME.LE.34)THEN
      BILVL=150.6
   ELSEIF(TIME.GT.34.AND.TIME.LE.35)THEN
      BILVL=150.5
   ELSEIF(TIME.GT.35.AND.TIME.LE.36)THEN
      BILVL=150.3
   ELSEIF(TIME.GT.36.AND.TIME.LE.37)THEN
      BILVL=150.2
   ELSEIF(TIME.GT.37.AND.TIME.LE.39)THEN
      BILVL=150.1
   ELSEIF(TIME.GT.39.AND.TIME.LE.40)THEN
      BILVL=150.0
   ELSEIF(TIME.GT.40.AND.TIME.LE.42)THEN
      BILVL=149.9
   ELSEIF(TIME.GT.42.AND.TIME.LE.43)THEN
      BILVL=149.8
   ELSEIF(TIME.GT.43.AND.TIME.LE.46)THEN
      BILVL=149.7
   ELSEIF(TIME.GT.46.AND.TIME.LE.49)THEN
      BILVL=149.6
   ELSEIF((TIME.GT.49.AND.TIME.LE.56).OR.(TIME.GT.76))THEN
      BILVL=149.5
   ELSEIF(TIME.GT.56.AND.TIME.LE.76)THEN
```

```
            BILVL=149.4
         ENDIF
      END
C
C PI COMPENSATOR DATA(W/BACKLASH): FEEDWATER FLOW
      SUBROUTINE BICOM2(TIME,BIFLO)
         IF(TIME.LE.0)THEN
            BIFLO=2022.
         ELSEIF(TIME.GT.0.AND.TIME.LE.1)THEN
            BIFLO=1878.
         ELSEIF(TIME.GT.1.AND.TIME.LE.2)THEN
            BIFLO=1694.
         ELSEIF(TIME.GT.2.AND.TIME.LE.3)THEN
            BIFLO=1616.
         ELSEIF(TIME.GT.3.AND.TIME.LE.4)THEN
            BIFLO=1592.
         ELSEIF(TIME.GT.4.AND.TIME.LE.8)THEN
            BIFLO=1614.
         ELSEIF(TIME.GT.8.AND.TIME.LE.9)THEN
            BIFLO=1627.
         ELSEIF(TIME.GT.9.AND.TIME.LE.10)THEN
            BIFLO=1639.
         ELSEIF(TIME.GT.10.AND.TIME.LE.11)THEN
            BIFLO=1663.
         ELSEIF(TIME.GT.11.AND.TIME.LE.12)THEN
            BIFLO=1675.
         ELSEIF(TIME.GT.12.AND.TIME.LE.13)THEN
            BIFLO=1696.
         ELSEIF(TIME.GT.13.AND.TIME.LE.14)THEN
            BIFLO=1715.
         ELSEIF(TIME.GT.14.AND.TIME.LE.15)THEN
            BIFLO=1728.
         ELSEIF(TIME.GT.15.AND.TIME.LE.16)THEN
            BIFLO=1750.
         ELSEIF(TIME.GT.16.AND.TIME.LE.17)THEN
            BIFLO=1769.
         ELSEIF(TIME.GT.17.AND.TIME.LE.18)THEN
            BIFLO=1783.
         ELSEIF(TIME.GT.18.AND.TIME.LE.19)THEN
            BIFLO=1798.
         ELSEIF(TIME.GT.19.AND.TIME.LE.20)THEN
            BIFLO=1823.
         ELSEIF(TIME.GT.20.AND.TIME.LE.21)THEN
            BIFLO=1836.
         ELSEIF(TIME.GT.21.AND.TIME.LE.22)THEN
            BIFLO=1849.
         ELSEIF(TIME.GT.22.AND.TIME.LE.23)THEN
            BIFLO=1861.
         ELSEIF(TIME.GT.23.AND.TIME.LE.24)THEN
            BIFLO=1871.
         ELSEIF(TIME.GT.24.AND.TIME.LE.25)THEN
            BIFLO=1886.
         ELSEIF(TIME.GT.25.AND.TIME.LE.26)THEN
            BIFLO=1903.
         ELSEIF(TIME.GT.26.AND.TIME.LE.27)THEN
            BIFLO=1914.
         ELSEIF(TIME.GT.27.AND.TIME.LE.28)THEN
            BIFLO=1923.
         ELSEIF(TIME.GT.28.AND.TIME.LE.29)THEN
            BIFLO=1932.
         ELSEIF(TIME.GT.29.AND.TIME.LE.30)THEN
            BIFLO=1939.
         ELSEIF(TIME.GT.30.AND.TIME.LE.31)THEN
            BIFLO=1947.
         ELSEIF(TIME.GT.31.AND.TIME.LE.32)THEN
            BIFLO=1953.
```

```
      ELSEIF(TIME.GT.32.AND.TIME.LE.33)THEN
           BIFLO=1959.
      ELSEIF(TIME.GT.33.AND.TIME.LE.34)THEN
           BIFLO=1965.
      ELSEIF(TIME.GT.34.AND.TIME.LE.35)THEN
           BIFLO=1969.
      ELSEIF(TIME.GT.35.AND.TIME.LE.36)THEN
           BIFLO=1974.
      ELSEIF(TIME.GT.36.AND.TIME.LE.37)THEN
           BIFLO=1980.
      ELSEIF(TIME.GT.37.AND.TIME.LE.38)THEN
           BIFLO=1983.
      ELSEIF(TIME.GT.38.AND.TIME.LE.39)THEN
           BIFLO=1989.
      ELSEIF(TIME.GT.39.AND.TIME.LE.40)THEN
           BIFLO=1992.
      ELSEIF(TIME.GT.40.AND.TIME.LE.41)THEN
           BIFLO=1995.
      ELSEIF(TIME.GT.41.AND.TIME.LE.42)THEN
           BIFLO=1998.
      ELSEIF(TIME.GT.42.AND.TIME.LE.43)THEN
           BIFLO=2001.
      ELSEIF(TIME.GT.43.AND.TIME.LE.44)THEN
           BIFLO=2004.
      ELSEIF(TIME.GT.44.AND.TIME.LE.45)THEN
           BIFLO=2005.
      ELSEIF(TIME.GT.45.AND.TIME.LE.46)THEN
           BIFLO=2007.
      ELSEIF(TIME.GT.46.AND.TIME.LE.47)THEN
           BIFLO=2008.
      ELSEIF(TIME.GT.47.AND.TIME.LE.48)THEN
           BIFLO=2011.
      ELSEIF(TIME.GT.48.AND.TIME.LE.49)THEN
           BIFLO=2013.
      ELSEIF(TIME.GT.49.AND.TIME.LE.51)THEN
           BIFLO=2014.
      ELSEIF(TIME.GT.51.AND.TIME.LE.52)THEN
           BIFLO=2016.
      ELSEIF(TIME.GT.52.AND.TIME.LE.54)THEN
           BIFLO=2017.
      ELSEIF(TIME.GT.54.AND.TIME.LE.56)THEN
           BIFLO=2019.
      ELSEIF(TIME.GT.56.AND.TIME.LE.58)THEN
           BIFLO=2020.
      ELSEIF(TIME.GT.58.AND.TIME.LE.61)THEN
           BIFLO=2022.
      ELSEIF(TIME.GT.61.AND.TIME.LE.67)THEN
           BIFLO=2023.
      ELSEIF(TIME.GT.67.AND.TIME.LE.84)THEN
           BIFLO=2024.
      ELSEIF(TIME.GT.84)THEN
           BIFLO=2026.
      ENDIF
   END
C  PI COMPENSATOR DATA(W/BACKLASH): LEVEL(INCHES TAF) msiv closure
   SUBROUTINE MICOM1(TIME,MILVL)
      IF(TIME.LE.0)THEN
           MILVL=160.0
      ELSEIF(TIME.GT.0.AND.TIME.LE.1)THEN
           MILVL=156.1
      ELSEIF(TIME.GT.1.AND.TIME.LE.2)THEN
           MILVL=154.3
      ELSEIF(TIME.GT.2.AND.TIME.LE.3)THEN
           MILVL=151.2
      ELSEIF(TIME.GT.3.AND.TIME.LE.4)THEN
           MILVL=148.6
```

```
ELSEIF(TIME.GT.4.AND.TIME.LE.5)THEN
      MILVL=146.8
ELSEIF(TIME.GT.5.AND.TIME.LE.6)THEN
      MILVL=145.8
ELSEIF(TIME.GT.6.AND.TIME.LE.7)THEN
      MILVL=145.4
ELSEIF(TIME.GT.7.AND.TIME.LE.8)THEN
      MILVL=145.1
ELSEIF(TIME.GT.8.AND.TIME.LE.9)THEN
      MILVL=144.9
ELSEIF(TIME.GT.9.AND.TIME.LE.10)THEN
      MILVL=144.6
ELSEIF(TIME.GT.10.AND.TIME.LE.11)THEN
      MILVL=144.3
ELSEIF(TIME.GT.11.AND.TIME.LE.12)THEN
      MILVL=139.7
ELSEIF(TIME.GT.12.AND.TIME.LE.13)THEN
      MILVL=140.4
ELSEIF(TIME.GT.13.AND.TIME.LE.14)THEN
      MILVL=141.1
ELSEIF(TIME.GT.14.AND.TIME.LE.15)THEN
      MILVL=141.9
ELSEIF(TIME.GT.15.AND.TIME.LE.16)THEN
      MILVL=142.7
ELSEIF(TIME.GT.16.AND.TIME.LE.17)THEN
      MILVL=143.5
ELSEIF(TIME.GT.17.AND.TIME.LE.18)THEN
      MILVL=144.4
ELSEIF(TIME.GT.18.AND.TIME.LE.19)THEN
      MILVL=145.4
ELSEIF(TIME.GT.19.AND.TIME.LE.20)THEN
      MILVL=146.4
ELSEIF(TIME.GT.20.AND.TIME.LE.21)THEN
      MILVL=147.5
ELSEIF(TIME.GT.21.AND.TIME.LE.22)THEN
      MILVL=148.6
ELSEIF(TIME.GT.22.AND.TIME.LE.23)THEN
      MILVL=149.7
ELSEIF(TIME.GT.23.AND.TIME.LE.24)THEN
      MILVL=150.9
ELSEIF(TIME.GT.24.AND.TIME.LE.25)THEN
      MILVL=151.9
ELSEIF(TIME.GT.25.AND.TIME.LE.26)THEN
      MILVL=153.0
ELSEIF(TIME.GT.26.AND.TIME.LE.27)THEN
      MILVL=154.0
ELSEIF(TIME.GT.27.AND.TIME.LE.28)THEN
      MILVL=155.0
ELSEIF(TIME.GT.28.AND.TIME.LE.29)THEN
      MILVL=156.0
ELSEIF(TIME.GT.29.AND.TIME.LE.30)THEN
      MILVL=156.9
ELSEIF(TIME.GT.30.AND.TIME.LE.31)THEN
      MILVL=157.8
ELSEIF(TIME.GT.31.AND.TIME.LE.32)THEN
      MILVL=158.6
ELSEIF(TIME.GT.32.AND.TIME.LE.33)THEN
      MILVL=159.4
ELSEIF(TIME.GT.33.AND.TIME.LE.34)THEN
      MILVL=160.1
ELSEIF(TIME.GT.34.AND.TIME.LE.35)THEN
      MILVL=160.8
ELSEIF(TIME.GT.35.AND.TIME.LE.36)THEN
      MILVL=161.5
ELSEIF(TIME.GT.36.AND.TIME.LE.37)THEN
      MILVL=162.1
```

```
ELSEIF(TIME.GT.37.AND.TIME.LE.38)THEN
      MILVL=162.7
ELSEIF(TIME.GT.38.AND.TIME.LE.39)THEN
      MILVL=163.2
ELSEIF(TIME.GT.39.AND.TIME.LE.40)THEN
      MILVL=163.7
ELSEIF(TIME.GT.40.AND.TIME.LE.41)THEN
      MILVL=164.2
ELSEIF(TIME.GT.41.AND.TIME.LE.42)THEN
      MILVL=164.7
ELSEIF(TIME.GT.42.AND.TIME.LE.43)THEN
      MILVL=165.1
ELSEIF(TIME.GT.43.AND.TIME.LE.44)THEN
      MILVL=165.5
ELSEIF(TIME.GT.44.AND.TIME.LE.45)THEN
      MILVL=165.9
ELSEIF(TIME.GT.45.AND.TIME.LE.46)THEN
      MILVL=166.2
ELSEIF(TIME.GT.46.AND.TIME.LE.47)THEN
      MILVL=166.6
ELSEIF(TIME.GT.47.AND.TIME.LE.48)THEN
      MILVL=166.9
ELSEIF(TIME.GT.48.AND.TIME.LE.49)THEN
      MILVL=167.1
ELSEIF(TIME.GT.49.AND.TIME.LE.50)THEN
      MILVL=167.4
ELSEIF(TIME.GT.50.AND.TIME.LE.51)THEN
      MILVL=167.6
ELSEIF(TIME.GT.51.AND.TIME.LE.52)THEN
      MILVL=167.9
ELSEIF(TIME.GT.52.AND.TIME.LE.53)THEN
      MILVL=168.1
ELSEIF(TIME.GT.53.AND.TIME.LE.54)THEN
      MILVL=168.3
ELSEIF(TIME.GT.54.AND.TIME.LE.55)THEN
      MILVL=168.4
ELSEIF(TIME.GT.55.AND.TIME.LE.56)THEN
      MILVL=168.6
ELSEIF(TIME.GT.56.AND.TIME.LE.57)THEN
      MILVL=168.8
ELSEIF(TIME.GT.57.AND.TIME.LE.58)THEN
      MILVL=168.9
ELSEIF(TIME.GT.58.AND.TIME.LE.59)THEN
      MILVL=169.0
ELSEIF(TIME.GT.59.AND.TIME.LE.60)THEN
      MILVL=169.1
ELSEIF(TIME.GT.60.AND.TIME.LE.61)THEN
      MILVL=169.2
ELSEIF(TIME.GT.61.AND.TIME.LE.62)THEN
      MILVL=169.5
ELSEIF(TIME.GT.62.AND.TIME.LE.63)THEN
      MILVL=169.7
ELSEIF(TIME.GT.63.AND.TIME.LE.64)THEN
      MILVL=170.0
ELSEIF(TIME.GT.64.AND.TIME.LE.65)THEN
      MILVL=170.2
ELSEIF(TIME.GT.65.AND.TIME.LE.66)THEN
      MILVL=170.4
ELSEIF(TIME.GT.66.AND.TIME.LE.67)THEN
      MILVL=170.6
ELSEIF(TIME.GT.67.AND.TIME.LE.68)THEN
      MILVL=170.8
ELSEIF(TIME.GT.68.AND.TIME.LE.69)THEN
      MILVL=170.9
ELSEIF(TIME.GT.69.AND.TIME.LE.70)THEN
      MILVL=171.1
```

```
          ELSEIF(TIME.GT.70.AND.TIME.LE.71)THEN
               MILVL=171.2
          ELSEIF(TIME.GT.71.AND.TIME.LE.72)THEN
               MILVL=171.4
          ELSEIF(TIME.GT.72.AND.TIME.LE.73)THEN
               MILVL=171.5
          ELSEIF(TIME.GT.73.AND.TIME.LE.74)THEN
               MILVL=171.6
          ELSEIF(TIME.GT.74.AND.TIME.LE.76)THEN
               MILVL=171.7
          ELSEIF(TIME.GT.76.AND.TIME.LE.77)THEN
               MILVL=171.8
          ELSEIF(TIME.GT.77.AND.TIME.LE.80)THEN
               MILVL=171.9
          ELSEIF(TIME.GT.80)THEN
               MILVL=172.0
          ENDIF
       END
C
C PI COMPENSATOR DATA(W/BACKLASH): FEEDWATER FLOW msiv closure
       SUBROUTINE MICOM2(TIME,MIFLO)
          IF(TIME.LE.0)THEN
               MIFLO=2018.
          ELSEIF(TIME.GT.0.AND.TIME.LE.1)THEN
               MIFLO=707.
          ELSEIF(TIME.GT.1.AND.TIME.LE.2)THEN
               MIFLO=617.
          ELSEIF(TIME.GT.2.AND.TIME.LE.3)THEN
               MIFLO=726.
          ELSEIF(TIME.GT.3.AND.TIME.LE.4)THEN
               MIFLO=882.
          ELSEIF(TIME.GT.4.AND.TIME.LE.5)THEN
               MIFLO=998.
          ELSEIF(TIME.GT.5.AND.TIME.LE.6)THEN
               MIFLO=1063.
          ELSEIF(TIME.GT.6.AND.TIME.LE.7)THEN
               MIFLO=1099.
          ELSEIF(TIME.GT.7.AND.TIME.LE.8)THEN
               MIFLO=1121.
          ELSEIF(TIME.GT.8.AND.TIME.LE.9)THEN
               MIFLO=1133.
          ELSEIF(TIME.GT.9.AND.TIME.LE.10)THEN
               MIFLO=1142.
          ELSEIF(TIME.GT.10.AND.TIME.LE.11)THEN
               MIFLO=1151.
          ELSEIF(TIME.GT.11.AND.TIME.LE.12)THEN
               MIFLO=1190.
          ELSEIF(TIME.GT.12.AND.TIME.LE.13)THEN
               MIFLO=1293.
          ELSEIF(TIME.GT.13.AND.TIME.LE.15)THEN
               MIFLO=1305.
          ELSEIF(TIME.GT.15.AND.TIME.LE.16)THEN
               MIFLO=1280.
          ELSEIF(TIME.GT.16.AND.TIME.LE.17)THEN
               MIFLO=1236.
          ELSEIF(TIME.GT.17.AND.TIME.LE.18)THEN
               MIFLO=1203.
          ELSEIF(TIME.GT.18.AND.TIME.LE.19)THEN
               MIFLO=1176.
          ELSEIF(TIME.GT.19.AND.TIME.LE.20)THEN
               MIFLO=1119.
          ELSEIF(TIME.GT.20.AND.TIME.LE.21)THEN
               MIFLO=1091.
          ELSEIF(TIME.GT.21.AND.TIME.LE.22)THEN
               MIFLO=1034.
          ELSEIF(TIME.GT.22.AND.TIME.LE.23)THEN
```

```
      MIFLO=986.
   ELSEIF(TIME.GT.23.AND.TIME.LE.24)THEN
      MIFLO=945.
   ELSEIF(TIME.GT.24.AND.TIME.LE.25)THEN
      MIFLO=886.
   ELSEIF(TIME.GT.25.AND.TIME.LE.26)THEN
      MIFLO=856.
   ELSEIF(TIME.GT.26.AND.TIME.LE.27)THEN
      MIFLO=797.
   ELSEIF(TIME.GT.27.AND.TIME.LE.28)THEN
      MIFLO=764.
   ELSEIF(TIME.GT.28.AND.TIME.LE.29)THEN
      MIFLO=718.
   ELSEIF(TIME.GT.29.AND.TIME.LE.30)THEN
      MIFLO=678.
   ELSEIF(TIME.GT.30.AND.TIME.LE.31)THEN
      MIFLO=650.
   ELSEIF(TIME.GT.31.AND.TIME.LE.32)THEN
      MIFLO=595.
   ELSEIF(TIME.GT.32.AND.TIME.LE.33)THEN
      MIFLO=570.
   ELSEIF(TIME.GT.33.AND.TIME.LE.34)THEN
      MIFLO=532.
   ELSEIF(TIME.GT.34.AND.TIME.LE.35)THEN
      MIFLO=500.
   ELSEIF(TIME.GT.35.AND.TIME.LE.36)THEN
      MIFLO=478.
   ELSEIF(TIME.GT.36.AND.TIME.LE.37)THEN
      MIFLO=438.
   ELSEIF(TIME.GT.37.AND.TIME.LE.38)THEN
      MIFLO=418.
   ELSEIF(TIME.GT.38.AND.TIME.LE.39)THEN
      MIFLO=392.
   ELSEIF(TIME.GT.39.AND.TIME.LE.40)THEN
      MIFLO=356.
   ELSEIF(TIME.GT.40.AND.TIME.LE.41)THEN
      MIFLO=342.
   ELSEIF(TIME.GT.41.AND.TIME.LE.42)THEN
      MIFLO=324.
   ELSEIF(TIME.GT.42.AND.TIME.LE.43)THEN
      MIFLO=291.
   ELSEIF(TIME.GT.43.AND.TIME.LE.44)THEN
      MIFLO=276.
   ELSEIF(TIME.GT.44.AND.TIME.LE.45)THEN
      MIFLO=261.
   ELSEIF(TIME.GT.45.AND.TIME.LE.46)THEN
      MIFLO=246.
   ELSEIF(TIME.GT.46.AND.TIME.LE.47)THEN
      MIFLO=220.
   ELSEIF(TIME.GT.47.AND.TIME.LE.48)THEN
      MIFLO=208.
   ELSEIF(TIME.GT.48.AND.TIME.LE.49)THEN
      MIFLO=196.
   ELSEIF(TIME.GT.49.AND.TIME.LE.50)THEN
      MIFLO=174.
   ELSEIF(TIME.GT.50.AND.TIME.LE.51)THEN
      MIFLO=164.
   ELSEIF(TIME.GT.51.AND.TIME.LE.52)THEN
      MIFLO=153.
   ELSEIF(TIME.GT.52.AND.TIME.LE.53)THEN
      MIFLO=143.
   ELSEIF(TIME.GT.53.AND.TIME.LE.54)THEN
      MIFLO=132.
   ELSEIF(TIME.GT.54.AND.TIME.LE.55)THEN
      MIFLO=123.
   ELSEIF(TIME.GT.55.AND.TIME.LE.56)THEN
```

```
          MIFLO=111.
      ELSEIF(TIME.GT.56.AND.TIME.LE.57)THEN
          MIFLO=98.
      ELSEIF(TIME.GT.57.AND.TIME.LE.58)THEN
          MIFLO=91.
      ELSEIF(TIME.GT.58.AND.TIME.LE.59)THEN
          MIFLO=84.
      ELSEIF(TIME.GT.59.AND.TIME.LE.60)THEN
          MIFLO=76.
      ELSEIF(TIME.GT.60.AND.TIME.LE.61)THEN
          MIFLO=69.
      ELSEIF(TIME.GT.61.AND.TIME.LE.62)THEN
          MIFLO=61.
      ELSEIF(TIME.GT.62.AND.TIME.LE.63)THEN
          MIFLO=43.
      ELSEIF(TIME.GT.63.AND.TIME.LE.64)THEN
          MIFLO=34.
      ELSEIF(TIME.GT.64.AND.TIME.LE.65)THEN
          MIFLO=25.
      ELSEIF(TIME.GT.65.AND.TIME.LE.66)THEN
          MIFLO=7.
      ELSEIF(TIME.GT.66)THEN
          MIFLO=0.
      ENDIF
   END
```

```
$STORAGE:2
$NOFLOATCALLS
C PCTRAN6.FOR
C   PI COMPENSATOR DATA(W/BACKLASH): LEVEL(INCHES TAF) reactor trip
      SUBROUTINE RICOM1(TIME,RILVL)
         IF(TIME.LE.0)THEN
              RILVL=160.0
         ELSEIF(TIME.GT.0.AND.TIME.LE.1)THEN
              RILVL=156.6
         ELSEIF(TIME.GT.1.AND.TIME.LE.2)THEN
              RILVL=153.6
         ELSEIF(TIME.GT.2.AND.TIME.LE.3)THEN
              RILVL=151.0
         ELSEIF(TIME.GT.3.AND.TIME.LE.4)THEN
              RILVL=148.7
         ELSEIF(TIME.GT.4.AND.TIME.LE.5)THEN
              RILVL=146.7
         ELSEIF(TIME.GT.5.AND.TIME.LE.6)THEN
              RILVL=144.8
         ELSEIF(TIME.GT.6.AND.TIME.LE.7)THEN
              RILVL=143.0
         ELSEIF(TIME.GT.7.AND.TIME.LE.8)THEN
              RILVL=141.0
         ELSEIF(TIME.GT.8.AND.TIME.LE.9)THEN
              RILVL=139.1
         ELSEIF(TIME.GT.9.AND.TIME.LE.10)THEN
              RILVL=136.9
         ELSEIF(TIME.GT.10.AND.TIME.LE.11)THEN
              RILVL=137.7
         ELSEIF(TIME.GT.11.AND.TIME.LE.12)THEN
              RILVL=140.0
         ELSEIF(TIME.GT.12.AND.TIME.LE.13)THEN
              RILVL=142.4
         ELSEIF(TIME.GT.13.AND.TIME.LE.14)THEN
              RILVL=144.7
         ELSEIF(TIME.GT.14.AND.TIME.LE.15)THEN
              RILVL=146.9
         ELSEIF(TIME.GT.15.AND.TIME.LE.16)THEN
              RILVL=149.0
         ELSEIF(TIME.GT.16.AND.TIME.LE.17)THEN
              RILVL=150.9
         ELSEIF(TIME.GT.17.AND.TIME.LE.18)THEN
              RILVL=152.8
         ELSEIF(TIME.GT.18.AND.TIME.LE.19)THEN
              RILVL=154.5
         ELSEIF(TIME.GT.19.AND.TIME.LE.20)THEN
              RILVL=156.1
         ELSEIF(TIME.GT.20.AND.TIME.LE.21)THEN
              RILVL=157.6
         ELSEIF(TIME.GT.21.AND.TIME.LE.22)THEN
              RILVL=159.0
         ELSEIF(TIME.GT.22.AND.TIME.LE.23)THEN
              RILVL=160.3
         ELSEIF(TIME.GT.23.AND.TIME.LE.24)THEN
              RILVL=161.5
         ELSEIF(TIME.GT.24.AND.TIME.LE.25)THEN
              RILVL=162.6
         ELSEIF(TIME.GT.25.AND.TIME.LE.26)THEN
              RILVL=163.7
         ELSEIF(TIME.GT.26.AND.TIME.LE.27)THEN
              RILVL=164.6
         ELSEIF(TIME.GT.27.AND.TIME.LE.28)THEN
              RILVL=165.4
         ELSEIF(TIME.GT.28.AND.TIME.LE.29)THEN
              RILVL=166.2
         ELSEIF(TIME.GT.29.AND.TIME.LE.30)THEN
```

```
            RILVL=166.9
         ELSEIF(TIME.GT.30.AND.TIME.LE.31)THEN
            RILVL=167.6
         ELSEIF(TIME.GT.31.AND.TIME.LE.32)THEN
            RILVL=168.2
         ELSEIF(TIME.GT.32.AND.TIME.LE.33)THEN
            RILVL=168.7
         ELSEIF(TIME.GT.33.AND.TIME.LE.34)THEN
            RILVL=169.2
         ELSEIF(TIME.GT.34.AND.TIME.LE.35)THEN
            RILVL=169.6
         ELSEIF(TIME.GT.35.AND.TIME.LE.36)THEN
            RILVL=170.0
         ELSEIF(TIME.GT.36.AND.TIME.LE.37)THEN
            RILVL=170.3
         ELSEIF(TIME.GT.37.AND.TIME.LE.38)THEN
            RILVL=170.6
         ELSEIF(TIME.GT.38.AND.TIME.LE.39)THEN
            RILVL=170.9
         ELSEIF(TIME.GT.39.AND.TIME.LE.40)THEN
            RILVL=171.1
         ELSEIF(TIME.GT.40.AND.TIME.LE.41)THEN
            RILVL=171.4
         ELSEIF(TIME.GT.41.AND.TIME.LE.42)THEN
            RILVL=171.8
         ELSEIF(TIME.GT.42.AND.TIME.LE.43)THEN
            RILVL=172.2
         ELSEIF(TIME.GT.43.AND.TIME.LE.44)THEN
            RILVL=172.5
         ELSEIF(TIME.GT.44.AND.TIME.LE.45)THEN
            RILVL=172.8
         ELSEIF(TIME.GT.45.AND.TIME.LE.46)THEN
            RILVL=173.0
         ELSEIF(TIME.GT.46.AND.TIME.LE.47)THEN
            RILVL=173.3
         ELSEIF(TIME.GT.47.AND.TIME.LE.48)THEN
            RILVL=173.5
         ELSEIF(TIME.GT.48.AND.TIME.LE.49)THEN
            RILVL=173.6
         ELSEIF(TIME.GT.49.AND.TIME.LE.50)THEN
            RILVL=173.8
         ELSEIF(TIME.GT.50.AND.TIME.LE.51)THEN
            RILVL=173.9
         ELSEIF(TIME.GT.51.AND.TIME.LE.53)THEN
            RILVL=174.0
         ELSEIF(TIME.GT.53.AND.TIME.LE.56)THEN
            RILVL=174.1
         ELSEIF(TIME.GT.56.AND.TIME.LE.77)THEN
            RILVL=174.2
         ELSEIF(TIME.GT.77.AND.TIME.LE.107)THEN
            RILVL=174.1
         ELSEIF(TIME.GT.107)THEN
            RILVL=174.0
         ENDIF
      END
C
C PI COMPENSATOR DATA(W/BACKLASH): FEEDWATER FLOW reactor trip
      SUBROUTINE RICOM2(TIME,RIFLO)
         IF(TIME.LE.0)THEN
            RIFLO=2022.
         ELSEIF(TIME.GT.0.AND.TIME.LE.1)THEN
            RIFLO=2078.
         ELSEIF(TIME.GT.1.AND.TIME.LE.2)THEN
            RIFLO=2125.
         ELSEIF(TIME.GT.2.AND.TIME.LE.4)THEN
            RIFLO=2157.
```

```
ELSEIF(TIME.GT.4.AND.TIME.LE.5)THEN
     RIFLO=2113.
ELSEIF(TIME.GT.5.AND.TIME.LE.6)THEN
     RIFLO=2064.
ELSEIF(TIME.GT.6.AND.TIME.LE.7)THEN
     RIFLO=1954.
ELSEIF(TIME.GT.7.AND.TIME.LE.8)THEN
     RIFLO=1883.
ELSEIF(TIME.GT.8.AND.TIME.LE.9)THEN
     RIFLO=1778.
ELSEIF(TIME.GT.9.AND.TIME.LE.10)THEN
     RIFLO=1707.
ELSEIF(TIME.GT.10.AND.TIME.LE.11)THEN
     RIFLO=1579.
ELSEIF(TIME.GT.11.AND.TIME.LE.12)THEN
     RIFLO=1419.
ELSEIF(TIME.GT.12.AND.TIME.LE.13)THEN
     RIFLO=1355.
ELSEIF(TIME.GT.13.AND.TIME.LE.14)THEN
     RIFLO=1272.
ELSEIF(TIME.GT.14.AND.TIME.LE.15)THEN
     RIFLO=1172.
ELSEIF(TIME.GT.15.AND.TIME.LE.16)THEN
     RIFLO=1084.
ELSEIF(TIME.GT.16.AND.TIME.LE.17)THEN
     RIFLO=1008.
ELSEIF(TIME.GT.17.AND.TIME.LE.18)THEN
     RIFLO=907.
ELSEIF(TIME.GT.18.AND.TIME.LE.19)THEN
     RIFLO=857.
ELSEIF(TIME.GT.19.AND.TIME.LE.20)THEN
     RIFLO=766.
ELSEIF(TIME.GT.20.AND.TIME.LE.21)THEN
     RIFLO=710.
ELSEIF(TIME.GT.21.AND.TIME.LE.22)THEN
     RIFLO=640.
ELSEIF(TIME.GT.22.AND.TIME.LE.23)THEN
     RIFLO=583.
ELSEIF(TIME.GT.23.AND.TIME.LE.24)THEN
     RIFLO=528.
ELSEIF(TIME.GT.24.AND.TIME.LE.25)THEN
     RIFLO=473.
ELSEIF(TIME.GT.25.AND.TIME.LE.26)THEN
     RIFLO=432.
ELSEIF(TIME.GT.26.AND.TIME.LE.27)THEN
     RIFLO=376.
ELSEIF(TIME.GT.27.AND.TIME.LE.28)THEN
     RIFLO=346.
ELSEIF(TIME.GT.28.AND.TIME.LE.29)THEN
     RIFLO=311.
ELSEIF(TIME.GT.29.AND.TIME.LE.30)THEN
     RIFLO=270.
ELSEIF(TIME.GT.30.AND.TIME.LE.31)THEN
     RIFLO=249.
ELSEIF(TIME.GT.31.AND.TIME.LE.32)THEN
     RIFLO=207.
ELSEIF(TIME.GT.32.AND.TIME.LE.33)THEN
     RIFLO=187.
ELSEIF(TIME.GT.33.AND.TIME.LE.34)THEN
     RIFLO=168.
ELSEIF(TIME.GT.34.AND.TIME.LE.35)THEN
     RIFLO=135.
ELSEIF(TIME.GT.35.AND.TIME.LE.36)THEN
     RIFLO=119.
ELSEIF(TIME.GT.36.AND.TIME.LE.37)THEN
     RIFLO=106.
```

```
    ELSEIF(TIME.GT.37.AND.TIME.LE.38)THEN
         RIFLO=80.
    ELSEIF(TIME.GT.38.AND.TIME.LE.39)THEN
         RIFLO=68.
    ELSEIF(TIME.GT.39.AND.TIME.LE.40)THEN
         RIFLO=55.
    ELSEIF(TIME.GT.40.AND.TIME.LE.41)THEN
         RIFLO=45.
    ELSEIF(TIME.GT.41.AND.TIME.LE.42)THEN
         RIFLO=21.
    ELSEIF(TIME.GT.42.AND.TIME.LE.43)THEN
         RIFLO=8.
    ELSEIF(TIME.GT.43)THEN
         RIFLO=0.
    ENDIF
END
```

REFERENCES

Agarwal, A., S. Shah. 1989. "Tools and Techniques for Advanced Control of Power Plants." p. 1-20.

ANSI/ISA Standard S77.42-1987. 1987. "Fossil Fuel Plant Feedwater Control System - Drum Type." Instrument Society of America.

ALSIM User's Guide. 1990. Version 3.01. New Jersey Institute of Technology Control Systems Software Package.

Astrom, K., B. Wittemark. 1990. *Computer-Controlled Systems: Theory and Design*. Second Edition. Prentice-Hall.

Carmichael, L., G. Toye, A. Ipakchi, M. Sievers. 1984. "Digital Feedwater Controller for a BWR: A Concpetual Design Study." Electric Power Research Institute Report NP-3323. p. 2.1-4.70, A.1-A.7.

Dukelow, S. 1979. *The Control of Boilers*. ISA Press. p. 125-134.

Eastman, M., K. Gaydos, K. Graham, M. Lipner, N. Mueller, C. Narsrallah, A. Negrus, R. Paris, W. Schaefer, J. Waclo, D. Woods. 1989. "Advanced PWR Steam Generator-Feedwater Digital Control System." Electric Power Research Institute Report NP-4919-LD. p. 3.1-3.68.

CSD User's Guide. 1985. Cybernetics Technology Inc. New Jersey Institute of Technology Control Systems Software Package.

Friedland, B. 1986. *Control System Design*. McGraw-Hill.

Gaydos, K., R. Paris, P. Gantchev. 1986. "An Optimal Control Strategy in the Design of a PWR Feedwater Controller." Electric Power Research Institute Report NP-4769-SR. p. 5.47-5.73.

Goto, K. 1973. "Lag/Lead Compensation of the Valve Controlled Feedwater System." General Electric Company.

Grimble, M. 1979. "Solution of the Stochastic Optimal Control Problem in the S-Domain for Systems with Time Delay." *Proc IEEE*. 126:697-704.

Karpeta, C., P. Stirsky, K. Volf, S. Roubal. 1973. "Simulation Studies of the Optimal Direct Digital Control of the A-1 Reactor." *Nuclear Power Plant Control and Instrum. Symp. Proc. Prague, Czechoslovakia*. p. 289-302.

Matrix X/PC User's Guide. 1990. Version 7.1. Integrated Systems, Inc.

May, M. 1970 "Oyster Creek Startup Test Results" General Electric Company Report NEDE-13109.

Nakamura, H., H. Akaike. 1981. "Statistical Identification for Optimal Control of Supercritical Thermal Power Plants." *Automatica*. 17:143-155.

Parent, M., P. McMorran. 1982. "Multivariable Control in Nuclear Power Stations: Optimal Control." Atomic Energy of Canada, Limited Report AECL-7244. p. 1-38.

Parent, M., P. McMorran. 1982. "Multivariable Control in Nuclear Power Stations: Order Reduction." Atomic Energy of Canada, Limited Report AECL-7245. p. 1-34.

Paris, R., J. Belz, W. Catullo, A. Gegus. 1989. "Advanced PWR Steam Generator Digital Feedwater Control Systems." Electric Power Research Institute Report NP-6497-M. p. 2.1-2.15.

Paris, R., K. Gaydos, J. Hill, S. Whitson, R. Wirkhala. 1990. "Implementation of an Advanced Digital Feedwater Control System at the Prairie Island Nuclear Generating Station." Electric Power Research Institute Report NP-6758. p. 2.3-2.6.

Po, C. 1989. "PCTRAN/B Version 2.5 Personal Computer Transient Analyzer - BWR." Vol. 1 and 2.

Popovic, J., M. Hammer. 1987. "Testing and Installation of a BWR Digital Feedwater Control System." Electric Power Research Institute Report NP-5524. p. 2.1-2.15, 5.1-5.4, B.3-B.56.

Seishiro, K., M. Makino, T. Okutani, K. Hirayama. 1986. "Reliability of BWR Plant Digital Control." Electric Power Research Institute Report NP-4769-SR. p. 2.203-2.229.

Shah, S., L. Lehman, M. Sarchet. 1988. "Implementation Considerations for Digital Control Systems in Power

Plants." Electric Power Research Institute Report NP-6004. p. 3.1-3.2.

Singh, G., M. Tsiouris, L. Pearce, C. Burton. 1989. "Design and Operation of a Digital Low-Power Feedwater Control System for PWRs" Electric Power Research Institute Report NP-6149. p. 2.1-3.6.

Tylee, J. 1983. "On-Line Failure Detection in Nuclear Power Plant Instrumentation." *IEEE Trans. Automat. Contr.* AC-28:406-415.

Tye, C. 1980. "On-Line Computer Control of a Nuclear Reactor Using Optimal Control and State Estimation Methods." *Proc of the IFAC/IFIP Cont. 6th Duesseldorf, Germany.* p. 205-212.

Tysso, A. 1980. "Modern Control Theory Applied to Boiler Control. Proc. IEEE Conf. Decis. Control Inc. Symp. Adapt. Processes 19th Albuquerque, NM. p. 846-851.

Wallace, J., R. Clarke. 1983. "The Application of Kalman Filtering Estimation Techniques in Power Station Control Systems." *IEEE Trans. Automat. Contr.* AC-28:416-427.

Woods, P. 1968 "Oyster Creek Final Transient Analysis Report." General Electric Company Report GECR-5596.

Uchida, K., E. Shimemwa. 1986. "Closed Loop Properties of the Infinite-Time Linear-Quadratic Optimal Regulator for Systems with Delays." *Int. J. Contr.* 4:773-779.