12-1994

A PC-Based Signal Validation System for Nuclear Power Plants

Ali Seyfettin Erbay
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a thesis written by Ali Seyfettin Erbay entitled "A PC-Based Signal Validation System for Nuclear Power Plants." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Nuclear Engineering.

Belle R. Upadhyaya, Major Professor

We have read this thesis and recommend its acceptance:

Robert E. Uhrig, Jack F. Wasserman

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Ali Seyfettin Erbay entitled "A PC-Based Signal Validation System for Nuclear Power Plants." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Nuclear Engineering.

Belle R. Upadhyaya

Belle R. Upadhyaya, Major Professor

We have read this thesis and recommended its acceptance:

Robert E. Uhrig

Jack Wasserman

Accepted for the council:

Associate Vice Chancellor and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of the Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature A. Seyfettin Erbay

Date October 27, 1994

A PC-BASED SIGNAL VALIDATION SYSTEM

FOR NUCLEAR POWER PLANTS

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Ali Seyfettin Erbay

December 1994

DEDICATION

This thesis is dedicated to my teachers.

ACKNOWLEDGMENTS

I would like to thank my major professor, Dr. Belle R. Upadhyaya, for his invaluable guidance and suggestions, and my other committee members, Dr. Robert E. Uhrig, and Dr. Jack F. Wassserman, and the department head, Dr. Thomas W. Kerlin, for their comments and assistance. The cooperation of my teammates, Kadir Kavaklioglu and Evren Eryürek is gratefully acknowledged. The patience and understanding of my parents Rauf Aral and Gülay, and my sister Gül, have provided great motivation during this study.

I would also like to thank the Analysis and Measurement Services Corporation for providing valuable operational power plant data.

This research was sponsored by the Tennessee Valley Authority under contract TVA TV-84395V with The University of Tennessee. This assistance is gratefully acknowledged.

ABSTRACT

The safe operation and efficient control of a nuclear power plant requires reliable information about the state of the process. Therefore the validity of sensors which measure the process variables is of great importance. Signal validation is the detection, isolation and characterization of faulty signals. Properly validated process signals are also beneficial from the standpoint of increased plant availability and reliability of operator actions.

In recent years, several methods have been developed for signal validation (SV). Some of these methods include generalized consistency checking (GCC), process empirical modeling (PEM) for prediction, multi-dimensional process hypercube (PHC), univariate and multivariate autoregression modeling, and expert systems. The purpose of this research is to investigate the effectiveness of a few other techniques such as artificial neural networks (ANN) and extended Kalman filters for signal estimation during steady-state as well as transient operating conditions. The new and improved signal validation modules were integrated into one computer program for easy access. The final decision about the validity of signals was made using a fuzzy logic algorithm.

The integrated system consists of the following modules:

- Generalized Consistency Checking (GCC),
- Process Empirical Modeling (PEM),

- Artificial Neural Network (ANN) prediction, and
- Kalman Filtering Technique (KFT).

These modules operate in parallel and the system architecture is flexible for adding or removing a SV module.

The integrated system utilizes modern graphical user interface (GUI) techniques for displaying and accessing information. Due to the popularity and the increase in computing power and the decrease in the cost of PC's, nuclear power plants are also incorporating PC's into their engineering divisions to access process data over local area networks (LAN). The software in this study was therefore developed on an IBM compatible PC operating under Microsoft Windows 3.1™. Hypertext buttons, compatible with different aspects of Microsoft Windows 3.1™, were provided in parts of the GUI, for displaying the processed information and the results. The dynamic form of the empirical modeling and the Kalman filtering technique showed superior performance in signal validation.

The implementational details of the system were evaluated off-line, using steady-state and transient data from operating pressurized water reactor (PWR) nuclear power plants. The application of this new system was illustrated for a U-tube steam generator (UTSG) of a PWR nuclear power plant. A system executive was developed for controlling the functions of various modules, interfacing the input-output (I/O) with the environment, and for decision making. The use of new modules, improvement in the previous techniques, and the use of GUI have resulted in a robust and easily implementable signal validation system for power plants.

TABLE OF CONTENTS

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

LIST OF ACRONYMS

ANN	:	Artificial Neural Network
DDE	:	Dynamic Data Exchange
D/E	:	Decision / Estimator
DLL	:	Dynamic Link Library
EBR	:	Experimental Breeder Reactor
GCC	:	Generalized Consistency Checking
GUI	:	Graphical User Interface
HD	:	Hard Disk
I/O	:	Input / Output
KFT	:	Kalman Filtering Technique
LAN	:	Local Area Network
LLR	:	Logarithmic Likelihood Ratio
MDI	:	Multiple Document Interface
MISO	:	Multiple-Input Single-Output
OLE	:	Object Linked Embedding
PC	:	Personal Computer
PE	:	Processing Element
PEM	:	Process Empirical Modeling
PWR	:	Pressurized Water Reactor
RCS	:	Reactor Coolant System

RMSE : Root-Mean-Square Error

SNP : Sequoyah Nuclear Plant

SPRT : Sequential Probability Ratio Test

SV : Signal Validation

TVA : Tennessee Valley Authority

UTSG : U-Tube Steam Generator

Chapter 1

INTRODUCTION

1.1 Statement of the Problem

In order to achieve the desired operating configuration in any process, the system conditions must be measured. Examples of measurements are temperature, pressure, flow, level, motor current, vibration, etc. However, in order to operate within desired limits, it is important to know the reliability of plant measurements. Signal validation (SV) deals with this issue, and is defined as the detection, isolation and characterization of faulty signals. Also referred to as fault detection, signal validation checks inconsistencies among redundant measurements and estimates their expected values using other measurements and system models.

The benefits of signal validation are both economic and safety related. Catastrophic signal failure can result in plant shutdown and lost revenue. Pre-catastrophic failure detection would therefore minimize plant downtime and increase plant availability. The control action taken depends primarily upon the information provided by the plant instruments. Thus, increased plant productivity and increased reliability of operator actions, would result from the implementation of such a system.

The purpose of this study is to investigate some of the existing signal validation methods by incremental improvements and to develop new modules. Each of the SV modules performs a specific task. The architecture consists of four modules, an information base and a system executive integrated with a graphical user interface (GUI). The following four modules were integrated in the new PC-based system.

- Generalized Consistency Checking (GCC),
- Process Empirical Modeling (PEM),
- Artificial Neural Network (ANN) prediction, and
- Kalman Filtering Technique (KFT).

The primary advantage of using different SV algorithms is to compensate for prediction errors during transient operating conditions, in which some SV modules may not give good estimations of the measured variables.. Another potential benefit is to have software redundancy, so that false alarms may be reduced. These modules operate in parallel and the system architecture is flexible for adding or removing a SV module.

All the modules are used for validation during both steady-state and transient operating conditions. The entire system was developed in the PC-framework under Microsoft Windows 3.1™. Some improvements were made in the structure of static data-driven models by incorporating one and two-step regression. Kalman filtering is based on the use of a physical model of plant components and was implemented for the first time for a steam generator system in nuclear power plants. This is applicable to both steady-state and transient operations.

The system executive performs several tasks: sequencing of module operation, requisition of additional data, evaluating SV information from the various modules, and displaying instrument or system status to the operator. The decision-making within the system executive was developed using a fuzzy logic approach. The computer display was performed by GUI objects compatible with Microsoft Windows 3.1™.

1.2 Review of Previous Work

An extensive research in the area of signal validation and fault detection had been performed in the past. The initial research focused on methods which use redundant signals for a given process variable to check for inter-signal consistency [1]. This method is known as the parity-space technique and is used in most of the nuclear power plants in the United States. The simplest method of consistency checking between three signals is to use an average of the signal. This method was expanded by adding analytical redundancy. Analytical redundancy is achieved by estimation of the process variable using a system model. Model equations are based on conservation of mass, energy and momentum. Nonlinear empirical models were also developed for generating signal redundancy [2]. One of the primary goals of analytical and empirical redundancy is the detection of common-cause failures.

Many applications of the SV technology are found in the aerospace and nuclear industries. Fault detection has been implemented in flight control systems and the space shuttle [3, 4, 5]. Signal validation techniques have been applied at the Experimental Breeder Reactor-II (EBR-II) and commercially at Northwest Utilities Millstone Units 2 and 3 [6, 7]. A system state analyzer has also been applied to the surveillance of the EBR-II [8]. Signal Validation has recently been incorporated into a digital feedwater control system in several North American nuclear power plants [9].

The following methodologies were developed at The University of Tennessee for SV application to nuclear power plants [10, 11, 12]:

- GCC using redundant process signals and empirical redundancy,
- Univariate autoregression modeling for wideband frequency analysis,
- PEM to detect measurement system drift,
- Multi-dimensional process hypercube comparison for data compression and for tracking instrument and process behavior,
- Bias and noise detection for basic signal changes, and
- Rule-based expert system for qualitative signal validation.

Each of the modules was developed both as a stand-alone system and as part of a comprehensive SV system.

In addition to the above mentioned techniques, ANN's have also been utilized at The University of Tennessee for monitoring, estimation and control purposes. Back-propagation algorithm has proven to be an effective training method in developing these ANN's.

Since the original publication of R. E. Kalman's paper in 1960, the Kalman Filtering Technique (KFT) has been studied and developed thoroughly in several areas [13]. Without the help of KFT the Apollo mission to the moon could not have been successful. Aeronautics, flight engineering and missile tracking systems use KFT for tracking and navigational control. It is used in signal processing to solve system identification and deconvolution problems of linear systems. It has also found a large area of applications in communication and control. KFT is applied in geophysics for seismic signal processing [14].

Local sensor monitoring was also addressed by several investigators [15,16]. This assumes no sensor redundancy and no model-based independent estimation. An individual signal characterization and the availability of a sensor knowledge base are used in this approach.

Fuzzy logic has become one of the most commonly used techniques in control and decision-making with applications from a simple camcorder to a nuclear power plant (Fugen nuclear power reactor in Japan). The advent of fuzzy logic technology has offered another opportunity for signal processing and validation. The features offered by fuzzy logic can lend themselves to a more reliable and perhaps fault-tolerant approach. The fuzzy logic methodology for fault-tree analysis was previously developed at The University of Tennessee [17]. In the present work, it was incorporated for decision-making in the system executive module.

The current trend towards graphics and the use of visual images are among the important developments of this decade, not only for technical personnel using computers but also for nontechnical users. Sensory immersion, such as that provided by virtual reality, is becoming an option for understanding the underlying complex information. The widespread use of IBM-compatible personal computers (PC) and the low cost of high-performance chips made Microsoft Windows 3.1™ a very popular operating system which simulates multitasking-multiprocessing and uses modern graphical user interfaces (GUI) for custom control and display. Today, most of the nuclear power plant personnel uses PC's for applications from engineering computations to word processing.

1.3 Overview of the Methodology

The present study uses some of the previously developed techniques, as well as some newly developed modules such as KFT, ANN and fuzzy logic decision-making. Four modules were used for signal validation and state estimation: GCC, PEM, ANN and KFT.

The GCC module was included for a systematic check of consistencies among redundant signals measuring the same process variable. The algorithm provides information about measurement inconsistencies at each sampling time. The sequential probability ratio test (SPRT) was also included as part of this module to continuously check for sensor degradation, and to record the sensor degradation history [11].

The PEM module establishes nonlinear multiple-input single-output models. The measured sensor output is then compared against the predicted output estimated by the PEM model. Although the use of signal values at previous time instants is common in dynamic neural networks, in this study a dynamic PEM model was developed for the first time as part of the SV system. The performance of the dynamic empirical model is better than that of the static model.

ANN's are intrinsically parallel and non-algorithmic methods. The ability of the back-propagation method to learn any arbitrary nonlinear mapping from inputs to outputs, and the fault-tolerant property of a multi-layer network was utilized for the prediction of instrument outputs (state variables) to be validated [18, 19]. The PC-based signal validation system then compares the estimation against the measurement.

The Kalman filter, in general, uses a nonlinear system model to estimate system variables. However, the model may have uncertainities and may be less accurate bacause of a reduced model order. By using measurements as corrections to the prediction of the model, the KFT gives the proper estimate of the validated signal. The estimate is then compared against the measurement. The KFT module is developed using a nonlinear system model of a PWR U-tube steam generator (UTSG), previously developed at The University of Tennessee [20, 21, 22].

These four modules were integrated with a system executive, which makes the final decision according to the results of the modules. The decision-making algorithm uses a fault-tree approach to detect the faulty signal: if any of the SV modules reports a fault within the sensitivity of that module, the signal is marked as faulty. However, since each module has different sensitivity and design parameters, it is necessary to incorporate the differences in the decision making process. This was achieved very easily using fuzzy logic. Results of each module were converted to fuzzy sets, which can be defined as a possibility distribution of truthness of the sensor being faulty. The flexibility of defining this possibility distribution enables us to adjust the decision-making for different SV modules having different estimation characteristics. Then, each of these fuzzy sets is presented to the fault tree using fuzzy operations. The output of the fault tree is also a fuzzy set which is interpreted using prototype fuzzy sets such as *very bad*, *bad*, *medium*, *good* and *very good*.

Displaying the result of the decision-maker in a textual form may be confusing if the validated signals are many or are updated in such a short time interval that the user does not have enough time to read them. Modern GUI techniques make it possible to have a more flexible and innovative graphical display. Virtual objects placed on the information window make it easy to display complex results and navigate through the information space.

1.4 Contributions of the Thesis

The major accomplishment of this research is the design and implementation of a PC-based signal validation system for a UTSG. Many of the SV systems, incorporated in nuclear power plants are part of a comprehensive and complex main program. The development of a separate SV module, which was flexible in design and execution, was incorporated into the end-user PC's that operate under Microsoft Windows 3.1™.

Another accomplishment of this project is the design of two new signal validation modules: ANN and KFT. Several issues were considered in creating ANN models. Input-output signal selection, selection of training data, selection of network structure and training algorithm were some of these issues. In developing the KFT, previously developed models were used for prediction of the state variables. However, since the system is nonlinear in nature, the system equations were discretized so that the extended Kalman filter could be applied. Time-dependency is incorparated in the dynamic version of the PEM module. This provides a better estimation, especially when time lags exist among signals.

A new decision-making algorithm was also developed as part of this thesis. The use of a fuzzy logic methodology for fault-tree analysis enables the system to adapt itself to different sensitivities of the SV modules, and provides a quality index to the measurement. The output of the fuzzy logic fault-tree is displayed in graphical icons, which are easy to be interpreted and recognized by the end-user.

Finally, the off-line developed SV system was integrated on the Local Area Network (LAN) of an operating PWR nuclear power plant. An interface program was developed for this purpose in order to transfer live sensor data to end-user PC's.

1.5 Organization of the Thesis

The descriptions and algorithms of the SV modules are given in Chapter 2. Since GCC, SPRT, PEM and ANN are well-known techniques they are covered in one chapter. The KFT is a newly developed SV module and is described in Chapter 3.

Chapter 3 gives an introduction to the classical Kalman filter, orginally established for linear systems. Since most of the real-world applications are nonlinear by nature, the extended Kalman filter was described for such systems. Some issues for implementational consideration of KFT are also discussed.

In Chapter 4 a description of the UTSG is given. Model equations and control system equations are part of the prediction of the KFT module.

Chapter 5 describes the integration of the SV modules. A description of the fuzzy logic reasoning and the application to fault-tree methodology is given in this chapter. Other system executive components such as GUI and Input / Output are also explained in detail.

The results of application of the SV modules and the decision-making procedure using data from operational nuclear plants, are discussed in Chapter 6.

Summary, conclusions and recommendations for future work are presented in Chapter 7.

Chapter 2

DESCRIPTION OF THE SIGNAL VALIDATION MODULES

2.1 Introduction

The PC-based signal validation system consists of four different modules:

- Generalized Consistency Checking (GCC),
- Process Empirical Modeling (PEM),
- Artificial Neural Network (ANN) prediction, and
- Kalman Filtering Technique (KFT).

The first three modules were thoroughly investigated and developed at The University of Tennessee for several applications including signal validation, state estimation, monitoring and control [11, 23, 24]. In this chapter a brief description of these modules and changes to enhance their performances are given. The reader may refer to additional sources for derivation of equations and other forms of these modules.

Although Kalman filtering was developed and studied for three decades, its use in signal validation for a UTSG is new. In order to emphasize this aspect, KFT is explained separately in Chapter 3.

2.2 Generalized Consistency Checking (GCC) and Sequential Probability Ratio Test (SPRT)

The GCC and SPRT techniques were developed previously at The University of Tennessee and applied to a signal validation system [2, 10, 11]. GCC is a method for the systematic cross comparison of signals from redundant sensors measuring the same process variable. The algorithm provides information about measurement inconsistencies at each sampling instant. After excluding the signals with maximum inconsistency indices, the best estimate at any time is computed as a weighted average of the remaining signals. The procedure is then repeated for subsequent sampling instants. The algorithm does not make comparisons between sets of measurements at different times. Any two redundant measurements are defined to be inconsistent if the difference between their values is greater than a specified threshold value. This threshold value depends on the selected signal pair and is based on sensor tolerances or technical specifications. The inconsistency indices of the individual measurements and the best estimate for the given process variable are determined as functions of sampling time instants.

The availability of sufficient redundancy is an important requirement for this SV method. If only one signal is available, the SV is limited to the observation of unusual behavior by checking the changes in the sensor time constant and signal-to-noise ratio. In case of duplex redundancy, the algorithm is capable of detecting a sensor failure, but not the identification of the failure itself. The triple redundancy provides the capability of detecting and identifying the sensor failure. The process variable is also reconstructed in the form of a weighted average of the readings from redundant sensors after excluding the most inconsistent measurements.

To achieve the required levels of redundancy, a redundant array of like sensors is used when they are available. If direct or hardware redundancy is not available, carefully validated and tuned analytical models may be used to provide estimates of process variables. The analytical redundancies are obtained from physical or empirical relationships that exist among variables in the system measured by dissimilar sensors. Physical models representing mass, energy or momentum balances, or system description in the form of differential equations, have fixed structure or functional forms so that they fit only to a specific system component. A schematic of the GCC algorithm having analytical measurement diversity is shown in Figure 2.1.

The output of the decision / estimator (D/E) unit at a given time instant consists of the error messages to the user, the different error parameters (inconsistency and exclusion indices, SPRT parameters), and the estimate of the process variable based on the consistent subset of signals. The number of inputs to the D/E may vary and reaching an estimate is still possible even with one or more degraded input signals. However, a minimum of three signals is required to identify a faulty signal, and to obtain a reliable estimate [11].

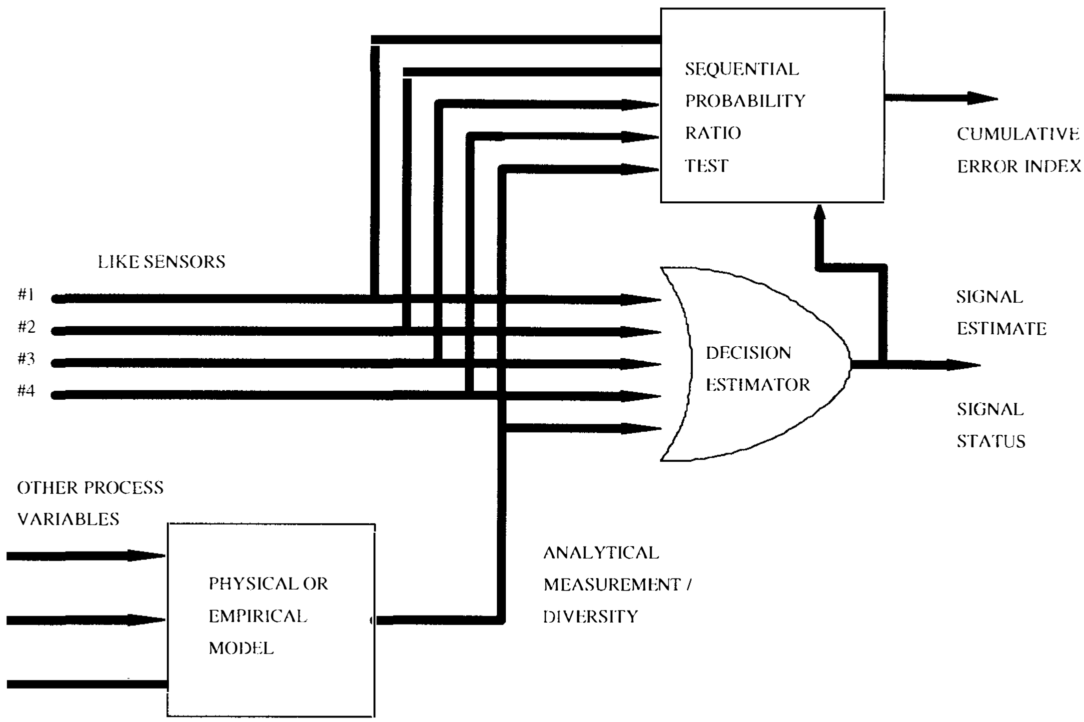

Figure 2.1: GCC module for single variable showing the decision / estimation and the SPRT units.

The first part of the algorithm determines the degrees of consistency among a given set of measurements at time instant k. A pairwise comparison of the measurements is made based on the individual sensor system tolerances. An inconsistency index (I_i) is computed for each signal. This index is used to exclude signals to determine a best estimate of the signal. In the event that the redundant group is partially consistent the estimate is computed from a weighted summation using the inconsistency index of each signal.

At time instant t, any two like measurements $m_i(t)$ and $m_j(t)$ are said to be consistent with respect to each other if

$$|m_i(t) - m_j(t)| \leq \varepsilon_i(t) + \varepsilon_j(t) \tag{2.1}$$

where ε is the error tolerance of each signal, respectively.

If Equation (2.1) is not satisfied, the signals are said to be inconsistent with each other. The error indices of both signals (I_i and I_j) are increased by one each time the given signal pair is inconsistent. This comparison is performed for all possible signal pairings. This error index, ranges between zero to ($n-1$), where n is the number of redundant signals. Further management and isolation of faulty readings is based on:

- The values of maximum (I_{max}) and minimum (I_{min}) error indices and
- The number of signals having the maximum (N_{max}) and minimum (N_{min}) error index.

Depending on the values of I_{min}, I_{max}, N_{min} and N_{max} and the individual inconsistency indices, the best estimate for the process variable is calculated directly, or only after repeating the whole reasoning with elimination of the most faulty signals. If $I_{max} = 0$, all measurements are consistent and their average is the estimate. If $I_{max} > 0$, but $I_{min} = 0$, the signals are partially inconsistent and the estimate is calculated as a weighted average. If $I_{max} > 0$ and $I_{min} > 0$, then N_{max} signals will be isolated as faulty and excluded from further calculations or may be given low weight. If all signals are inconsistent, that is $N_{max} = n\text{-}1$, then no estimate is possible on the basis of the current observation.

The estimate ($\tilde{x}(t)$) at the current sample is calculated using only fully or partially consistent signals.

$$\tilde{x}(t) = \frac{\sum_{i=1}^{n} w_i m_i(t)}{\sum_{i=1}^{n} w_i} \tag{2.2}$$

where $w_i = n\text{-}1\text{-}I_i$.

The SPRT has the ability to check and record sensor degradation. The SPRT makes decisions on the basis of cumulative information provided by the measurement history. Contrary to the GCC method, the SPRT does not make intersignal comparison or consistency checking among the signals.

The SPRT is an optimal decision-making procedure and requires a minimum number of samples from a sensor to make decisions based on specified missed- and false-alarm probabilities. These quantities provide a measure of confidence for the decision. The SPRT is applied to the difference between the sensor output and the estimated value of the process variable. The estimate is obtained from the GCC algorithm.

Let the measured value of a process variable (sensor value) be *m(t)* at time instant *t* and let the estimate of the process variable be $\tilde{x}(t)$ at the same time instant. The measurement residual $s(t) = m(t) - \tilde{x}(t)$ is computed at each sample during normal operation in order to determine a mean μ_0 and a variance σ_0^2. These define the Gaussian density function modeling of the normal mode signal error

$$p_0 = p(s;\mu_0,\sigma_0^2) = \frac{1}{\sqrt{2\pi\sigma_0^2}}\exp\left(-\frac{(s-\mu_0)^2}{2\sigma_0^2}\right) \quad (2.3)$$

For a normal sensor the mean of the error should be zero. The sensor failure can be detected by a change in the mean value (μ_0) or a change in the variance (σ_0^2). Failure thresholds in terms of mean value and variance are defined, and the corresponding Gaussian density functions $p_1 = p(s;\mu_1,\sigma_1^2)$ model the output statistics of degraded sensors. The approach used in this study is based on SPRT of the normal mode against an alternate degraded mode, assuming that both modes can be characterized by Gaussian distributions.

The SPRT uses recursive calculations of the logarithm of the likelihood ratio (LLR) function λ_n, representing the degradation information of a sensor based on n samples:

$$\lambda_n = \ln\left(\frac{p(s_1, s_2, \ldots, s_n | \mu_1, \sigma_1^2)}{p(s_1, s_2, \ldots, s_n | \mu_0, \sigma_0^2)}\right) \tag{2.4}$$

The LLR is updated at each sampling time, substituting the new sensor error s_{n+1} into the functional form of LLR's. Its value is compared against two boundaries ($A < 0$ and $B > 0$) derived from the specified error probabilities of false (α) and missed (β) alarms:

$$A = \ln\left(\frac{\beta}{1-\alpha}\right) \tag{2.5}$$

$$B = \ln\left(\frac{1-\beta}{\alpha}\right) \tag{2.6}$$

For a normal sensor, the ratio would decrease and finally reach the specified bound $A < 0$. Then the decision is made that the sensor is normal and the ratio is initialized by setting it equal to zero. The continuous version of the increase / decrease of the ratio for a sensor can be represented by a stochastic diffusion process drifting between two boundaries.

The sensor errors are assumed to be independent for each sample so that the density function in the expression for the LLR is obtained by multiplying the individual Gaussian density functions. Rearranging Equation (2.4), the recursive procedure becomes

$$\lambda_n = \lambda_{n-1} + \ln\left(\frac{p(s_n | \mu_1, \sigma_1^2)}{p(s_n | \mu_0, \sigma_0^2)}\right) \tag{2.7}$$

The general form of the LLR can be simplified in two ways to distinguish between bias degradation and excessive noise degradation. First, to check only the bias degradation ($\sigma_1 = \sigma_0 = \sigma$), the expression for the LLR reduces to (with $\mu_0 = 0$):

$$\lambda_n = \lambda_{n-1} + \frac{\mu_1}{\sigma^2}\left(s_n - \frac{\mu_1}{2}\right) \tag{2.8}$$

Second, to detect only the excessive noise levels ($\mu_1 = \mu_0 = 0$), LLR reduces to the form

$$\lambda_n = \lambda_{n-1} + \frac{1}{2}\left(\frac{1}{\sigma_0^2} - \frac{1}{\sigma_1^2}\right)s_n^2 + \ln\left(\frac{\sigma_0}{\sigma_1}\right) \tag{2.9}$$

While checking for the bias degradation, instead of analyzing for both the normal mode and the degraded mode, an a-priori assumption is made that a sensor is in the normal mode at the beginning of the SPRT. Consequently, a drift of the LLR parameter in the negative direction does not provide any additional information, since this is expected from a sensor in normal operation. A control action of resetting the LLR parameter to zero is applied, every time the parameter has a negative value. The previously accumulated information is of no value. If the LLR drifts to values larger than zero, the sensor is more likely in the degraded mode and no control action is applied in this case. The detection of a degraded signal needs more measurements, thus the recursive calculation of the LLR parameter proceeds.

The parameters *A* and *B* in Equations (2.5) and (2.6) are based on specified false-alarm and missed-alarm probabilities respectively. Another method is to use the one-sided

SPRT which is related to the previous two-sided SPRT. In this case, a specified mean time (T) is used to determine a degradation threshold. The relationship between the new bound B^* and the specified mean time between false alarms (T) is given as

$$T = \frac{2\sigma^2}{\mu_1^2}\left(e^{B^*} - B^* - 1\right) \tag{2.10}$$

where B^* is given by

$$B^* = \ln\left(\frac{\mu_1^2}{2\sigma^2}T\right) \tag{2.11}$$

for large T.

The detection performance $\tau(T)$ is defined as the mean detection time for a specific mean time T between false alarms and has the form:

$$\tau(T) = \frac{2\sigma^2}{\mu_1^2}\left(\ln\left(\frac{\mu_1^2}{2\sigma^2}T\right) - \frac{3}{2}\right) \tag{2.12}$$

Maintaining the same specified mean time between false alarms, the relationship between the one-sided and two-sided systems is

$$e^{B^*} - B^* - 1 = A\frac{e^B - 1}{e^A - 1} - B \tag{2.13}$$

If the LLR for bias degradation is computed for a time greater than the specified mean time between false alarms and the LLR does not exceed the degradation bound, then the LLR is reinitialized to zero.

This one-sided approach developed for detecting bias degradation has several advantages:

- Checking the LLR parameter against lower bound $A < 0$ representing the normal state is not necessary.
- The extra time delay in detecting the degraded mode, due to the negative magnitude of the LLR accumulated under the normal mode, is eliminated when the control resetting to zero is applied. In this way, if only the degradation test is performed, less observation time is required.
- If a sensor degradation occurs, the probability of its ultimate detection is one.

The SPRT is performed for bias and noise degradations after the GCC analysis of the module is completed. Both of the algorithms are combined as one module (GCC) and produce several outputs for the use of decision-making:

- Estimate of the process variable,
- LLR for each signal,
- Inconsistency index for each signal, and
- Indication if the signal is excluded from calculations.

2.3 Process Empirical Modeling

The Process Empirical Modeling (PEM) module was developed and used previously at The University of Tennessee for SV applications [2, 10, 11]. The PEM establishes multiple-input single-output (MISO) models. The measured sensor output is then compared against a predicted output based on the PEM. The module provides an independent estimation of a process variable. Monitoring sensor degradation or drift by on-line monitoring of the sensor output is possible using the estimates of the PEM module.

The analytical measurement or prediction of a critical signal y as a function of related variables in a subsystem, during steady-state or quasi steady-state operations is given by

$$y = f(\underline{x}) = f(x_1, x_2, \ldots, x_n) \tag{2.14}$$

No assumption is made that the input variables are independent of the output variable.

The PEM creates an optimal nonlinear MISO model from a given data set. This data set has a similar function as the training data set used in ANN's. The form of the data-driven predictive model is

$$y = c_0 + \sum_{i=1}^{m} c_i \Phi_i(\underline{x}) \tag{2.15}$$

where

y = estimate of the process variable,

$\underline{x}$ = vector of input signals,

m = number of terms in the model,

Φ_i = nonlinear function of input signals, and

c_i = constant coefficient.

A geometrical description of the nonlinear modeling algorithm is as follows. The closest nonlinear cross-product vector relative to the output vector is determined. New nonlinear vectors are subsequently selected by projecting the remaining unselected vectors onto a subspace of the original vector space orthogonal to the selected vector [2]. This procedure continues until n terms are chosen. The final step is to fit a linear type model using the selected terms to compute the coefficients. A more complete mathematical description is given in Figure 2.2.

The modeling program varies both the polynomial order and the number of terms. The optimal model is selected such that the error between the predicted output and the measured output is less than a prescribed value. The overall fractional prediction error (ε_p) is defined as

$$\varepsilon_p^2 = \frac{1}{N}\sum_{k=1}^{N}\left(\frac{y_m(k) - y_p(k)}{y_p(k)}\right)^2 \tag{2.16}$$

where

N = number of measurements used in the fit,

$y_m(k)$ = measurement at time instant k, and

$y_p(k)$ = prediction at time instant k.

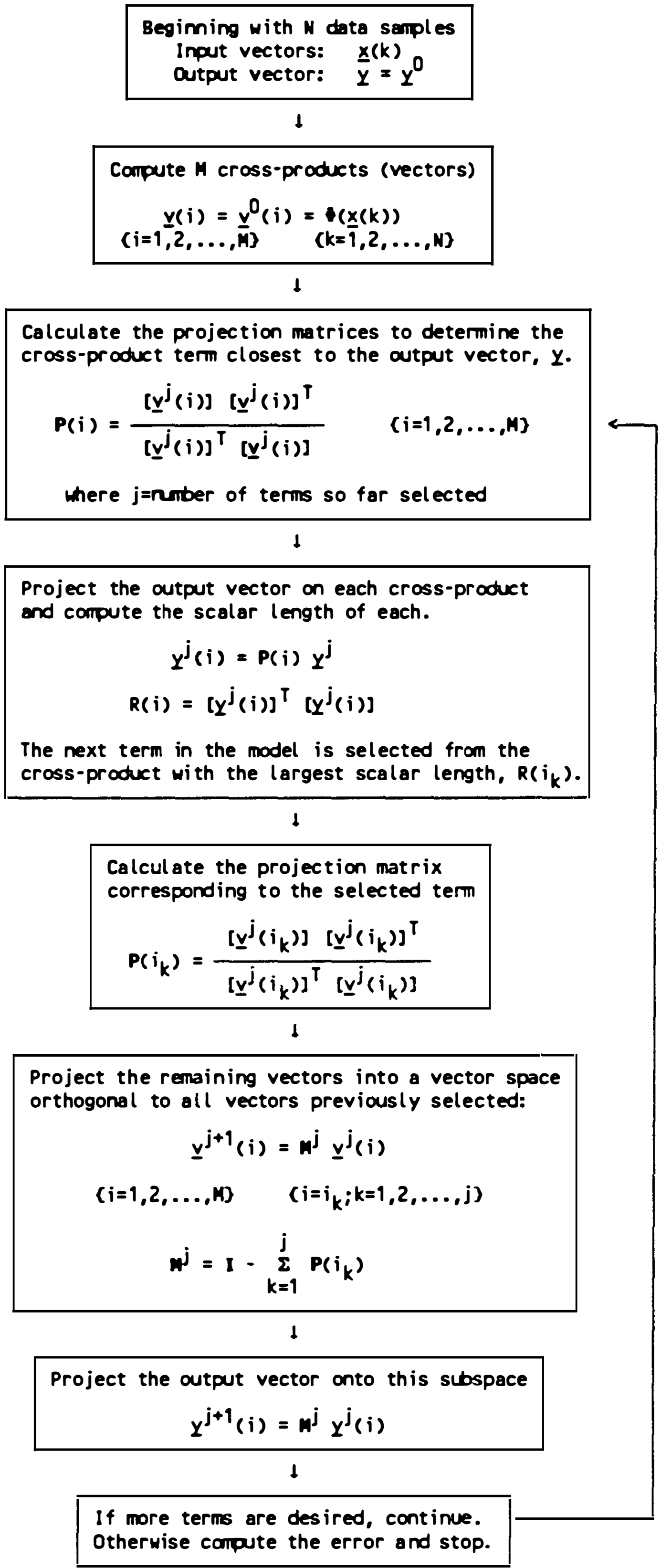

Figure 2.2: Process empirical modeling flow-chart [2].

The actual standard deviation of the prediction error of the signal at any time instant is estimated from:

$$\sigma_p(k) = \varepsilon_p y_p(k) \tag{2.17}$$

As a new contribution of this research, the static model given in Equation (2.14) is extended also for dynamic systems in which system variables can change significantly over time. A system may be modeled with a first order or a second order differential equation in the form:

$$\frac{dy}{dt} = f(\underline{x}) \tag{2.18}$$

or

$$\frac{dy}{dt} = f\left(\frac{d\underline{x}}{dt}, \underline{x}\right) \tag{2.19}$$

or

$$y = f\left(\frac{d\underline{x}}{dt}, \underline{x}\right) \tag{2.20}$$

Using finite-difference numerical technique, the first derivative is approximated as

$$\frac{dy}{dt} = \frac{y(t) - y(t-1)}{\Delta t} \tag{2.21}$$

where t denotes the discrete sampling time and Δt denotes the sampling time interval. Then Equation (2.18) can be converted from continuous time domain to discrete time domain as:

$$y(t) = f\left(\underline{x}(t), y(t-1)\right) \tag{2.22}$$

Equation (2.20) becomes

$$y(t) = f\left(\underline{x}(t), \underline{x}(t-1)\right) \tag{2.23}$$

Both forms of discrete representations (Equations (2.22) and (2.23)) were incorporated in the PEM module. The previous values of input and output vectors are treated as an additional input to the regular PEM model given in Equation (2.15). Thus, the algorithm remains the same for all forms of the model equation.

2.4 Artificial Neural Networks

Artificial neural networks (ANN) are computational models inspired by the architecture of the human nervous system. The basic unit in an ANN is analogous to a neuron in the human brain, and is called a processing element (PE). As shown in Figure 2.3, a PE has many connections to other PE's by means of input paths and output paths. The PE combines the weighted inputs usually by a simple summation. The combined input is then passed through a transfer function to produce the output of the PE. The output of the PE is called the activation value.

The output of a PE is connected to the input paths of other PE's through connection weights, so that the output of each PE is multiplied by the corresponding connection weights before reaching other PE's. In this manner, PE's combine the modified

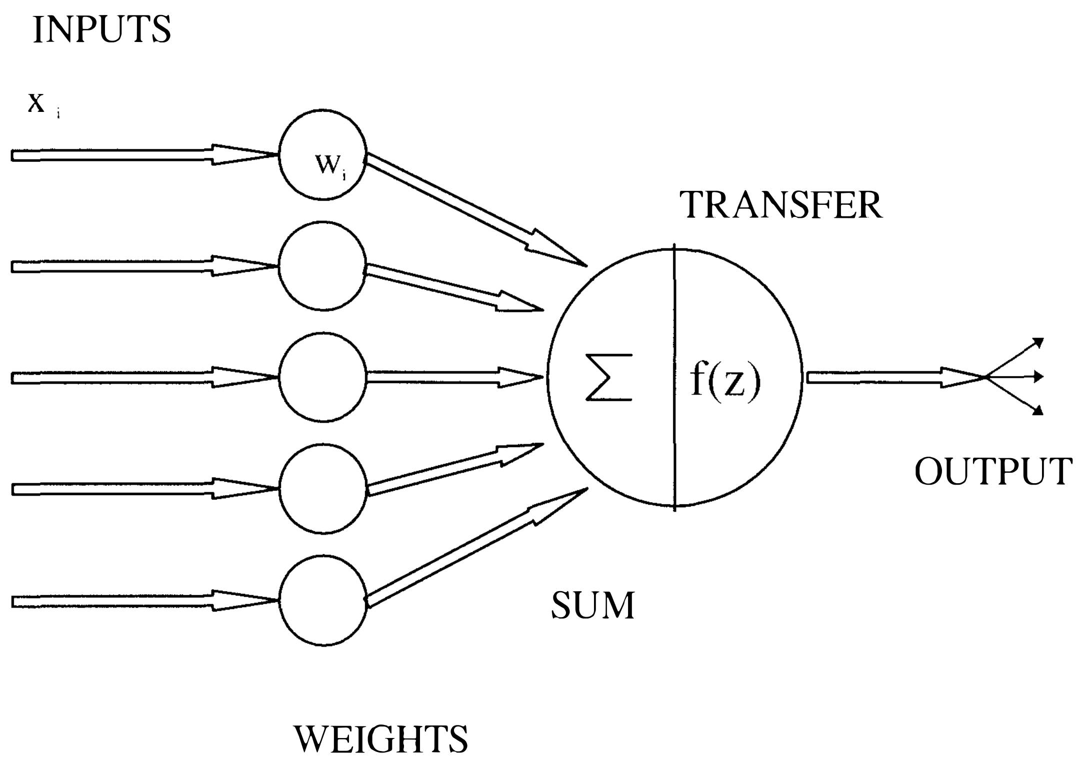

Figure 2.3: Schematic of a processing element.

activation values from other elements.

An ANN consists of many PE's interconnected in the above mentioned manner. PE's are grouped into layers or slabs which are designated in sequence from input to output layers. Then the connections are built between the neighboring layers. The input layer receives information from the outside, while the output layer transmits information to the outside. All the other layers in an ANN are called hidden layers. They receive and transmit information within the network. An ANN is said to be auto-associative if the output and the input information are identical. Otherwise, it is called hetero-associative.

An example of a three-layer topology of an auto-associative ANN is given in Figure 2.4. Figure 2.5 shows a three-layer hetero-associative ANN.

ANN's have mainly two phases of operation:

1. Learning, in which the ANN is constructed and developed.
2. Recall, in which the ANN performs its design function.

Many ANN's are classified according to the operational differences in these two phases. For example, the learning algorithm of ANN's may be classified as supervised learning and unsupervised learning. If the connections between the PE's are updated to match the desired output for an input pattern, then the learning process is called supervised. If there is no desired output, but the user seeks a clustering of the inputs, then the learning process is called unsupervised. The rule, which updates the connection weights of the ANN, may

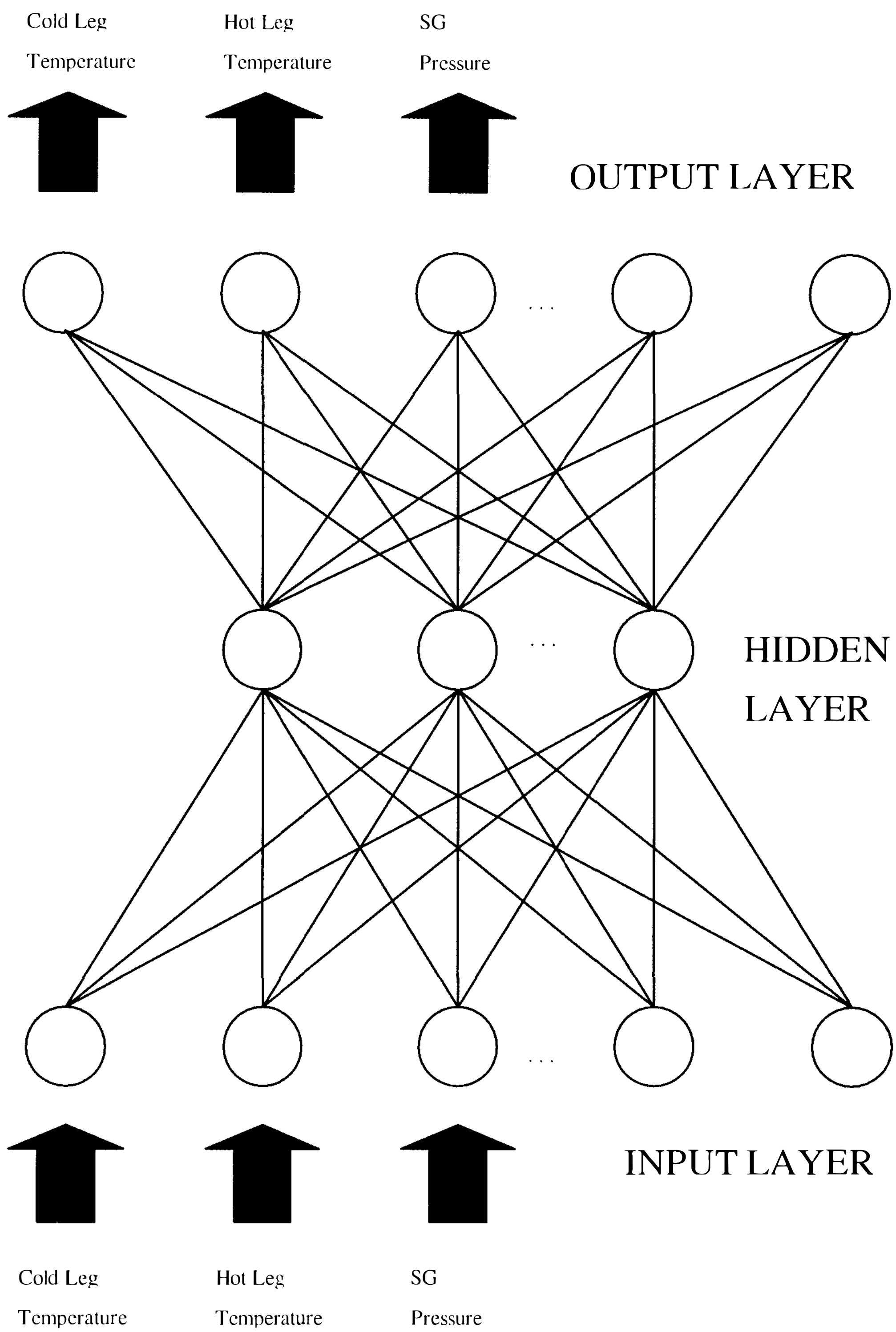

Figure 2.4: Auto-associative ANN for signal monitoring.

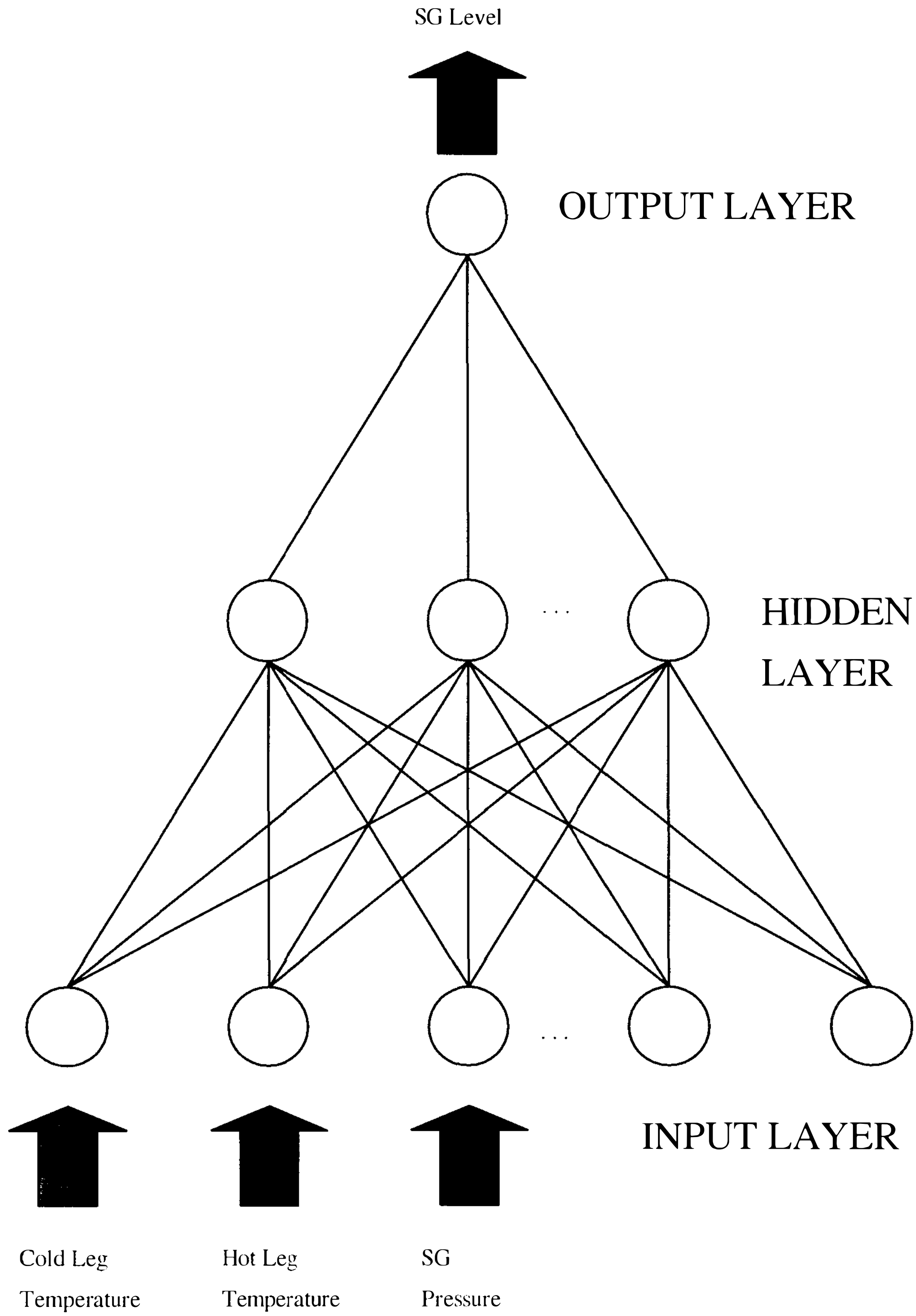

Figure 2.5: Hetero-associative back-propagation ANN for signal estimation.

also determine the type of the ANN (e.g. back-propagation algorithm).

During the recall process, the input pattern is propagated through the ANN in which the connections have been predetermined by the training. In this manner, a final output is determined for a given input. If there are no lateral of feedback connections between the PE's, the ANN is said to be feedforward, and the network recall is straightforward and takes one pass from the input layer to the output layer [24].

One of the algorithms to train the connection weights of an ANN is the back-propagation algorithm. It has been applied to problems from data encoding to signal processing, and from financial analysis to stock market prediction. The back-propagation algorithm is a generalization of the least-squares minimization of error algorithm. It uses gradient descent technique to minimize the cost function which is the mean-squared difference between the desired and the actual network outputs [25].

A typical back-propagation ANN has an input layer, an output layer and one or more hidden layers. For most of the applications one hidden layer is enough to map the input-output relationship [26]. Each layer is fully connected to the succeeding layer. During recall, information flows from the input layer to the output layer in a feedforward manner. During learning, information is also propagated back through the ANN and is used to update the connection weights. The network can either be hetero-associative or auto-associative.

A back-propagation PE transfers its inputs as follows:

$$x_j^s = f\left(\sum_i \left(w_{ji}^s x_i^{s-1}\right)\right)$$
$$x_j^s = f\left(I_j^s\right) \tag{2.24}$$

where

x_j^s = current output state of *j*th PE in layer *s*,

w_{ji}^s = weight on connection joining *i*th PE in layer (*s-1*) to jth PE in layer *s*,

I_j^s = weighted summation of inputs to jth PE in layer *s*, and

f(z) = nonlinear transfer function.

The hyperbolic tangent was used as the transfer function in this study and has the form:

$$f(z) = \frac{e^{\gamma z} - e^{-\gamma z}}{e^{\gamma z} + e^{-\gamma z}} \tag{2.25}$$

where γ is an adjustable parameter.

The reason for selecting the hyperbolic tangent instead of a sigmoid function is that, the output range of the sigmoid ranges from 0 to +1, whereas the hyperbolic tangent ranges from -1 to +1 giving a wider range to map nonlinear systems. A plot of *f(z)* is shown in Figure 2.6.

For a back-propagation ANN, the local error is given with

$$\varepsilon_j^s = f'\left(I_j^s\right)\sum_k \left(\varepsilon_k^{s+1} w_{kj}^{s+1}\right) \tag{2.26}$$

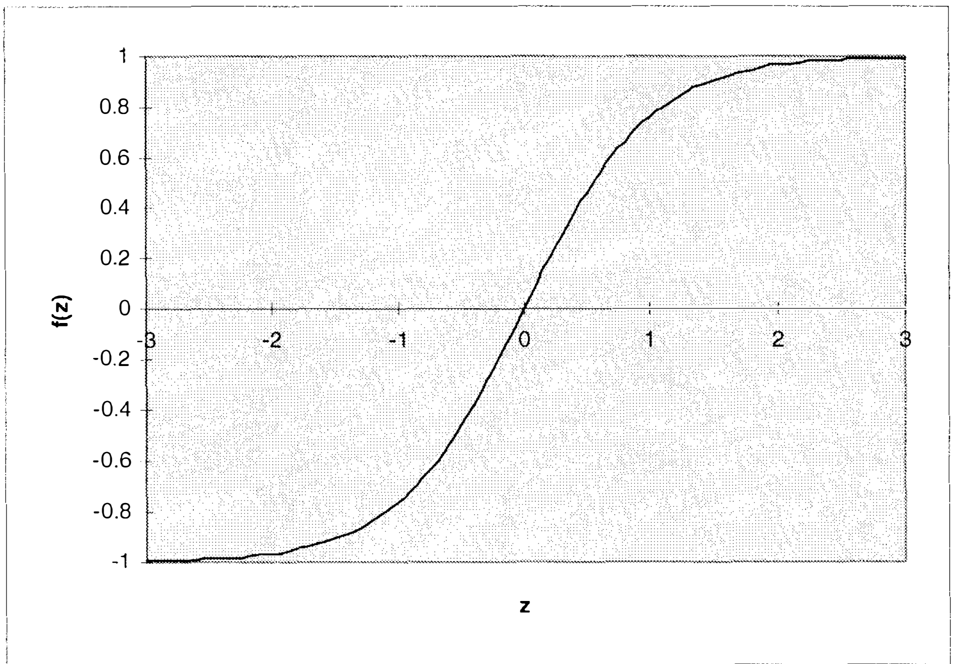

Figure 2.6: Plot of hyperbolic tangent given in Equation (2.25).

where $f'(z)$ is the partial derivation of $f(z)$ with respect to z.

The main learning mechanism in a back-propagation ANN is to propagate the input forward through the layers to the output layer, determine the error at the output layer using

$$\varepsilon_k^o = \left(d_k - o_k\right) f'\left(I_k\right) \tag{2.27}$$

and then propagate the error back through the ANN from the output layer to the input layer using Equation (2.26). Using the gradient descent rule, the weights are updated with a weight difference of

$$\Delta w_{ji}^s = \beta \varepsilon_j^s x_i^{s-1} \tag{2.28}$$

where β is the learning coefficient in the standard back-propagation training algorithm. The fast back-propagation algorithm usually includes a momentum term by adding a vector The fast back-propagation algorithm has usually includes a momentum term by adding a vector $\alpha \Delta w$ (which is a fraction α of the previous wight change) to the new weight change. The incremental weight is then defined as

$$\Delta w_{ji}^s = \beta \varepsilon_j^s x_i^{s-1} + \alpha \Delta w_{ji}^s \tag{2.29}$$

This prevents the network training from possible oscillations, helps avoid local minima and accelerates training [27].

As in the case of the process empirical modeling, artificial neural networks were also developed for dynamic estimation. Such systems may have the form shown in Equations

(2.18) - (2.20) and may be simplified to Equation (2.23). Then the inputs at previous time samples may be treated as separate inputs, fitting them into time-dependent ANN's. An example of a dynamic ANN is shown in Figure 2.7.

Both the PEM module and the ANN module in the PC-based signal validation system produce an estimate of the state variable with different sensor inputs. The estimates and the actual measurements are then used in the decision-making module to determine the status of the sensors.

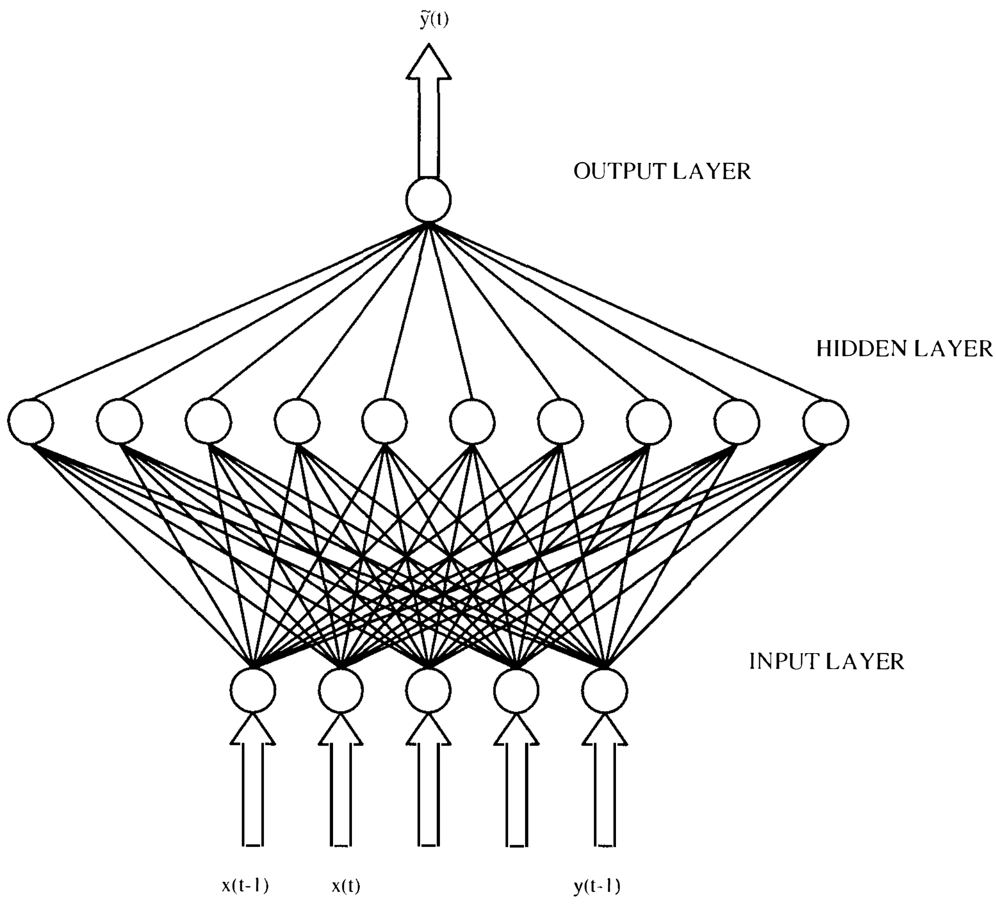

Figure 2.7: Back-propagation ANN for dynamic systems such as transient and semi-transient behaviors in Nuclear Power Plants.

Chapter 3

THE EXTENDED KALMAN FILTERING TECHNIQUE

3.1 The Linear Kalman Filter

The Kalman filtering technique (KFT) is an optimal state estimation algorithm for general stochastic systems. It requires the knowledge of a dynamic system model and is applicable to both stationary and nonstationary processes. The KFT has been studied and developed thoroughly in several areas since the original publication of R. E. Kalman's paper in 1960 [13]. The extensions of the original filter to nonlinear system estimation and to parameter estimation have been applied in many areas. Aeronautics, flight engineering and missile tracking systems use KFT for tracking and navigational control. This technology has been implemented in space missions. It is used in signal processing to solve system identification and deconvolution problems of linear systems. It has also found a large area of application in communication, control, as well as in seismic signal processing in geophysics [14, 28].

There are several rearrangements and extensions and numerous derivations of the KFT. However, the Kalman filter can be thought of as an optimal estimator that produces three types of outputs, given a noisy measurement sequence and the associated models (Figure 3.1). It can be thought of as a state estimator or a reconstructor, that is, it reconstructs estimates of the state $x(t)$ from noisy measurements $y(t)$. In this respect, it is

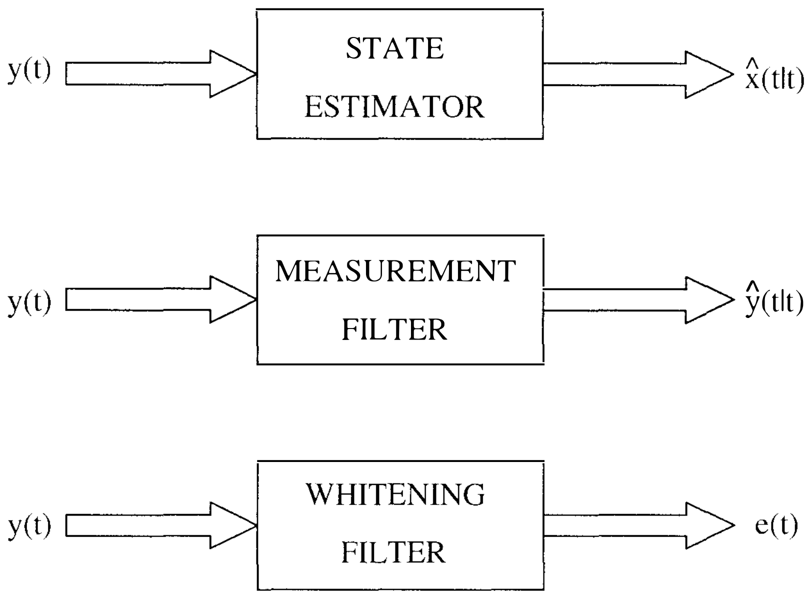

Figure 3.1: Various representations of the Kalman filter estimator.

almost an implicit solution of equations: since the state is not available directly, the models used can be considered as the means to implicitly extract *x(t)* from *y(t)*. Second, the Kalman estimator may be thought of as a measurement filter. It accepts a noisy measurement sequence *y(t)*, and produces a filtered measurement sequence $\hat{y}(t|t)$ as the output. Finally, the estimator serves as a whitening filter that accepts noisy correlated measurements *y(t)* and produces uncorrelated or white random process *e(t)*, called the innovation sequence. All these properties of the Kalman filter have been exploited in various applications including fault detection [29].

A process may be modeled by a set of stochastic linear vector difference equations in the state-space form as

$$x(t) = Ax(t-1) + Bw(t-1) \tag{3.1}$$

where x is the state vector with Gaussian noise sequence $\{w\}$ and noise covariance Q. The corresponding measurement model is given by

$$y(t) = Cx(t) + v(t) \tag{3.2}$$

where y is the measurement vector with Gaussian noise sequence $\{v\}$ and noise covariance R. Coefficient matrices A, B and C are determined using the parameters of the physical model. The equations that describe the state estimation are called the Kalman filter equations. Given the measurement sequence {y(t)} and the above defined model, the optimal filter minimizes the mean-squared error

$$E\left\{\left[x(t) - \hat{x}(t|t)\right]^T \left[x(t) - \hat{x}(t|t)\right]\right\} \tag{3.3}$$

The optimal filtered estimate $\hat{x}(t|t)$ is then computed recursively

$$\hat{x}(t|t) = \hat{x}(t|t-1) + G(t)e(t) \tag{3.4}$$

where

$\hat{x}(t|t-1)$ = one-step state prediction,

G(t) = Kalman gain, and

e(t) = innovation sequence, information gained from subsequent measurement.

The notation *(t|t-1)* denotes an estimation for time instant *t* with given measurements for time instant *t*-1.

The one-step predictor is given by

$$\hat{x}(t|t-1) = A\hat{x}(t-1|t-1) \tag{3.5}$$

The prediction error covariance matrix is updated as

$$P(t|t-1) = AP(t-1|t-1)A^T + BQB^T \tag{3.6}$$

The estimation error covariance matrix is

$$P(t|t) = [I - G(t)C]P(t|t-1) \tag{3.7}$$

where ***I*** is the identity matrix.

$$P(t|t-1) = E\left\{[x(t) - \hat{x}(t|t-1)][x(t) - \hat{x}(t|t-1)]^T\right\} \tag{3.8}$$

$$P(t|t) = E\left\{[x(t) - \hat{x}(t|t)][x(t) - \hat{x}(t|t)]^T\right\} \tag{3.9}$$

The innnovation sequence is given by

$$e(t) = y(t) - \hat{y}(t|t-1) = y(t) - C\hat{x}(t|t-1) \tag{3.10}$$

and the innovation covariance is

$$R_e(t) = CP(t|t-1)C' + R \tag{3.11}$$

Finally, the Kalman gain matrix is calculated as

$$G(t) = P(t|t-1)C' R_e^{-1}(t) \tag{3.12}$$

The recursive algorithm of the KFT is illustrated in Figure 3.2.

The recursive algorithm is initiated with $P(0|0) = P(0)$, which is the initial error covariance matrix of the initial state estimation $\hat{x}(0|0)$. The algorithm is executed for each measurement sample, and a filtered estimate is calculated.

3.2 Extension to State Estimation of Nonlinear Systems

The primary assumption made, while developing the Kalman filter equations was that the system to be modeled should be linear. However, most of the real-world modeling includes nonlinear equations, so that a modification to the standard Kalman filtering algorithm is needed. For example the U-tube steam generator model of a PWR is described by nonlinear equations and is used in this study for the application of KFT.

A common modification procedure is described [30]. First, the system is modeled using nonlinear difference equations in the state-space form as

$$x(t) = f\left(x(t-1)\right) + w(t-1) \tag{3.13}$$

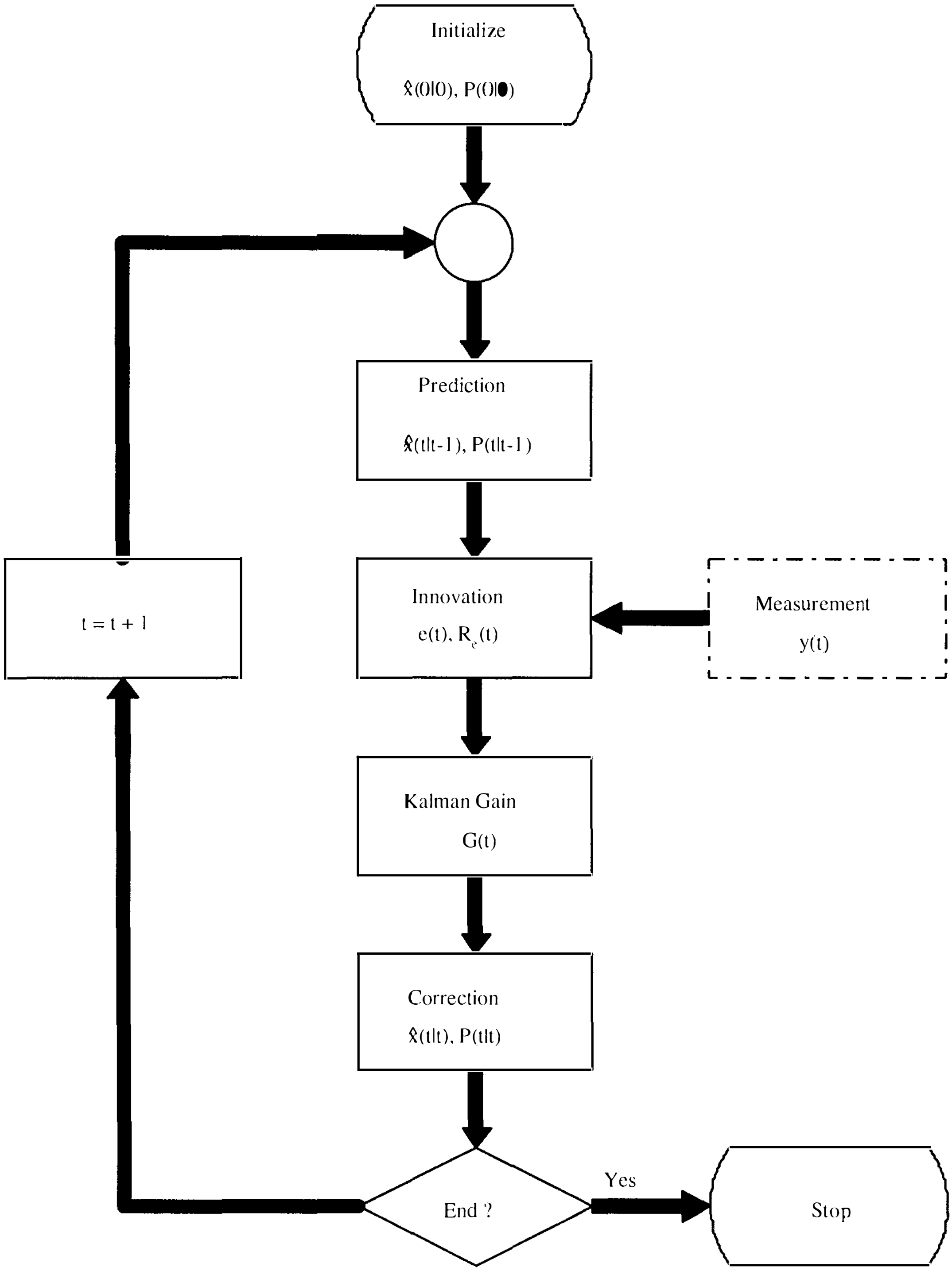

Figure 3.2: Kalman filter calculations.

and the corresponding measurement model

$$y(t) = h(x(t)) + v(t) \tag{3.14}$$

where

x(t) = state vector with Gaussian noise sequence $\{w\}$ and noise variance Q,

y(t) = measurement vector with Gaussian noise sequence $\{v\}$ and noise variance R, and

f(x(t)), *h(x(t))* = nonlinear functions of the state vector.

The optimal filter estimate is calculated using the following equations

$$\hat{x}(t|t) = \hat{x}(t|t-1) + G(t)e(t) \tag{3.15}$$

$$\hat{x}(t|t-1) = f\left(\hat{x}(t-1|t-1)\right) \tag{3.16}$$

The one-step prediction covariance matrix is

$$P\left(t|t-1\right) = F\left(\hat{x}\left(t-1|t-1\right)\right)P\left(t-1|t-1\right)F^{T}\left(\hat{x}\left(t-1|t-1\right)\right) + Q \tag{3.17}$$

The filter error covariance matrix is

$$P(t|t) = \left[I - G(t)H\left(\hat{x}(t|t-1)\right)\right]P(t|t-1) \tag{3.18}$$

The matrices F and H are defined as

$$F\left(\hat{x}\left(t-1|t-1\right)\right) = \left.\frac{\partial f(x)}{\partial x}\right|_{x=\hat{x}(t-1|t-1)} \tag{3.19}$$

$$H\left(\hat{x}\left(t|t-1\right)\right) = \left.\frac{\partial h(x)}{\partial x}\right|_{x=\hat{x}(t|t-1)} \tag{3.20}$$

The innovation sequence is given by

$$e(t) = y(t) - \hat{y}(t|t-1) = y(t) - h(\hat{x}(t|t-1)) \tag{3.21}$$

The innovation covariance matrix has the form

$$R_e(t) = H(\hat{x}(t|t-1))P(t|t-1)H^T(\hat{x}(t|t-1)) + R \tag{3.22}$$

Finally the Kalman filter gain matrix is calculated from

$$G(t) = P(t|t-1)H^T(\hat{x}(t|t-1))R_e^{-1}(t) \tag{3.23}$$

The algorithmic procedure for the extended Kalman filter in calculating the optimal estimates is similar to the one given in Figure 3.2, with the additional matrix calculations at each time step given in Equations (3.22) and (3.23).

The PC-Based Signal Validation System has a KFT module which is based on the extended Kalman filter, and uses a nonlinear system model.

3.3 Issues to be Considered in Implementing the Kalman Filter

The approach for developing Kalman filters has evolved from the solution of navigation and tracking problems. Designing a Kalman filter is a straightforward procedure as long as all the information about the process or system under investigation is available or can be gathered in a reasonable period of time. After deciding that a filter is necessary, the development proceeds through various phases of the Kalman filter design methodology [31]:

- Model development,

- Simulation, and
- Application.

The first phase consists of developing models for the process phenomenology, that is, a "process model" in the form of linear or nonlinear dynamic mathematical equations. Typically, this requires that the signal processor has the needed knowledge or that an expert in the area is available. Simultaneously, the measurement instrumentation is investigated in terms of bandwidth, response time, physical relations, etc., to develop a "measurement system" model. Finally, models of the inherent uncertainties must be developed. Here both random and systematic errors should be considered.

Once the models of the process, measurements, and noise are completed, then a simulator should be constructed to ensure that reasonable measurements are being produced. These phases of the KFT module were successfully developed by previous studies at The University of Tennessee for U-tube steam generators [20, 21, 22]. Discretized versions of the models are incorporated in the KFT module for state estimation. Sensor data from two operational nuclear power plants were used to verify the performance of the module.

It should be emphasized that several assumptions were made in deriving the Kalman filter equations. The basic recursive formula given in Equation (3.4) shows the importance of having a convenient way of recursively determining the innovations of the observed process. The state-space formulation is very helpful in this regard, but in many problems such models are not readily available. In such cases, much effort may often be spent by first trying to obtain good models for Kalman filter applications.

Most of the time the basic assumption of the model can not be described by a linear form as given in Equations (3.1) and (3.2). In fact, most of the system models in a typical PWR are nonlinear differential equations of the first or second order. Also, since having a state-space model in the form of Equations (3.13) and (3.14) is very important to develop an extended Kalman filter, differential equations must be converted to difference equations. Suppose that a system is modeled by nonlinear differential equations in the form

$$\frac{dx}{dt} = f\left(x(t)\right) \tag{3.24}$$

Using the forward-difference technique, Equation (3.24) can be approximated with a first order error as

$$\frac{x(t+1) - x(t)}{\Delta t} = f\left(x(t)\right) \tag{3.25}$$

or

$$\begin{aligned} x(t+1) &= f\left(x(t)\right)\Delta t + x(t) \\ x(t) &= f\left(x(t-1)\right)\Delta t + x(t-1) \end{aligned} \tag{3.26}$$

where Δt is the sampling time interval of measurements from process sensors.

The filter equations are in the form of a predictor-corrector algorithm. Any small uncertainties in the process model will be compensated by the corrector term.

Chapter 4

U-TUBE STEAM GENERATOR MODEL

4.1 Description of a Typical U-Tube Steam Generator

The most widely used type of steam generator in PWR systems is the recirculation type U-tube steam generator (UTSG). The general arrangement of a typical Westinghouse UTSG is given in Figure 4.1 [22].

The primary coolant enters the steam generator through an inlet nozzle at the left bottom of the inlet plenum. The coolant flows inside the U-tubes first upward and then downward, and thus transfers heat to the secondary fluid in the shell side of the steam generator. The primary fluid leaves the outlet plenum through an outlet nozzle connected to the cold leg piping [32].

Feedwater enters inside the downcomer shell at a level just above the U-tubes region. It flows down through an annulus inside the shell and mixes with water coming from the drum section. The water enters the tube bundle region where heat is transferred to the fluid. As it flows over the outside of the U-tubes, a mixture of steam and water is formed. The mixture enters the riser region where the nozzle effect increases the natural driving force. As the flow passes through the separator region, water is removed from the

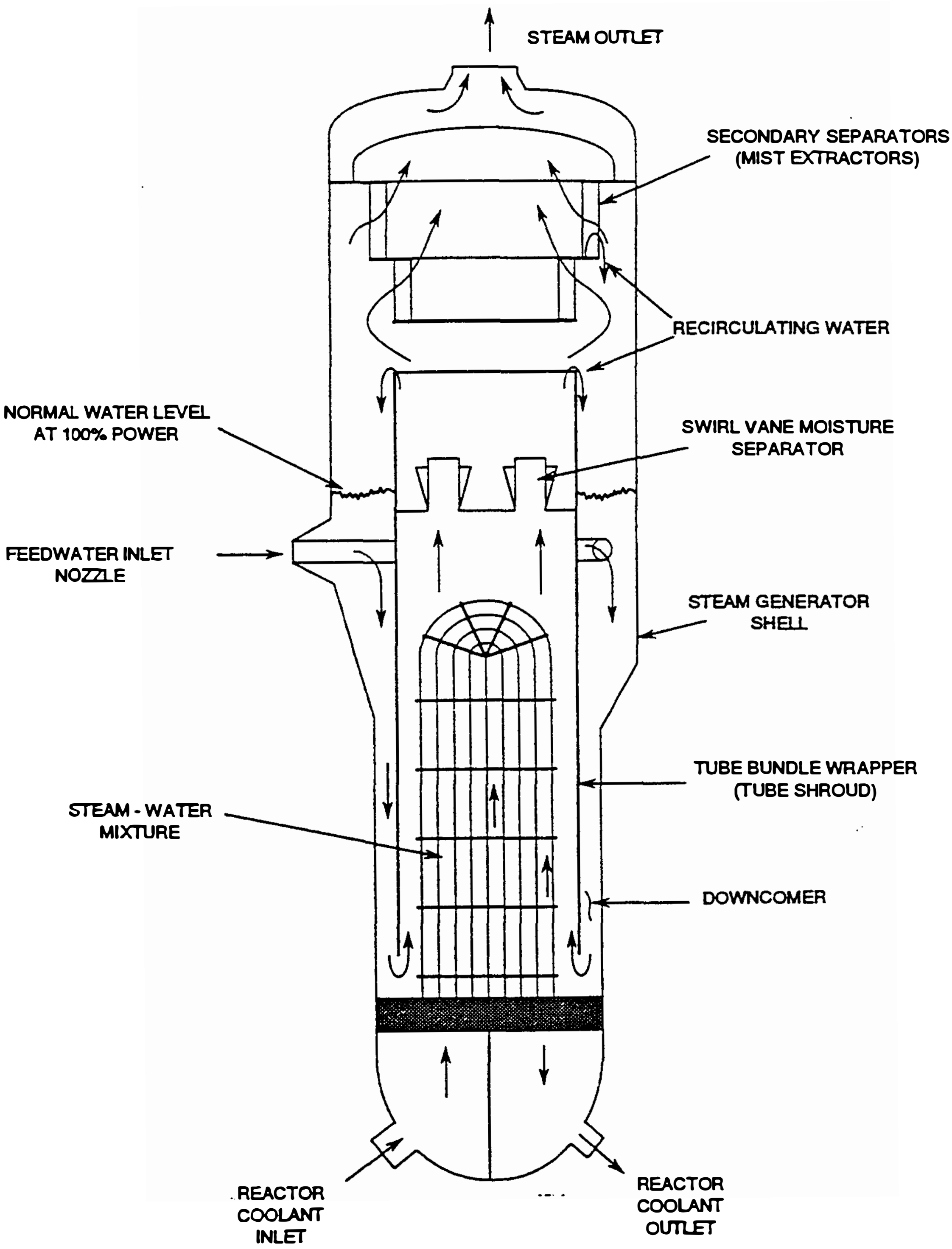

Figure 4.1: Schematic diagram of a typical Westinghouse U-tube steam generator [22]

steam and returned to the drum section. The steam leaving the separator passes through steam dryers and exits the steam generator with a quality of approximately 99.75%. The design parameters of a Westinghouse UTSG are listed in Table 4.1.

4.2 Steam Generator Model

Many theoretical models of the UTSG have been developed at The University of Tennessee. Ali's detailed nonlinear model was developed by Naghedolfeizi and extended by Eryürek for the Sequoyah Nuclear Plant (SNP) application [20, 21, 22]. The model can predict the dynamic behavior of thermal hydraulic processes in a UTSG system. The model is developed using the conservation of mass, energy and momentum principle with the following assumptions:

- Both water and steam are considered to be saturated.
- Density and specific heat capacity of feedwater, fluid in the subcooled region, and the primary side fluid are assumed to be constant.
- Heat transfer coefficients are constants.
- Steam leaving the UTSG is assumed to be 100% saturated.
- Heat transfer between the downcomer and tube bundle regions is negligible.

The thermodynamic properties of the saturated water and steam are assumed to be linear

Table 4.1: UTSG design parameters.

Parameter	**Value**
Number of U-tubes	3388
Tube outside diameter	0.875 inches
Tube metal thickness	0.05 inches
Height of U-tubes	35.54 ft
Total height of steam generator	67.67 ft
Effective flow area in tube region	60.87 ft^2
Effective flow area in downcomer region	32 ft^2
Effective flow area in riser region	48.7 ft^2
Effective flow area in drum region	110.74 ft^2
Riser Height	9.63 ft
Primary water mass flow rate	39.39 million lbm/hr
Volume of primary water in UTSG	1077 ft^3
Specific heat capacity of primary water	1.39 btu/lbm-°F
Inlet temperature of primary water	592.5 °F
Outlet temperature of primary water	542.5 °F
Average pressure in primary side	2250 psia
Average density of primary water	45.71 lbm/ft^3
Outlet steam flow rate	3.731 million lbm/hr
Steam pressure	849.7 psia

Table 4.1 Continued

Parameter	**Value**
Steam temperature at saturation pressure	521.9 °F
Inlet temperature of feedwater	434.3 °F
Average density of secondary subcooled water	52.32 lbm/ft^3
Effective heat transfer area	51500 ft^2
Film heat transfer coefficient of primary water in tubes	4500 btu/ft^2-hr-°F
Film heat transfer coefficient of secondary subcooled water	1972 btu/ft^2-hr-°F
Film heat transfer coefficient of secondary boiling water	6000 btu/ft^2-hr-°F
Metal tube conductivity	15 btu/lbm-°F

functions of the steam pressure for a range of ±100 psi from the normal operating point. The following equation defines the mathematical expression of this assumption.

$$F_P = X_m + K_n P \tag{4.1}$$

where

F_P = saturated steam or water property,

X_m = constant,

$K_n = \frac{\partial F_P}{\partial P}$, and

P = steam pressure.

The steam flow leaving the UTSG is considered to be a critical flow. The flow is defined in terms of steam generator pressure and steam valve coefficient as:

$$W_s = C_l P \tag{4.2}$$

where

W_s = steam flow rate,

C_l = steam valve coefficient, and

P = steam generator pressure.

A set of 19 state variables defines the nonlinear mathematical model of the UTSG. The forcing functions of the isolated UTSG model are:

- primary inlet temperature,
- steam valve coefficient,

- feedwater temperature.

The mathematical formulation of the UTSG is based on the model shown in Figure 4.2 [21]. The governing equations of the UTSG are given next and the description of the variables are given in Table 4.2.

Primary Side Equations

$$\frac{dT_{pi}}{dt} = \frac{W_{pi}}{M_{pi}}\left(\theta_i - T_{pi}\right) \tag{4.3}$$

$$\frac{dT_{p1}}{dt} = \frac{W_{pi}}{\rho_{pi}A_pL_{s1}}\left(T_{pi} - T_{p1}\right) + \frac{U_{pm}S_{pm1}}{M_{p1}C_{p1}}\left(T_{m1} - T_{p1}\right) \tag{4.4}$$

$$\frac{dT_{p2}}{dt} = \frac{W_{pi}}{\rho_{pi}A_pL_{s2}}\left(T_{p1} - T_{p2}\right) + \frac{U_{pm}S_{pm2}}{M_{p1}C_{p1}}\left(T_{m2} - T_{p2}\right) + \frac{\left(T_{p2} - T_{p1}\right)}{L_{s2}}\frac{dL_{s1}}{dt} \tag{4.5}$$

$$\frac{dT_{p3}}{dt} = \frac{W_{pi}}{\rho_{pi}A_pL_{s4}}\left(T_{p2} - T_{p3}\right) + \frac{U_{pm}S_{pm2}}{M_{p1}C_{p1}}\left(T_{m3} - T_{p3}\right) \tag{4.6}$$

$$\frac{dT_{p4}}{dt} = \frac{W_{pi}}{\rho_{pi}A_pL_{s1}}\left(T_{p1} - T_{p2}\right) + \frac{U_{pm}S_{pm2}}{M_{p1}C_{p1}}\left(T_{m4} - T_{p4}\right) + \frac{\left(T_{p3} - T_{p4}\right)}{L_{s1}}\frac{dL_{s1}}{dt} \tag{4.7}$$

$$\frac{dT_{po}}{dt} = \frac{W_{pi}}{M_{po}}\left(T_{p4} - T_{po}\right) \tag{4.8}$$

Metal Tube Equations

$$\frac{dT_{m1}}{dt} = \frac{U_{pm}S_{pm1}}{M_{m1}C_m}T_{p1} - \frac{U_{pm}S_{pm1} + U_{ms1}S_{ms1}}{M_{m1}C_m}T_{m1} + \frac{U_{ms1}S_{pm2}}{M_{m2}C_m}\frac{\left(T_d - T_{sat}\right)}{2} + \frac{\left(T_{m2} - T_{m1}\right)}{2L_{s1}}\frac{dL_{s1}}{dt} \tag{4.9}$$

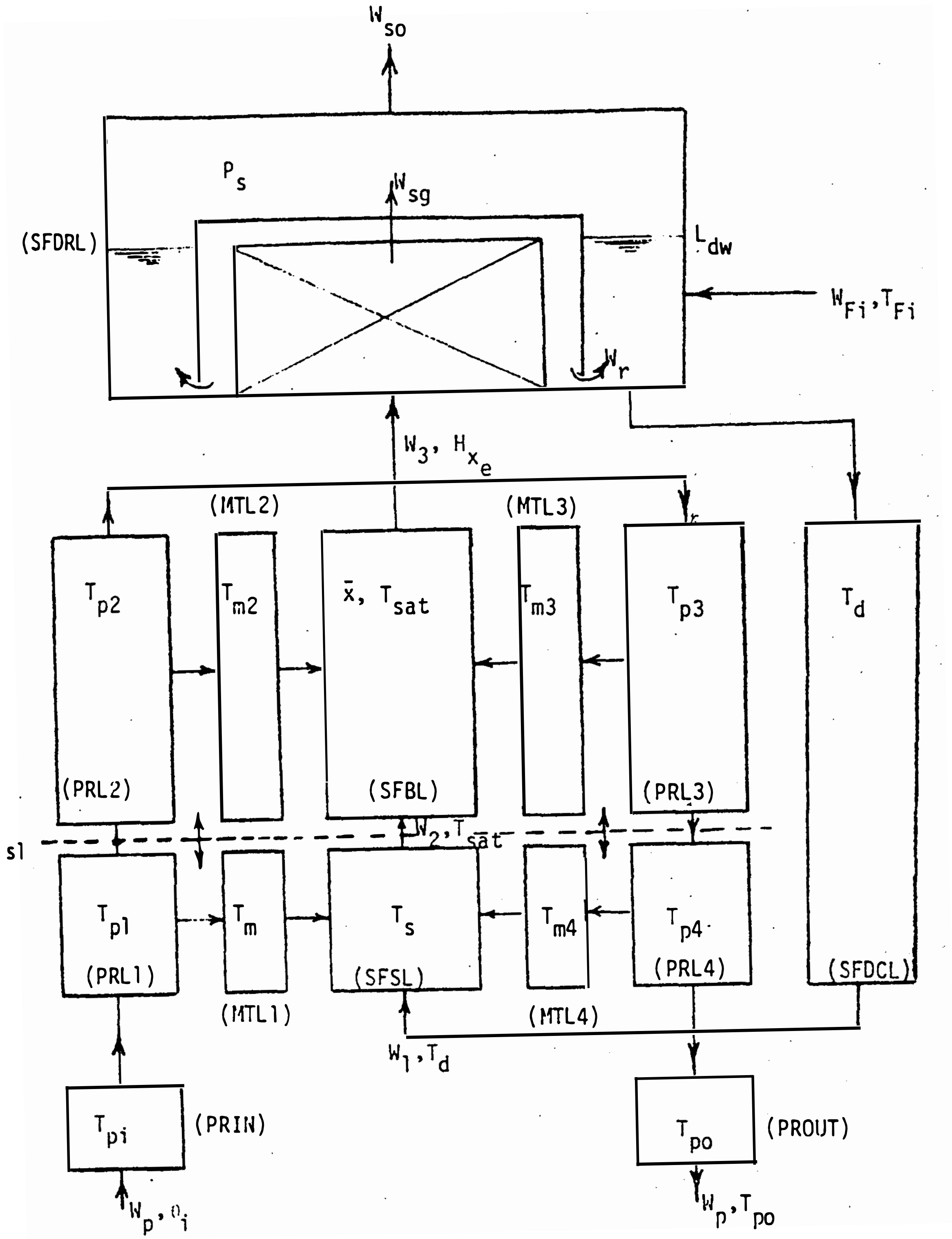

Figure 4.2: Schematic diagram of the UTSG model [21].

Table 4.2: UTSG model variables used in Equations (4.3) - (4.36).

Variable	**Definition**
A_{fs}	secondary flow area in the U-tube region
A_{dw}	effective area of the drum water region
C_1	effective pressure drop coefficient in the recirculation loop
C_l	steam valve coefficient
C_m	specific heat capacity of the metal tubes
$C_{p1,2}$	specific heat capacity of the primary fluid and subcooled region
h_b	average enthalpy of the boiling region
$h_{f,fg}$	saturated and latent enthalpies of water
h_{ex}	exit enthalpy of the boiling region
$K_{1\text{-}6}$	$\frac{\partial V_f}{\partial P}, \frac{\partial V_{fg}}{\partial P}, \frac{\partial h_f}{\partial P}, \frac{\partial h_{fg}}{\partial P}, \frac{\partial T_{sat}}{\partial P}, \frac{\partial \rho_g}{\partial P}$
L	effective height of U-tubes
L_d	downcomer length
L_{dw}	water level in the drum section of the steam generator
$L_{s1,2}$	subcooled and boiling lengths
$M_{m1,2}$	metal mass in metal nodes 1,2
$M_{p1\text{-}4}$	mass of water in the primary nodes 1-4
M_{pi}	mass of water in inlet plenum
P	steam generator pressure
$P_{r1,2}$	inside and outside perimeters of the U-tubes

Table 4.2 Continued

Variable	Definition
$S_{ms1,2}$	heat transfer areas from the U-tubes to the secondary side in the subcooled and boiling regions
$S_{pm1,2}$	heat transfer areas from the primary side to the U-tubes in nodes 1,2
T_d	downcomer temperature
T_{dw}	drum water temperature
$T_{m1\text{-}4}$	metal tube temperatures in nodes 1-4
$T_{p1\text{-}4}$	primary coolant temperatures in nodes 1-4
T_{pi}	coolant temperature in inlet plenum
T_{po}	coolant temperature in outlet plenum
T_{sat}	saturated temperature of the water and steam in UTSG
U_{pm}	heat transfer coefficient from the primary side to the metal side
$U_{ms1,2}$	heat transfer coefficient from the metal side to the subcooled and boiling regions
V_{dr}	volume of the drum section
$V_{f,g}$	specific volume of the saturated water and steam
V_{fg}	V_g - V_f
V_r	volume of riser region
W_{st}	steam flow rate
$X_{1\text{-}6}$	constant parameters
X_e	exit quality of the steam leaving the boiling region

Table 4.2 Continued

Variable	**Definition**
ρ_b	average density of the fluid in the boiling region
ρ_g	density of the saturated steam
ρ_r	density of the fluid in the riser region

$$\frac{dT_{m2}}{dt}=\frac{U_{pm}S_{pm2}}{M_{m2}C_m}T_{p2}-\frac{U_{pm}S_{pm2}+U_{ms2}S_{ms2}}{M_{m2}C_m}T_{m2}+\frac{U_{ms2}S_{pm2}}{M_{m2}C_m}T_{sat}+\frac{(T_{m2}-T_{m1})}{2L_{s2}}\frac{dL_{s1}}{dt} \tag{4.10}$$

$$\frac{dT_{m3}}{dt}=\frac{U_{pm}S_{pm2}}{M_{m2}C_m}T_{p3}-\frac{U_{pm}S_{pm2}+U_{ms2}S_{ms2}}{M_{m2}C_m}T_{m3}+\frac{U_{ms2}S_{pm2}}{M_{m2}C_m}T_{sat}+\frac{(T_{m3}-T_{m4})}{2L_{s2}}\frac{dL_{s1}}{dt} \tag{4.11}$$

$$\frac{dT_{m4}}{dt}=\frac{U_{pm}S_{pm1}}{M_{m1}C_m}T_{p4}-\frac{U_{pm}S_{pm1}+U_{ms1}S_{ms1}}{M_{m1}C_m}T_{m4}+\frac{U_{ms1}S_{pm2}}{M_{m2}C_m}\frac{(T_d-T_{sat})}{2}+\frac{(T_{m3}-T_{m4})}{2L_{s1}}\frac{dL_{s1}}{dt} \tag{4.12}$$

Secondary Side Equations

Subcooled Region Equations

$$\frac{dL_{s1}}{dt}=\frac{(W_1-W_2)}{\rho_{s1}A_{fs}} \tag{4.13}$$

$$\frac{d}{dt}\left(\rho_{s1}A_{fs}L_{s1}C_{p2}\frac{(T_d+T_{sat})}{2}\right)=U_{ms1}P_{r2}L_{s1}(T_{m1}+T_{m4}-T_d-T_{sat})+W_1C_{p2}T_d-W_2C_{p2}T_{sat} \tag{4.14}$$

Boiling Region Equations

$$\frac{d}{dt}\left(\rho_b A_{fs}L_{s2}\right)=W_2-W_3 \tag{4.15}$$

$$\frac{d\rho_v}{dt}=-\frac{\left(K_1+K_2\frac{X_e}{2}\right)}{\left(V_f+\frac{X_e}{2}V_{fg}\right)^2}\frac{dP}{dt}-\frac{V_{fg}}{2\left(V_f+\frac{X_e}{2}V_{fg}\right)^2}\frac{dX_e}{dt} \tag{4.16}$$

$$\frac{d}{dt}\left(\rho_b A_{fs} L_{s2}\right) = U_{ms2} P_{r2} L_{s2}\left(T_{m2} - T_{sat}\right) + U_{ms2} P_{r2} L_{s2}\left(T_{m3} - T_{sat}\right) + W_2 h_f - W_3 h_{ex}$$

(4.17)

Drum Region Equations

$$\frac{d}{dt}\left(V_r \rho_r\right) = W_3 - W_4 \tag{4.18}$$

$$\frac{d\rho_r}{dt} = -\frac{\left(K_1 + K_2 X_e\right)}{\left(V_f + X_e V_{fg}\right)^2} \frac{dP}{dt} - \frac{V_{fg}}{\left(V_f + X_e V_{fg}\right)^2} \frac{dX_e}{dt} \tag{4.19}$$

$$\frac{d}{dt}\left(\rho_{dw} A_{dw} L_{dw}\right) = W_{fi} + \left(1 - X_e\right) W_4 - W_1 \tag{4.20}$$

$$\frac{d}{dt}\left(\rho_{dw} A_{dw} L_{dw} T_{dw}\right) = W_{fi} T_{fi} + \left(1 - X_e\right) W_4 T_{sat} - W_1 T_{dw} \tag{4.21}$$

$$\left(V_{dr} - A_{dw} L_{dw}\right) \frac{d\rho_g}{dt} - \left(\rho_g A_{dw}\right) \frac{dL_{dw}}{dt} = X_e W_4 - C_l P \tag{4.22}$$

Downcomer Region Equation

$$\frac{dT_d}{dt} = \frac{W_1}{M_d}\left(T_{dw} - T_d\right) \tag{4.23}$$

Recirculation Loop Equation

$$W_1 = \frac{C_1}{12}\left(\rho_d\left(L_{dw} + L_d - L_{s1}\right) - L_r \rho_r\right)^{\frac{1}{2}} \tag{4.24}$$

Thermodynamic Properties of Water and Steam

$$h_b = h_f + \frac{X_e}{2} h_{fg} \tag{4.25}$$

$$h_{ex} = h_f + X_e h_{fg} \tag{4.26}$$

$$h_f = X_3 + K_3 P \tag{4.27}$$

$$h_{fg} = X_4 + K_4 P \tag{4.28}$$

$$L_{s2} = L - L_{s1} \tag{4.29}$$

$$T_{sat} = X_1 + K_5 P \tag{4.30}$$

$$V_f = X_1 + K_1 P \tag{4.31}$$

$$V_{fg} = X_2 + K_2 P \tag{4.32}$$

$$W_{st} = C_l P \tag{4.33}$$

$$\rho_b = \frac{1}{V_f + \frac{X_e}{2} V_{fg}} \tag{4.34}$$

$$\rho_r = \frac{1}{V_f + \frac{X_e}{2} V_{fg}} \tag{4.35}$$

$$\rho_g = X_6 + K_6 P \tag{4.36}$$

4.3 Steam Generator Control System

A three-element controller is considered as the UTSG control system in this study. The three-element controller is used to regulate the water level in the steam generator and utilizes three signals, namely, feedwater flow rate, steam flow rate and steam generator water level. It maintains the level at a desired set point, which is derived from the first-stage turbine impulse pressure, by controlling the feedwater flow rate to the system.

The block diagram representation of a three-element controller designed by the Westinghouse Corporation and used at the Sequoyah Nuclear Plant (SNP) is shown in Figure 4.3. It includes a filter, proportional and integral (PI) controllers, and feedwater valve dynamics. The actuating level signal is preprocessed using a low-pass filter before entering the first PI control element having a gain factor $G_1(s)$. This helps to diminish the effect of high frequency noise in the signal. The negative feedwater flow rate and positive steam flow rate signals are summed with the output signal of the first PI controller having a gain $G_1(s)$ and passed through the second PI control element having a gain $G_2(s)$. The resulting signal leaving the controller governs the feedwater valve positioner which has a second order system characteristic.

The mathematical formulations of the UTSG controller are based on the schematic shown in Figure 4.4. The governing equations of the controller are given next and the description of the variables are given in Table 4.3.

$$\frac{dV}{dt} = \frac{L_{dw} - L_{dw0} - V}{\tau} \tag{4.37}$$

$$\frac{dU}{dt} = \frac{G_1\left(L_{dw} - L_{dw0} - V\right)}{\tau} + \frac{V}{\tau_1} \tag{4.38}$$

$$\frac{dW}{dt} = G_2 V\left(\frac{1}{\tau_1} - \frac{G_1}{\tau}\right) + \frac{U}{\tau_2} + \frac{G_1 G_2\left(L_{dw} - L_{dw0}\right)}{\tau} \tag{4.39}$$

$$\frac{dZ}{dt} = C_l P - W_{fi} \tag{4.40}$$

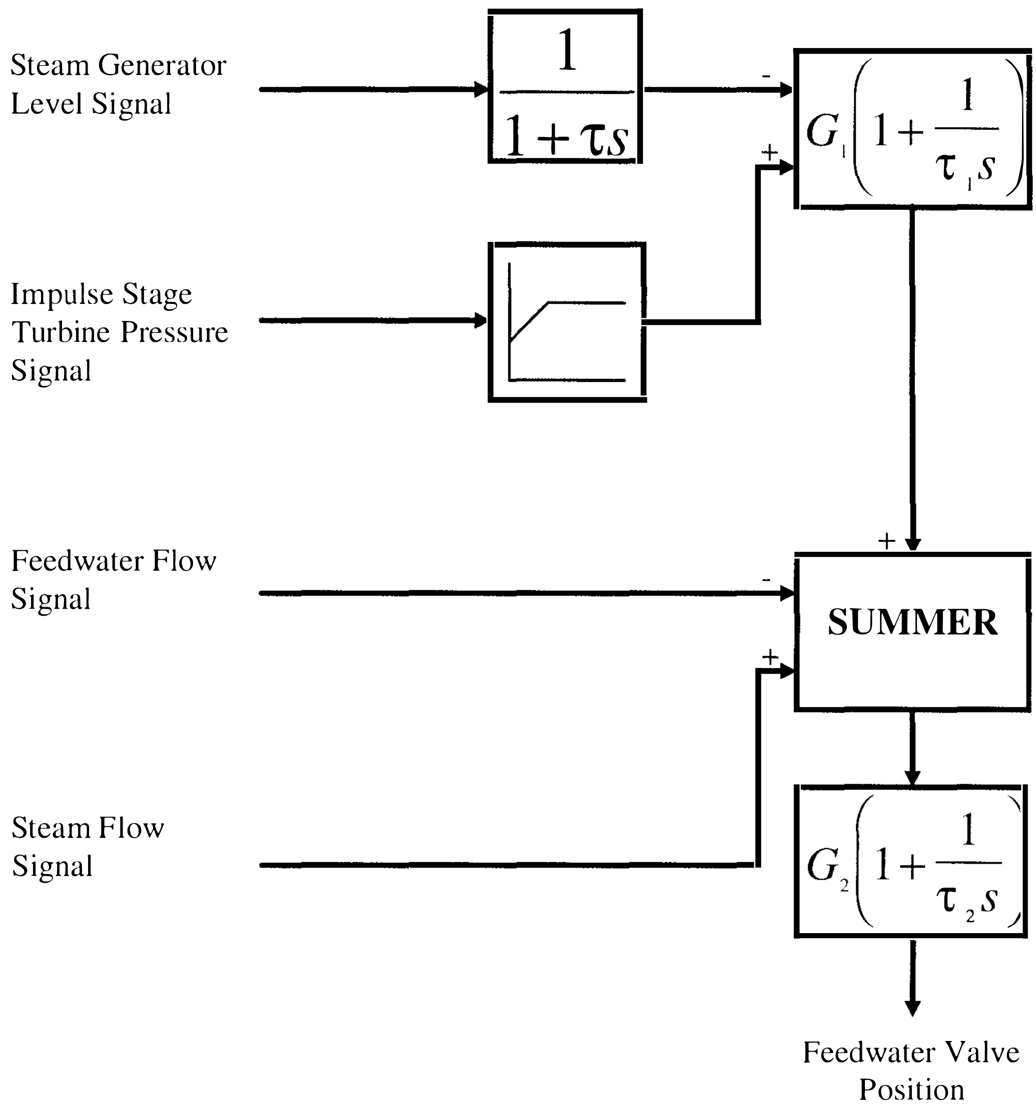

Figure 4.3: Block diagram representation of the three-element controller.

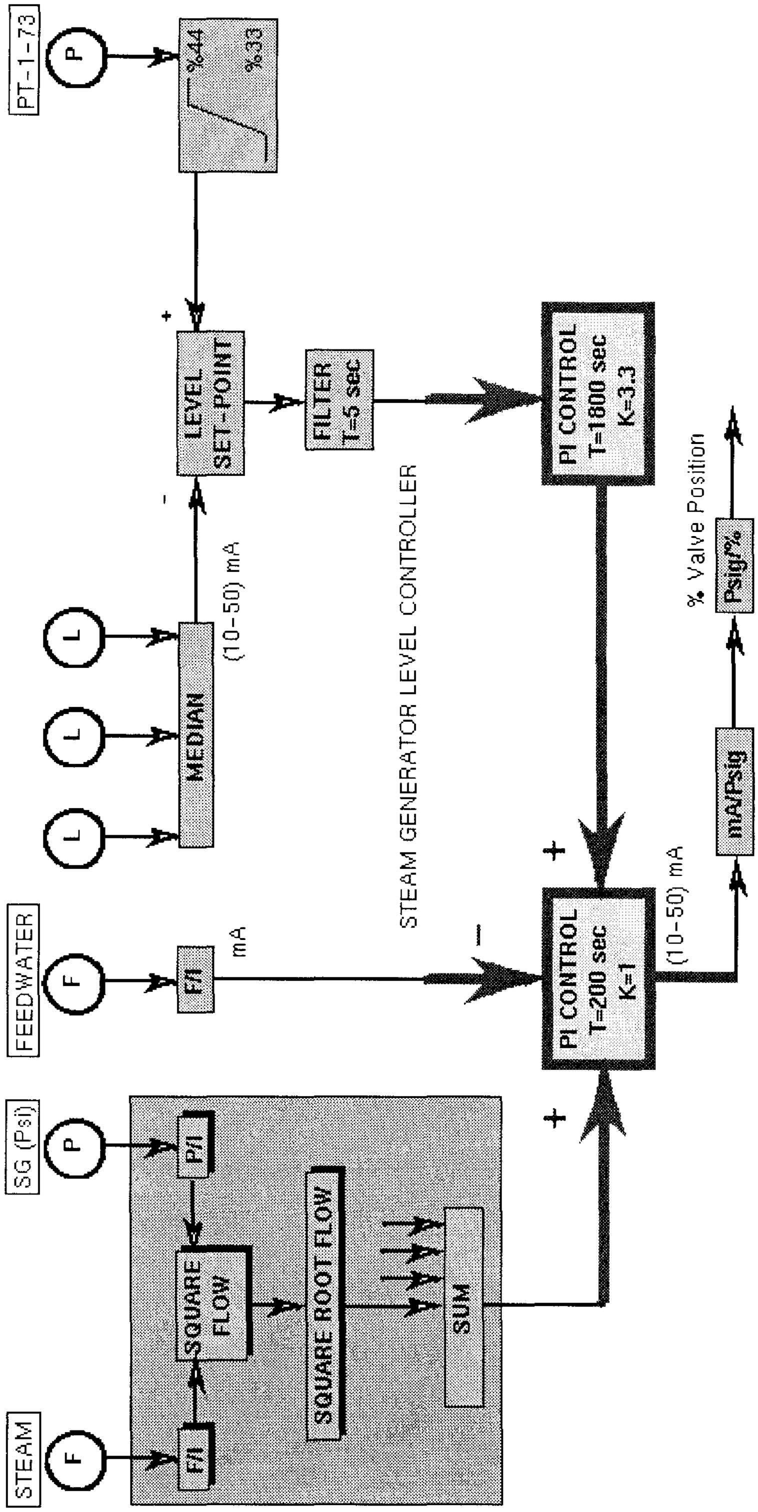

Figure 4.4: Design schematic of the UTSG controller used in this study.

Table 4.3: Three-element controller variables used in Equations (4.37) - (4.41).

Variable	Definition	Design Value
G_1	gain factor of the first PI controller	3.3
G_2	gain factor of the second PI controller	1
G_v	gain factor feedwater valve system	32.2
C_l	feedwater valve coefficient	
L_{dw}	water level in the UTSG (measured above the bundles)	
M	flow signal to the controller element	
P	steam generator pressure	
U	control signal leaving the first PI controller	
V	level signal leaving the filter element	
W	control signal leaving the second PI controller	
W_{fi}	feedwater flow rate	
Z	dummy variable	
τ	filter time constant	5 seconds
τ_1	time constant of the first PI controller	30 minutes
τ_2	time constant of the second PI controller	200 seconds
ζ	damping ratio of the feedwater valve system	3.18
ω_n	natural frequency of the feedwater valve system	0.63 radians/second

$$\frac{d^2W_{fi}}{dt}+2\zeta\omega_n\frac{dW_{fi}}{dt}+\omega_n^2\left(W_{fi}-W_{fi0}\right)-G_vG_2\omega_n^2\left(C_lP-W_{fi}\right)-G_v\omega_n^2\left(W+\frac{Z}{\tau_2}\right)=0 \qquad (4.41)$$

Equations (4.37) - (4.41) are used to model the UTSG controller and appended to the steam UTSG model and are set up in the difference equation form.

Chapter 5

SIGNAL VALIDATION SYSTEM INTEGRATION

5.1 Introduction

The PC-based signal validation system consists of four modules as described in Chapters 2 and 3. Each of the modules produces an estimate of a sensor output to be validated. In addition, the GCC module calculates inconsistency indices which are also utilized in decision-making. The difference between the measured sensor output and the module estimated state value is converted to a fuzzy set, which is presented to the decision-making algorithm of the system executive. The outcome of the decision-making process is compared with the information base, which consists of a library of prototype membership functions. A final decision about the signals to be validated is reached in linguistic forms as *very bad*, *bad*, *medium*, *good* or *very good*. An overall schematic of the PC-based signal validation system is shown in Figure 5.1.

The system executive also controls input-output (I/O) among various devices and the programs. One important task of the system executive is to receive live data from an operational nuclear power plant. This is accomplished by using a local area network (LAN) and gathering information from a data acquisition computer. Another important task of the system executive is to display processed and measured data to the user.

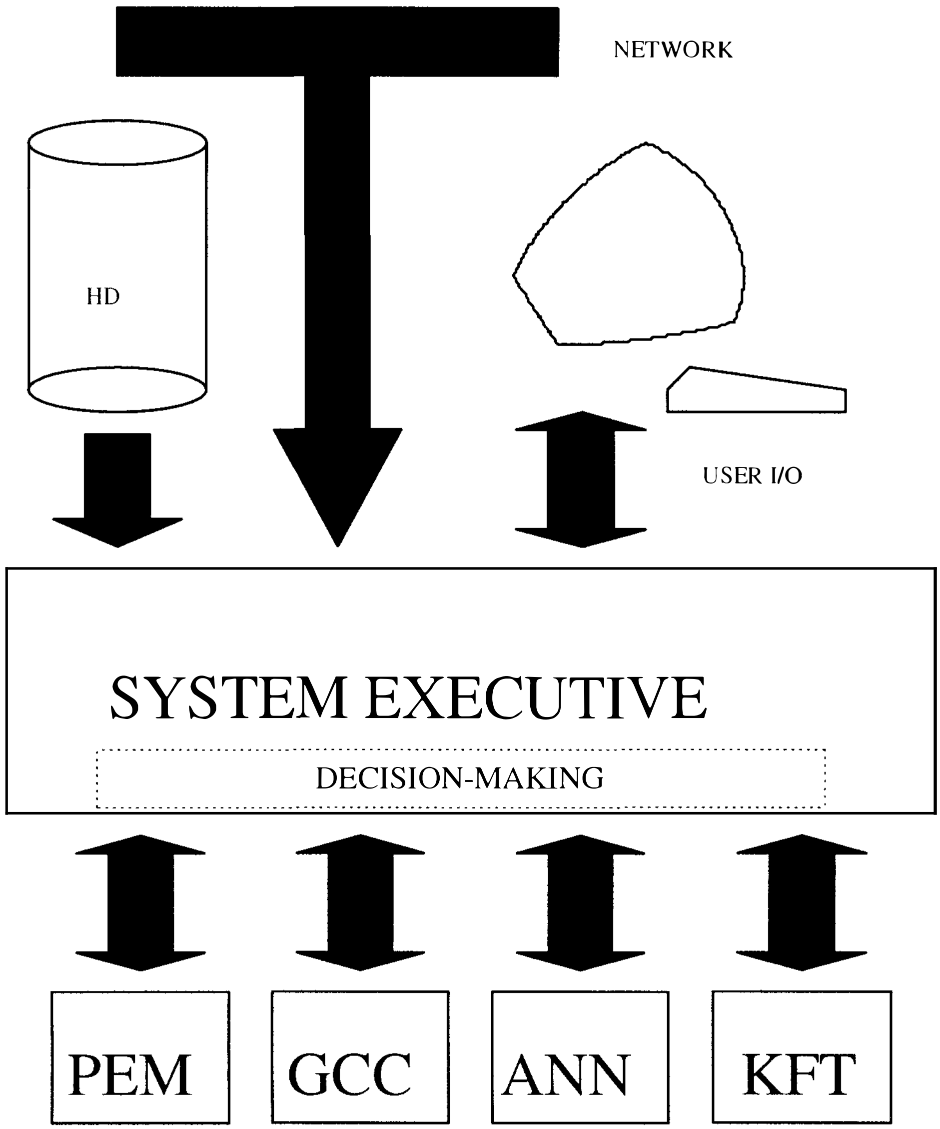

Figure 5.1: Integration of signal validation modules with the system executive.

This task must be accomplished in a user-friendly environment in which the user would be able to navigate through the information space easily. Hypertext links and Microsoft Windows 3.1™ standards enable the design of GUI objects, which the user can recognize by relating them with every-day objects [33]. Navigation through this information space is managed by point-and-click operations of the mouse interface of a standard PC. A large volume of information and linguistic readings are converted to graphical objects such as plots and icons.

5.2 System Executive Design

5.2.1 Overview of Fuzzy Logic Reasoning

A fuzzy logic decision-making approach was developed for combining the signal validation results from the four modules. Problems in decision-making and in other areas such as pattern recognition, control, structural engineering and validation involve numerous aspects of uncertainty [34]. Additional vagueness is introduced as models become more complex but not necessarily more meaningful.

As far as uncertain data are concerned, we have neither instruments nor reasoning at our disposal, as well defined and unquestionable as those used in probability theory. When measurements are bad or no longer possible and when we really have to make use of human reasoning, then the theories dealing with the treatment of uncertainty provide the required complement and fill in the gap left in the field of knowledge representation. Fuzzy sets and fuzzy logic theory, founded by Zadeh, provide a systematic framework for dealing with uncertain systems [35].

As an example, consider a measurement of a variable, such as the steam generator water level. Measurements may indicate that the steam generator level is at 60%, 70% or 80%, which is a crisp value. However, human reasoning interprets this reading as *low*, *normal* or *high*. Fuzzy logic uses such values in its computations, and the variable that takes linguistic values is called a "fuzzy variable". The values *low*, *normal* and *high* are called "fuzzy values".

In crisp logic, if 70% is defined as a normal measurement, 68% may be a low measurement. However, the difference between the two measurements is 2% which may be very well an error in the measurement, resulting in false decision. Fuzzy computation uses an extension of the set theory and assigns a membership function ($\mu_A(x)$) for each fuzzy value. If we consider the measurement variable steam generator water level, the value low may be assigned as a fuzzy set:

$$Low = \frac{1}{10\%} + \frac{1}{20\%} + \frac{1}{30\%} + \frac{0.6}{40\%} + \frac{0.3}{50\%} + \frac{0.1}{60\%} + \frac{0}{70\%} + \frac{0}{80\%} + \frac{0}{90\%} + \frac{0}{100\%} \quad (5.1)$$

The + signs in Equation (5.1) should not be interpreted as additions, but rather a union of set operation. Equation (5.1) indicates that a grade of membership is assigned for every measurement in the class *low*. The numerator denotes the membership function ($\mu_A(x)$) for every crisp steam generator water level measurement x, shown in the denominator. The nearer the value is to 1, the more it belongs to the fuzzy set *low*. A graphical representation is given in Figure 5.2.

Several composition methods of fuzzy relations exist to build an inference engine [36, 37]. The extension principle is used to provide a general method for extending nonfuzzy mathematical concepts, such as logical operations, to deal with fuzzy quantities. The max-min composition is one of the applications of the extension principle which defines a new fuzzy set as a result of fuzzy operations on two fuzzy sets (R_1 and R_2) :

$$R_1 \circ R_2 = \left\{ \left((x, y), \mu_{R_1 \circ R_2}(x, y) \right) \middle| (x, y) \in X \times Y \right\} \tag{5.2}$$

where $\circ$ denotes any mathematical operation and $\times$ denotes the Cartesian product of two sets. The corresponding membership functions are determined as:

$$\mu_{R_1 \circ R_2}(x, y) = \max_{x \circ y} \left(\min\left(\mu_{R_1}(x), \mu_{R_2}(y) \right) \right) \tag{5.3}$$

In these two equations, it should be noted, that the mathematical operation is not performed on the membership functions, but rather on the elements of the set.

As an example of how the max-min composition works, let $X = [1,100]$ be the universe of discourse where the fuzzy sets

$$A = \frac{0.3}{1} + \frac{0.4}{3} + \frac{0.6}{5} + \frac{0.8}{6} + \frac{1}{7} + \frac{0.7}{8} + \frac{0.5}{9} \tag{5.4}$$

and

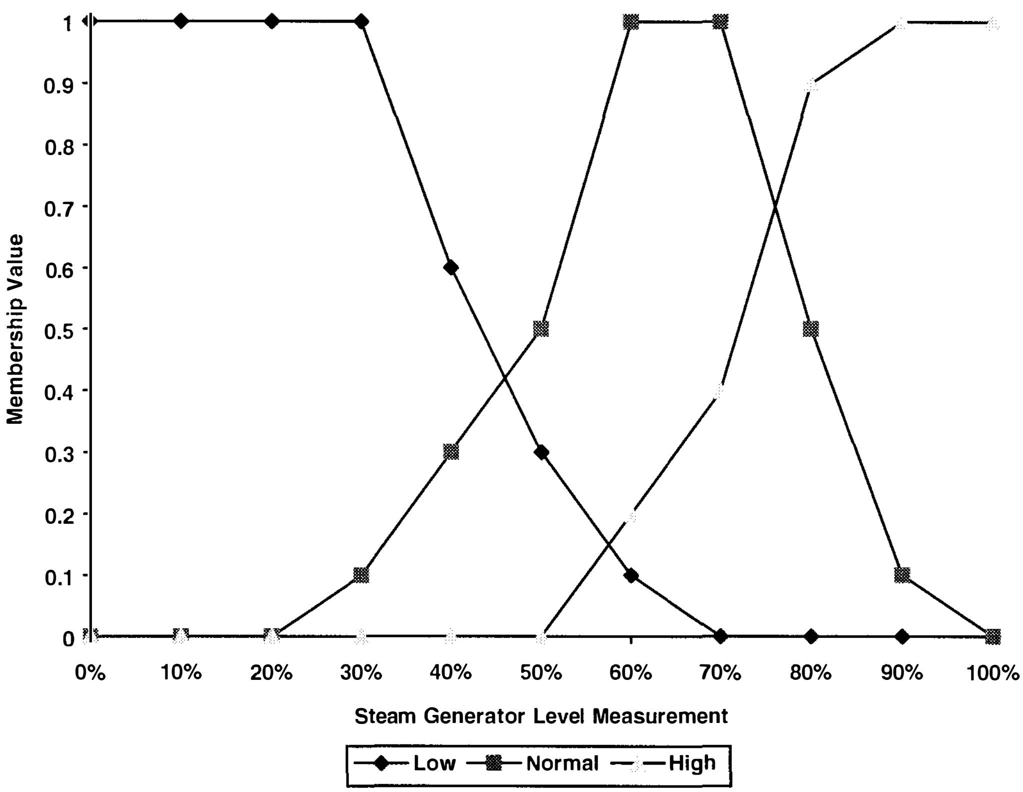

Figure 5.2: Representation of fuzzy variable *steam generator level* with three fuzzy values: *low*, *normal* and *high*.

$$B = \frac{0.4}{2} + \frac{0.6}{4} + \frac{0.8}{5} + \frac{1}{6} + \frac{0.8}{8} + \frac{0.6}{9} + \frac{0.3}{10} \tag{5.5}$$

are defined. To compute the fuzzy-arithmetic-product of these two sets using the max-min composition as

$$\mu_{R_1 \cdot R_2}(x, y) = \max_{x \cdot y}\left(\min\left(\mu_{R_1}(x), \mu_{R_2}(y)\right)\right) \tag{5.6}$$

where "." denotes the arithmetic product of two crisp values, we first produce the Cartesian products of these two sets, and assign the minimum of the two membership functions of set A and B as a membership function. A set for

$$\begin{aligned} A \times B = {} & \frac{0.3}{(1,2)} + \frac{0.3}{(1,4)} + \frac{0.3}{(1,5)} + \frac{0.3}{(1,6)} + \frac{0.3}{(1,8)} + \frac{0.3}{(1,9)} + \frac{0.3}{(1,10)} + \\ & \frac{0.4}{(3,2)} + \frac{0.4}{(3,4)} + \frac{0.4}{(3,5)} + \frac{0.4}{(3,6)} + \frac{0.4}{(3,8)} + \frac{0.4}{(3,9)} + \frac{0.3}{(3,10)} + \\ & \frac{0.4}{(5,2)} + \frac{0.6}{(5,4)} + \frac{0.6}{(5,5)} + \frac{0.6}{(5,6)} + \frac{0.6}{(5,8)} + \frac{0.6}{(5,9)} + \frac{0.3}{(5,10)} + \\ & \frac{0.4}{(6,2)} + \frac{0.6}{(6,4)} + \frac{0.8}{(6,5)} + \frac{0.8}{(6,6)} + \frac{0.8}{(6,8)} + \frac{0.6}{(6,9)} + \frac{0.3}{(6,10)} + \\ & \frac{0.4}{(7,2)} + \frac{0.6}{(7,4)} + \frac{0.8}{(7,5)} + \frac{1}{(7,6)} + \frac{0.8}{(7,8)} + \frac{0.6}{(7,9)} + \frac{0.3}{(7,10)} + \\ & \frac{0.4}{(8,2)} + \frac{0.6}{(8,4)} + \frac{0.7}{(8,5)} + \frac{0.7}{(8,6)} + \frac{0.7}{(8,8)} + \frac{0.6}{(8,9)} + \frac{0.3}{(8,10)} + \\ & \frac{0.4}{(9,2)} + \frac{0.5}{(9,4)} + \frac{0.5}{(9,5)} + \frac{0.5}{(9,6)} + \frac{0.5}{(9,8)} + \frac{0.5}{(9,9)} + \frac{0.3}{(9,10)} \end{aligned} \tag{5.7}$$

is produced. The members in the denominator are multiplied, since the arithmetic product of sets A and B is wanted. In the resulting set, if same two members appear, the one that has the largest membership function is left. Finally, the fuzzy-arithmetic-product of A and B is formed as

$$A \cdot B = \frac{0.3}{2} + \frac{0.3}{4} + \frac{0.3}{5} + \frac{0.4}{6} + \frac{0.3}{8} + \frac{0.3}{9} + \frac{0.4}{10} + \frac{0.4}{12} + \frac{0.4}{14} + \frac{0.4}{15} + \frac{0.4}{16} + \frac{0.4}{18} + \frac{0.6}{20} + \frac{0.6}{24} + \frac{0.6}{25} + \frac{0.4}{27} + \frac{0.6}{28} + \frac{0.6}{30} + \frac{0.6}{32} + \frac{0.8}{35} + \frac{0.8}{36} + \frac{0.7}{40} + \frac{1}{42} + \frac{0.6}{45} + \frac{0.8}{48} + \frac{0.3}{50} + \frac{0.6}{54} + \frac{0.8}{56} + \frac{0.3}{60} + \frac{0.6}{63} + \frac{0.7}{64} + \frac{0.3}{70} + \frac{0.6}{72} + \frac{0.3}{80} + \frac{0.5}{81} + \frac{0.3}{90} \quad (5.8)$$

As Equations (5.6) - (5.8) show, $A.B$ is computed by changing both the membership functions and the members.

5.2.2 Fault-Tree Methodology Using Fuzzy Logic

Fault-tree methodologies graphically illustrate the failure logic associated with the development of a particular system failure (top event) from basic subcomponent failures (primary events). The term “event” denotes a dynamic change of state that occurs to system elements, which may include hardware, software, human, or environmental factors. A fault-tree represents a detailed and deductive analysis that requires extensive system information. The knowledge incorporated in a fault-tree can be articulated in logical rules of the form “IF A is *true* THEN B is *true*.” “However, it is well known that this type of syllogism fails to give an answer when the satisfaction of the antecedent clause is only partial.” Zadeh suggested a new type of fuzzy conditional inference, referred to as generalized modus ponens, and reads as follows [35]:

Premise: A is *partially true*

Implication: IF A is *true* THEN B is *true*

Conclusion: B is *partially true*

In generalized modus ponens, the antecedent is true only to some degree; hence, it is desired to compute the grade to which the consequent is satisfied. Fuzzy sets provide a natural environment for this type of computation because fuzzy variables (e.g. B) can take fuzzy values (e.g. *partially true*).

In the methodology used in this study, the primary events in the fault-tree are considered as fuzzy sets, and the term *true* is employed as a topic-neutral logical "true." For example, consider that one of the primary events in a fault-tree will occur if the pressure exceeds 2300 psia. Boolean logic requires that all pressure measurements higher than 2300 psia should satisfy the prerequisite set to the same degree. Contrary to that philosophy, the linguistic variable *Truth* is introduced, which will assign different degrees of membership to the fuzzy set *Pressure Higher Than 2300 psia* for different pressure measurements. In other words, a measurement of 2400 psia will be associated with a level of presumption of 1, where a crisp value of 2301 psia might receive a membership value 0.1. Therefore, every crisp measurement of the variable A is associated with a point $v(A)$ in the interval of the linguistic variable *Truth*, $V = [0,1]$, representing the truth value of the proposition "u is A" [17].

The decision-making algorithm consists of three steps:

1. Construction of fuzzy sets from errors between measurements and module estimations (for the GCC module from inconsistency indices).
2. Propagation of fuzzy sets through the fault-tree (fuzzy OR gate).
3. Comparision of the resultant fuzzy set with prototype fuzzy sets (*very bad*, *bad*, *medium*, *good* or *very good*) using dissemble index calculations.

In the first step, the ANN, PEM and KFT modules of the PC-based signal validation system produce an estimate. The absolute difference between the estimated and measured value is used to construct a fuzzy set in the truthness domain. For the GCC module, the inconsistency index is used to construct this fuzzy set. A graphical representation of converting from crisp error to fuzzy truthness is given in Figure 5.3a. Suppose the difference between the measured and estimated steam generator pressure is 30 psi, from Figure 5.3a, this yields with 70% belief, the sensor is faulty (If the error is more than 40 psi, it is definite that the sensor is faulty). The truth 0.7 is then taken as basis of the maximum of the membership function and a triangular membership function in the *Truth* domain *[0,1]* is constructed as shown in Figure 5.3b. Here *Truth* is an indication of the truthness of the sensor being faulty. The relationship between the confidence and error changes for each state variable and for each module. If the signal validation module produces estimates closer to the measurements, then the relationship between the error and confidence will be on a much tighter scale (e.g. 30 psi will mark a 100% confidence, rather than 30 psi marking 70% confidence of the sensor being faulty).

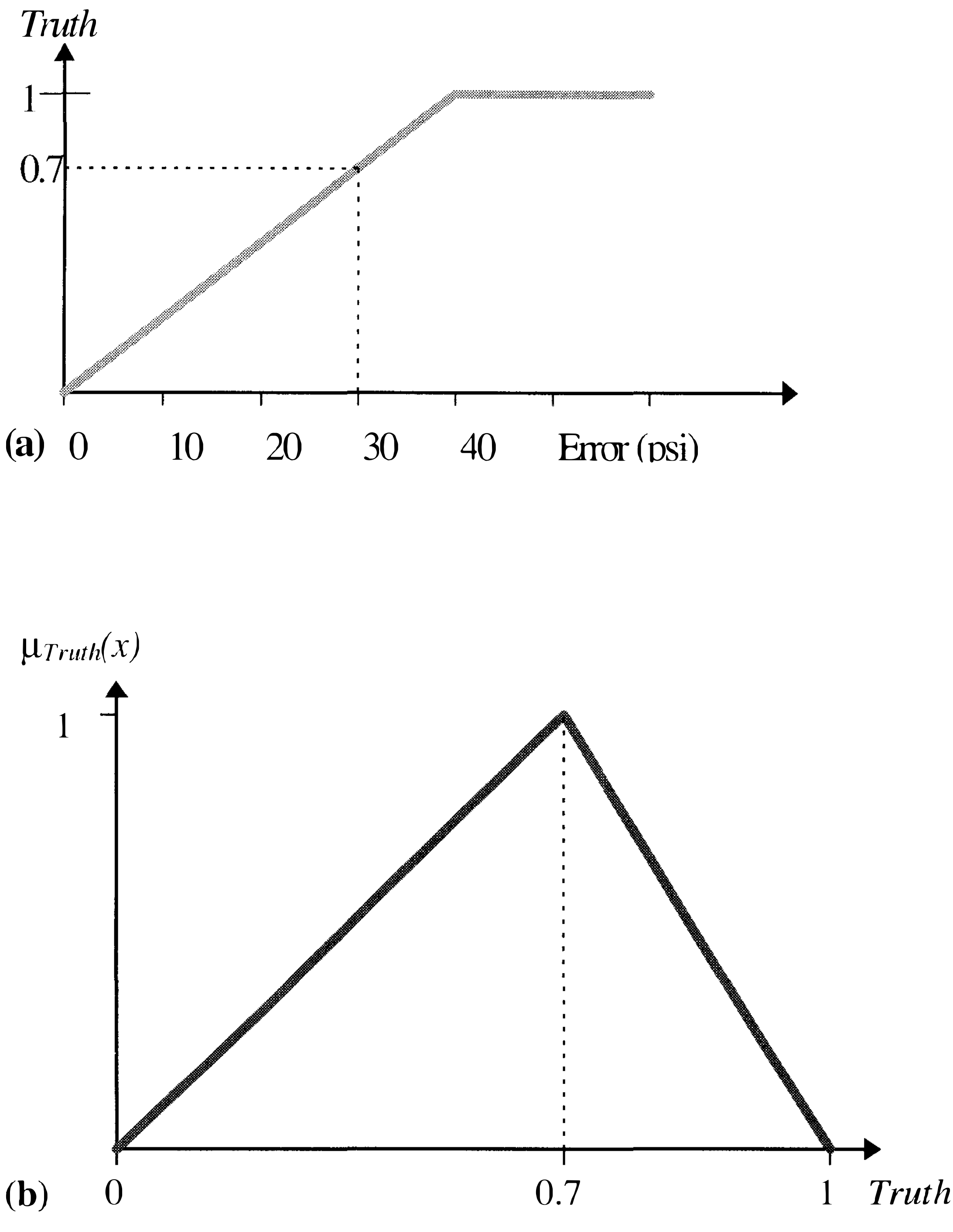

Figure 5.3: Construction of fuzzy sets from crisp errors between measurements and estimates (Valid for ANN, PEM and KFT modules).

The scale will also differ for different state variables (e.g. flow, level, etc.).

In the second step, every primary event (in this study the error between the measured and estimated state) of the fault-tree is considered as fuzzy and a membership function, $\mu_{PE}(x) \rightarrow [0,1]$, describing the degree of membership to a particular set, is constructed. The AND, OR, and NOT gates composing the fault-tree are treated linguistically, and their dyadic operation on the fuzzy sets constituting the primary events is computed through the extension principle. For example, the OR gate is modeled using the extension principle as:

$$\mu_{R_1 \cup R_2}(x, y) = \max_{x \cup y}\left(\min\left(\mu_{R_1}(x), \mu_{R_2}(y)\right)\right) \tag{5.9}$$

where $\cup$ denotes the maximum of two crisp values.

The fault-tree, used in decision-making for the PC-based signal validation system, is shown in Figure 5.4. In the final step of decision-making, the outcome of the logical operations is a new fuzzy set defined in the universe of discourse *[0,1]*. The top event is also considered as a fuzzy variable that takes five fuzzy values, namely, *safe*, *no fault*, *fault warning*, *fault*, *severe fault*. This value also can be interpreted as a sensor quality index such as *very good*, *good*, *medium*, *bad* and *very bad*. The five fuzzy values are algebraically depicted in the universe of discourse *[0,1]* with five membership functions that compose a library of prototypes. Generally, the result of the logical operations on the membership functions defining the primary events will be somewhat different from the

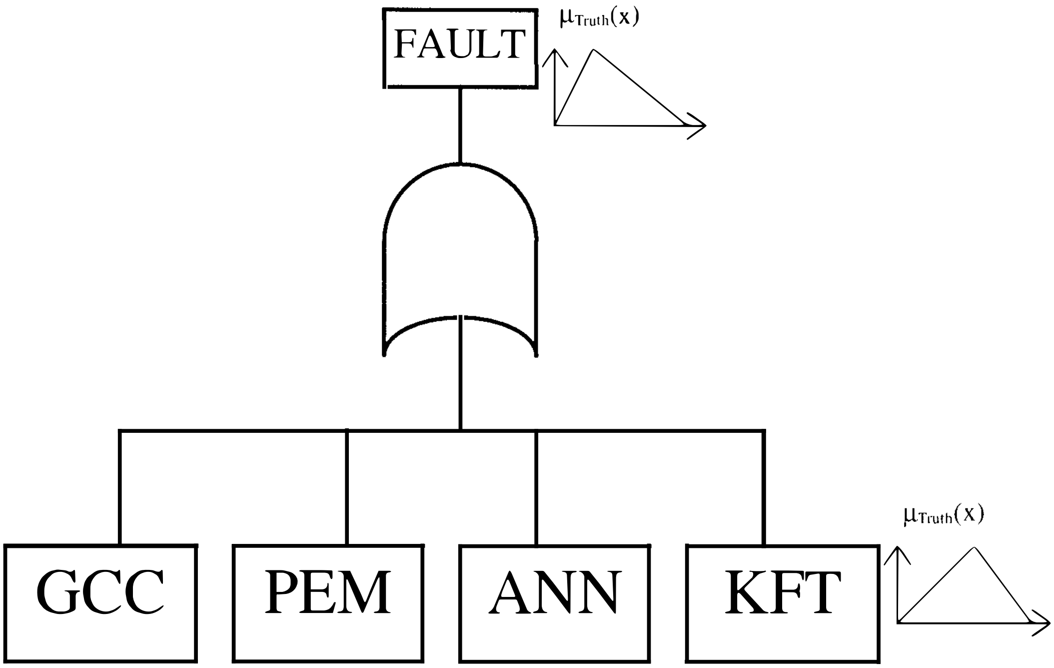

Figure 5.4: Fault-tree leading to sensor fault.

prototype membership functions defining the linguistic values of the top event. In order to draw a conclusion concerning the type of top event, the distance between the computed membership function and the prototype membership functions is calculated. This distance is also referred to as the dissemblence index and the minimum dissemblence index value is used to define the output of the fault-tree as *safe*, *no fault*, *fault warning*, *fault* or *severe fault*.

5.3 Graphical User Interface

The PC-based signal validation system is developed in Microsoft Windows 3.1™ environment with Microsoft Visual Basic 3.0. The Visual Basic programming system allows programmers to create attractive and useful applications that fully exploit the graphical user interface (GUI). Visual Basic makes the programmer more productive by providing appropriate aspects of the GUI development. The programmer creates GUI's for applications by drawing objects in a graphical manner. Then the properties on these objects are set to refine their appearance and behavior. This interface reacts to the user by responding to events that occur in the interface.

Using Visual Basic, the programmer can create powerful, full-featured applications that exploit the key features of Microsoft Windows, including multiple-document interface (MDI), object linking and embedding (OLE), dynamic data exchange (DDE), graphics and more. Visual Basic can be extended by adding custom controls and by calling procedures in the dynamic-link libraries (DLL's). The finished application is a true .EXE file that uses run-time DLL's which can be distributed freely [38].

The computational modules (GCC, PEM, ANN and KFT) were developed in standard FORTRAN language. They were compiled into DLL's and combined with the GUI provided by Visual Basic. A final .EXE file was produced which could be transported to any PC.

The GUI of the PC-based signal validation system provides hypertext buttons to the users. These enable the user to navigate in the information space with basic mouse operations, such as point and click. Virtual reality techniques are also combined in these buttons, such that they give the user a three-dimensional sense, recognizing itself as a real-world button. This simplifies the use of key sequences to accomplish a certain action. The procedure to be followed can be accomplished by navigating through the hypertext command buttons provided in the GUI of the PC-based signal validation system. An example of such a GUI, used in this study is shown in Figure 5.5.

The GUI of the PC-based signal validation system has different ways of displaying information. Instant measurements are displayed in digital and analog forms. The analog displays, as shown in Figure 5.5, are simulated using graphical objects. However, digital presentations of the measurements are always important for plant engineering systems.

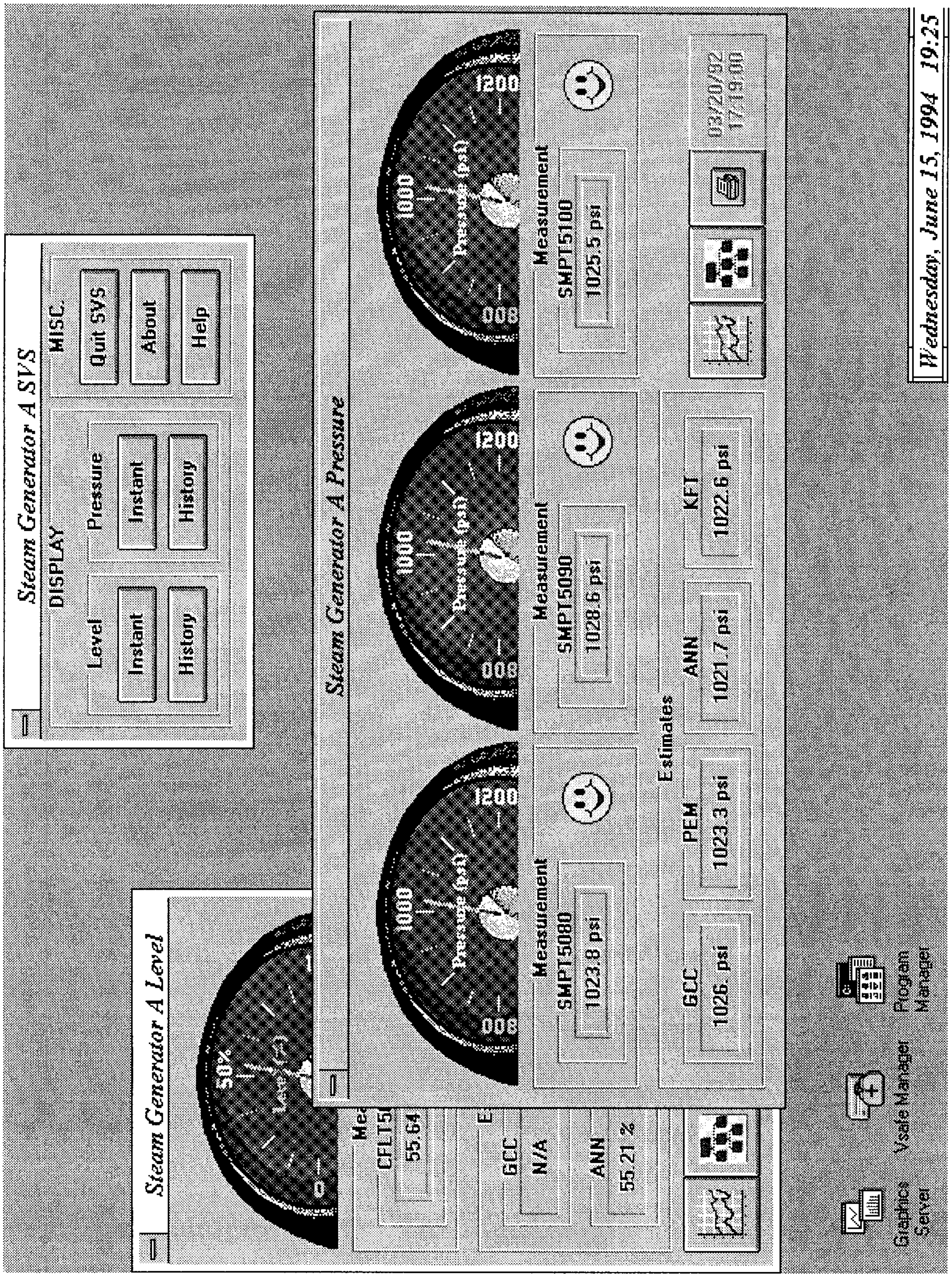

Figure 5.5: Initial GUI of the PC-based signal validation system.

Also, a historical trend plot of the measured and estimated values is of importance, to conclude a final decision about the system. The graphical plots are created with Visual Basic's extended custom control objects. Navigating to these plots are established by hypertext buttons, located at the border of each corresponding information window.

The results of the decision-making module are also displayed by means of modern techniques. If the sampling time is in the order of a minute or less, it may be difficult for the user to read out the final outcome of the fuzzy logic fault-tree in terms of linguistic values, such as *safe*, *no fault*, *fault warning*, *fault* and *severe fault*. Instead, icon representations of such values are used as shown in Figure 5.5. The icon "smiley" is used to indicate the final outcome of the decision-making module.

5.4 System Input / Output Operations

Another task, the system executive performs is the input / output (I/O) operation. Such operations include acquiring sensor data, feeding them to the SV modules, getting the results from the individual SV modules and displaying them to the user.

Acquiring data from the sensors is indirectly accomplished by using the data acquisition computer's capabilities. This computer collects and stores sensor data which represent several state values. These values are first stored in a certain block of memory, and then transferred to a hard disk (HD) in a compressed form. An interface program, running on the data acquisition computer extracts the sensor data to be used by the PC-based signal validation system and writes them to a file on the HD. The file has a time stamp at the beginning and is followed by the sensor outputs of interest. At each sampling time, the file is rewound and each of the data field is updated. An example of such an interface file is given in Figure 5.6.

The PC-based signal validation system uses the local area network (LAN) to access this file on the data acquisition computer at the plant. The file is opened in a shared mode, and for each new data sampling the system executive reads this file from the beginning. In this manner, the desired inputs to the SV modules are acquired. A schematic representation of this information access is shown in Figure 5.7.

It is important to install a proper network program in the PC. In this study SunSelect PC Networking File System™ and DEC Pathworks™ are used to access information over the local area network (LAN).

```
24-MAR-1994 12:32:21.37
   99.59474          18
   15063.58          18
   99.96840          18
   99.77785          18
   153.9693          18
   153.3074          18
   157.4243          18
   152.9930          18
   137.8695          18
   157.6095          18
   2.313216          18
   2.266341          18
   437.5445          18
   3408.549          18
   1179.498          18
   9896.985          18
```

Figure 5.6: A typical format of an interface file.

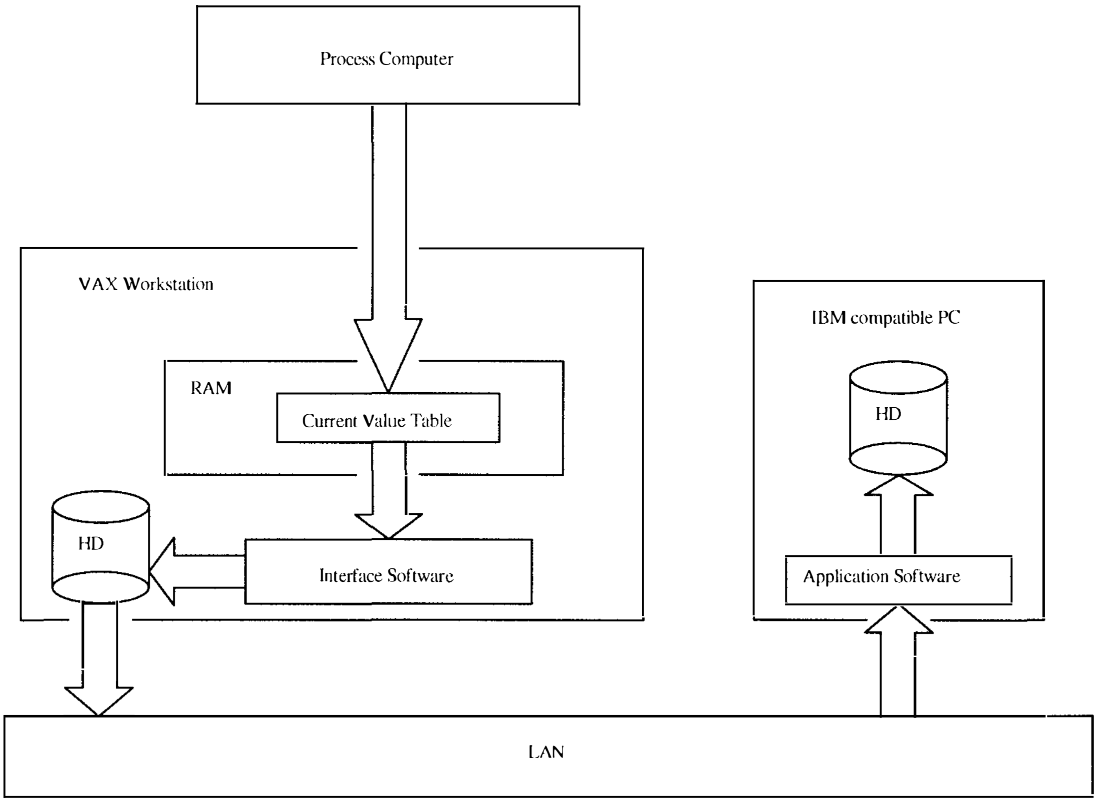

Figure 5.7: Information flow from process computer to the PC-based signal validation system.

Chapter 6

APPLICATIONS TO PWR PLANT Measurements

6.1 Introduction

The PC-based signal validation system uses different algorithms to validate the sensors of interest. While the generalized consistency checking (GCC) module takes advantage of having multiple channels (redundancy) for each state variable to be measured, other modules use different, but physically related measurements to make an estimation of the same state variable. The calculational diversity of having four different signal validation (SV) modules provides an effective monitoring of plant signals and reduces the probability of missing a fault due to model errors.

The GCC module was previously developed at The University of Tennessee, and was modified to be integrated as a library function into the PC-based signal validation system. The software for process empirical modeling (PEM) module was also developed in a previous research program. The resultant nonlinear equations produced by the PEM, were incorporated as a function in the program. The ANN's were created by a commercial software package called NeuralWorks [27]. The output of this package was a C subroutine, which was incorporated into the PC-based signal validation system as a library function.

The study was performed off-line using operational data from two different commercial PWR nuclear power plants. For porprietary reasons, they are referred to as PWR-1 and PWR-2. PWR-1 data consists of shut-down data while PWR-2 data consists of turbine-trip data. The developed signal validation platform is being modified for transfer to TVA's Sequoyah Nuclear Plant (SNP). In this chapter, each section describes results for each different nuclear power plant data and for each different computational structure for estimating the state variable of interest.

The steam generator level and steam generator pressure variables of a UTSG in a four-loop pressurized water reactor plant were chosen as examples for testing the signal validation modules. Each module was tested individually for these two state variables, and the PC-based program displays information about these two measurements in separate information windows.

6.2 Generalized Consistency Checking and Sequential Probability Ratio Test

The generalized consistency checking (GCC) was performed on two different measurements: steam generator narrow range water level and steam generator pressure. The steam generator wide range water level has only one measurement channel.

Therefore the consistency checking is not applicable. The corresponding signals used in this study are shown in Table 6.1.

The probabilities of false (α) and missed (β) alarms were specified to be 0.0015, while the sensor standart deviation (σ) for the steam generator narrow range water level was taken to be 0.33% and for the steam generator pressure to be 2 psi.

The GCC estimates as shown in Figures 6.1, 6.4, 6.5 and 6.7 are calculated according to Equation (2.2). The GCC module found inconsistencies in steam generator narrow range water level channel 4 for PWR-1 as shown in Figure 6.2 and recorded a sensor degradation as shown in Figure 6.3. This sensor value was excluded from calculations for 98.46% of the samples.

The GCC module did not detect any fault in the steam generator narrow range water level for PWR-2 or steam generator pressure for PWR-1 (Figure 6.6). However, steam generator pressure channel 3 for PWR-2 was recorded to be faulty as shown in Figures 6.8 and 6.9.

Table 6.1: Signal list used by generalized consistency checking.

Measurement	PWR-1 Tag #	PWR-2 Tag #
Steam Generator A Narrow Range Water Level, Channel 1	CFLT6000	FWS-L517
Steam Generator A Narrow Range Water Level, Channel 2	CFLT5510	FWS-L518
Steam Generator A Narrow Range Water Level, Channel 3	CFLT5500	FWS-L519
Steam Generator A Narrow Range Water Level, Channel 4	CFLT5490	N/A
Steam Generator A Pressure, Channel 1	SMPT5080	MSS-P514
Steam Generator A Pressure, Channel 2	SMPT5090	MSS-P515
Steam Generator A Pressure, Channel 3	SMPT5100	MSS-P516

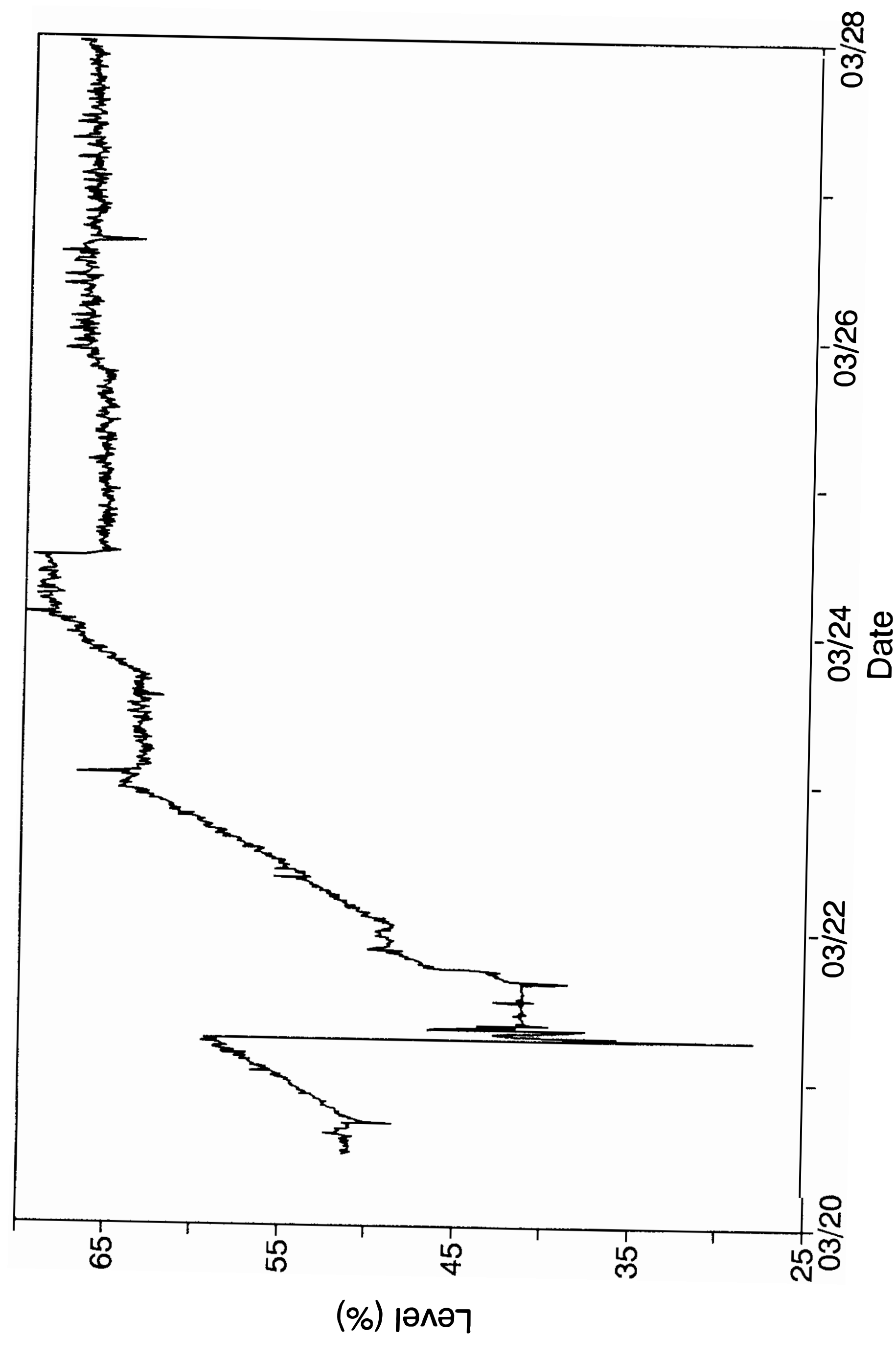

Figure 6.1: GCC estimate of steam generator narrow range water level for PWR-1.

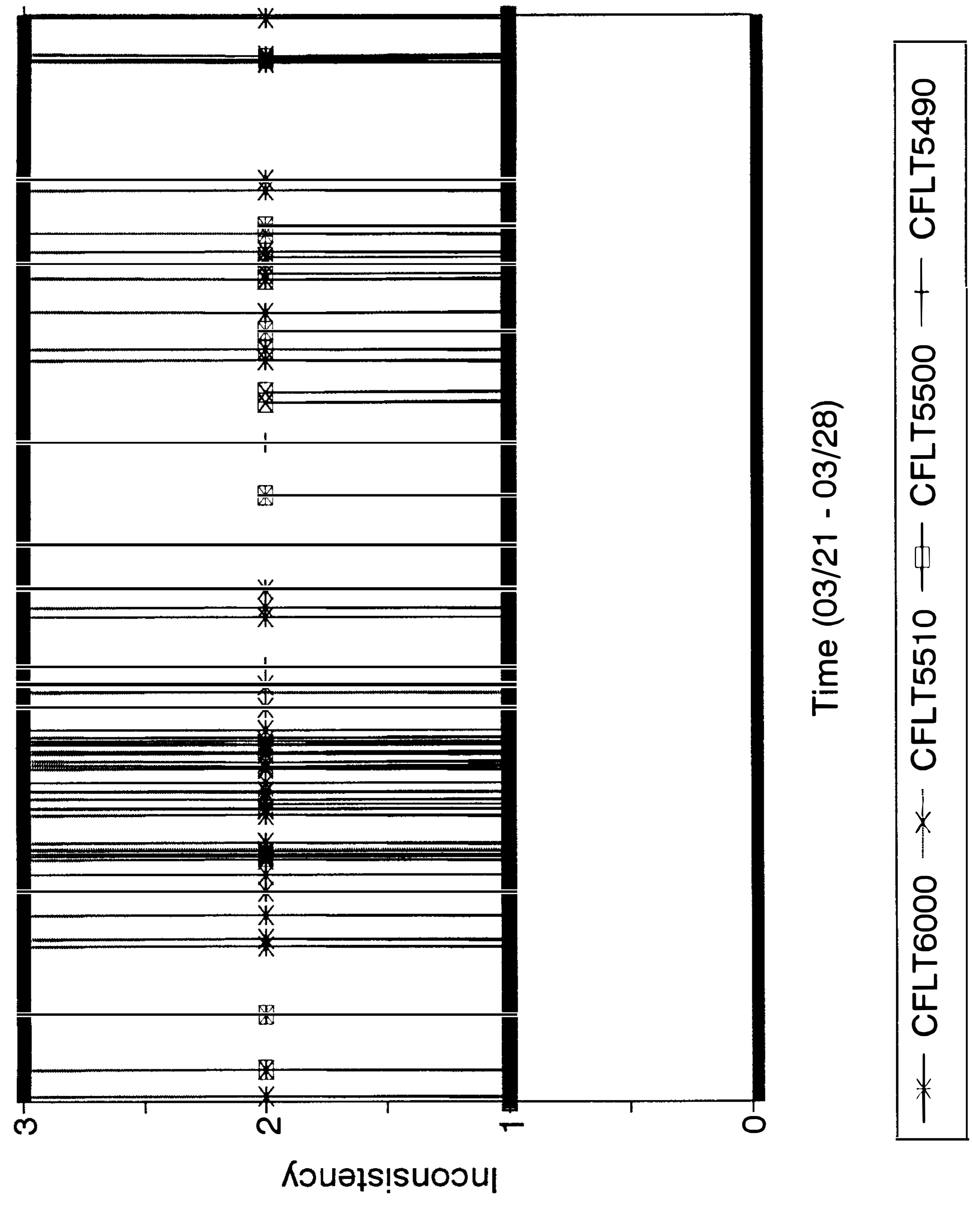

Figure 6.2: Inconsistency indices computed by GCC for the steam generator narrow range water level for PWR-1.

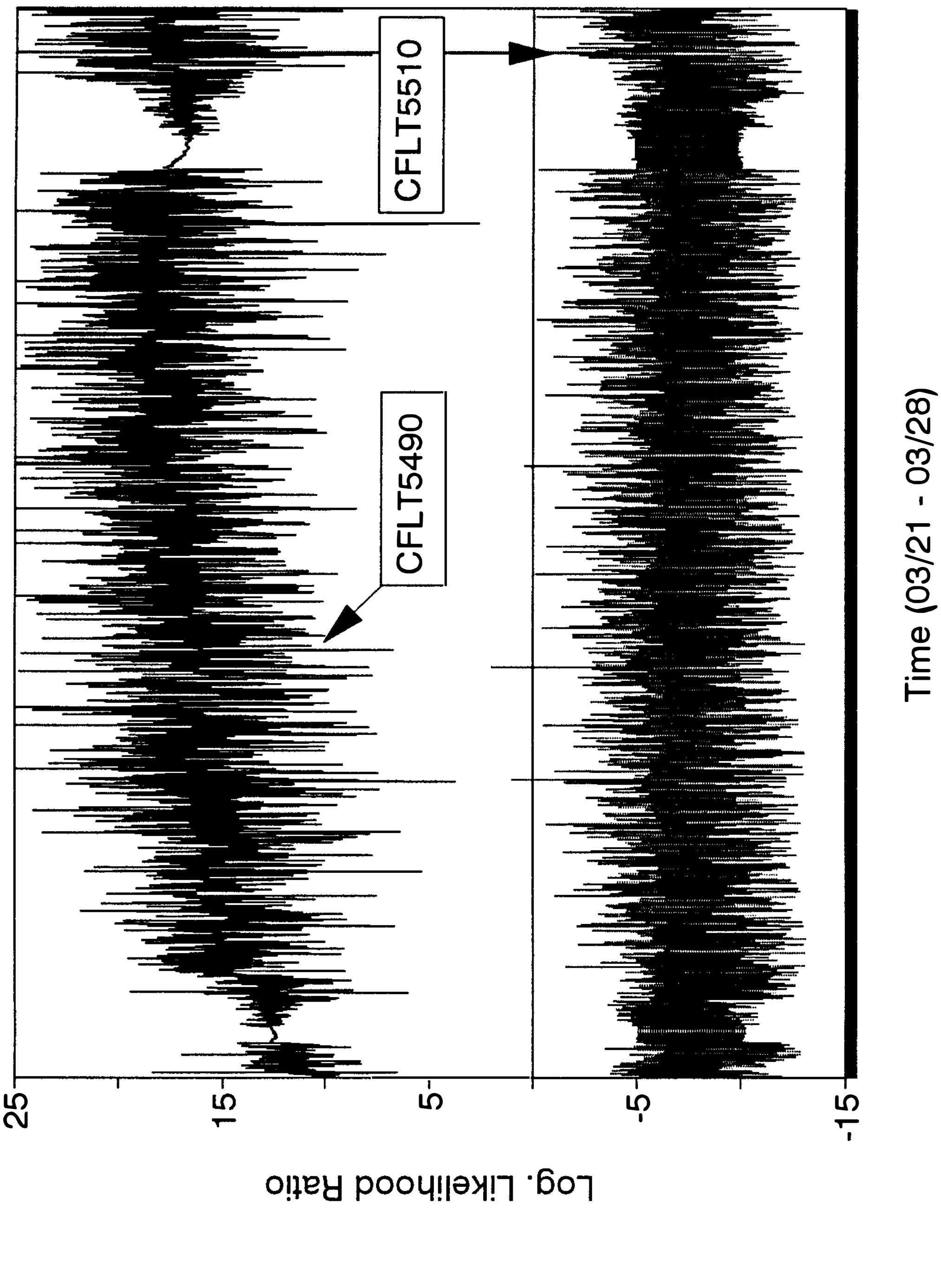

Figure 6.3: Log likelihood ratios computed by GCC for the steam generator narrow range water level for PWR-1.

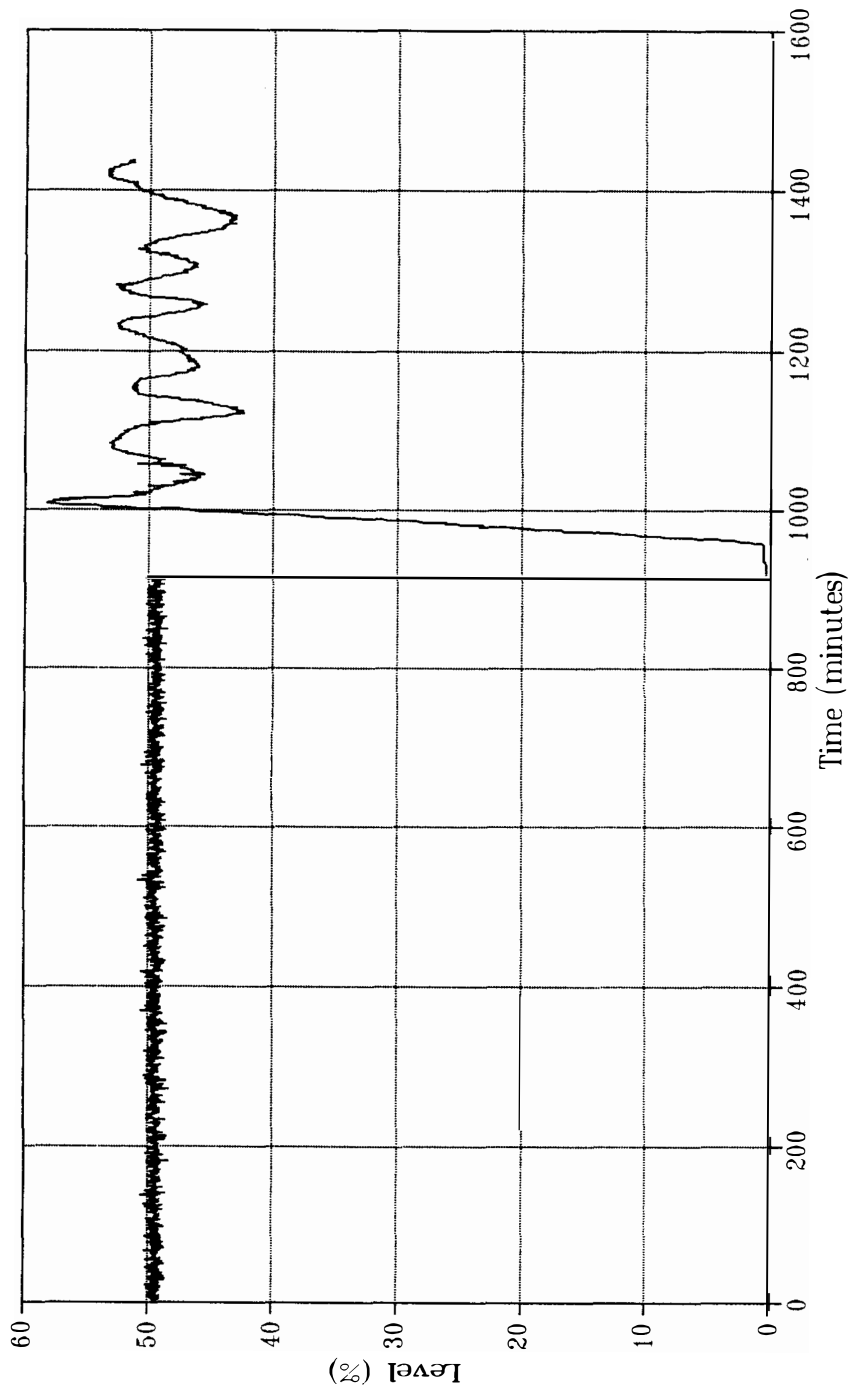

Figure 6.4: GCC estimate of steam generator narrow range water level for PWR-2.

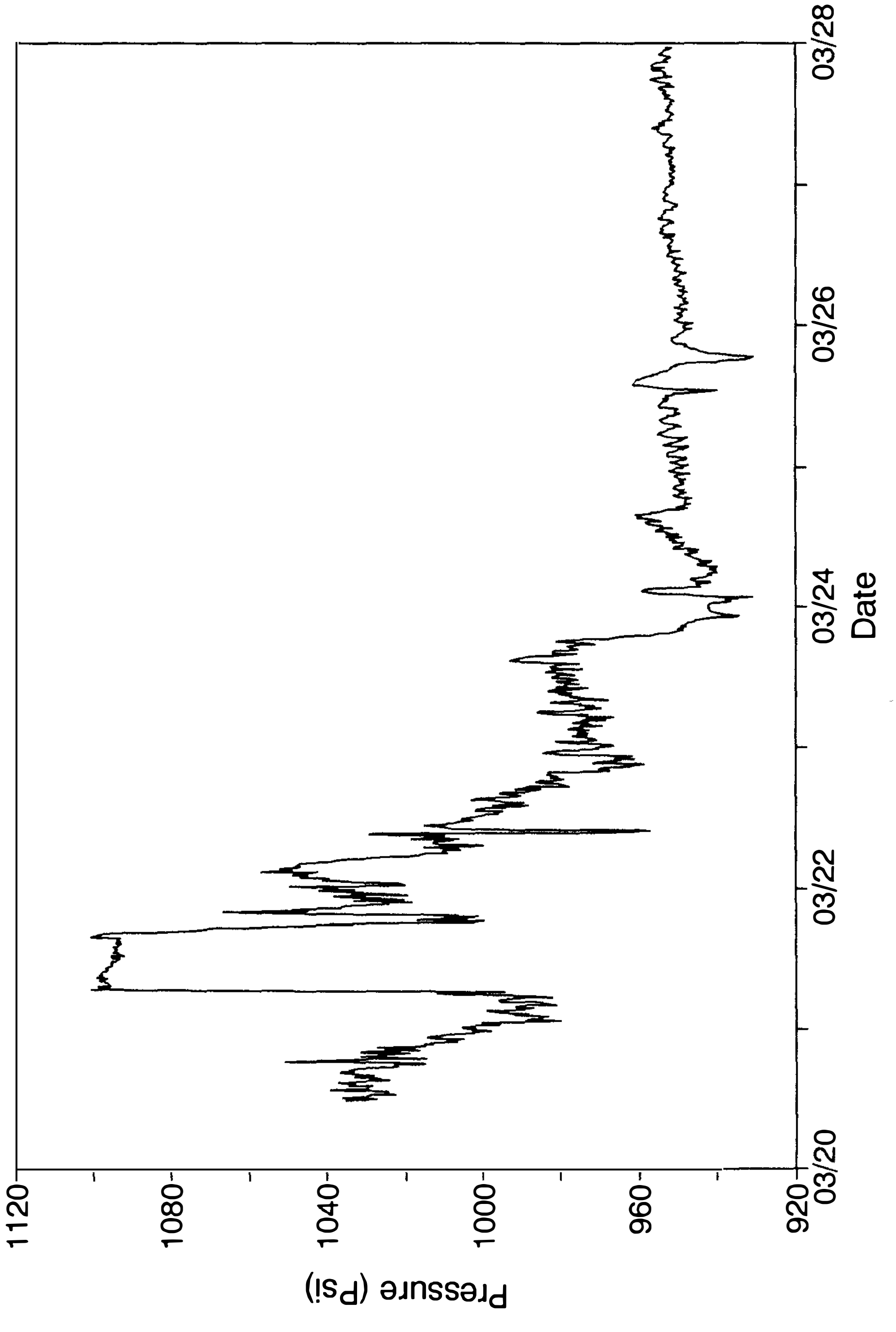

Figure 6.5: GCC estimate of steam generator pressure for PWR-1.

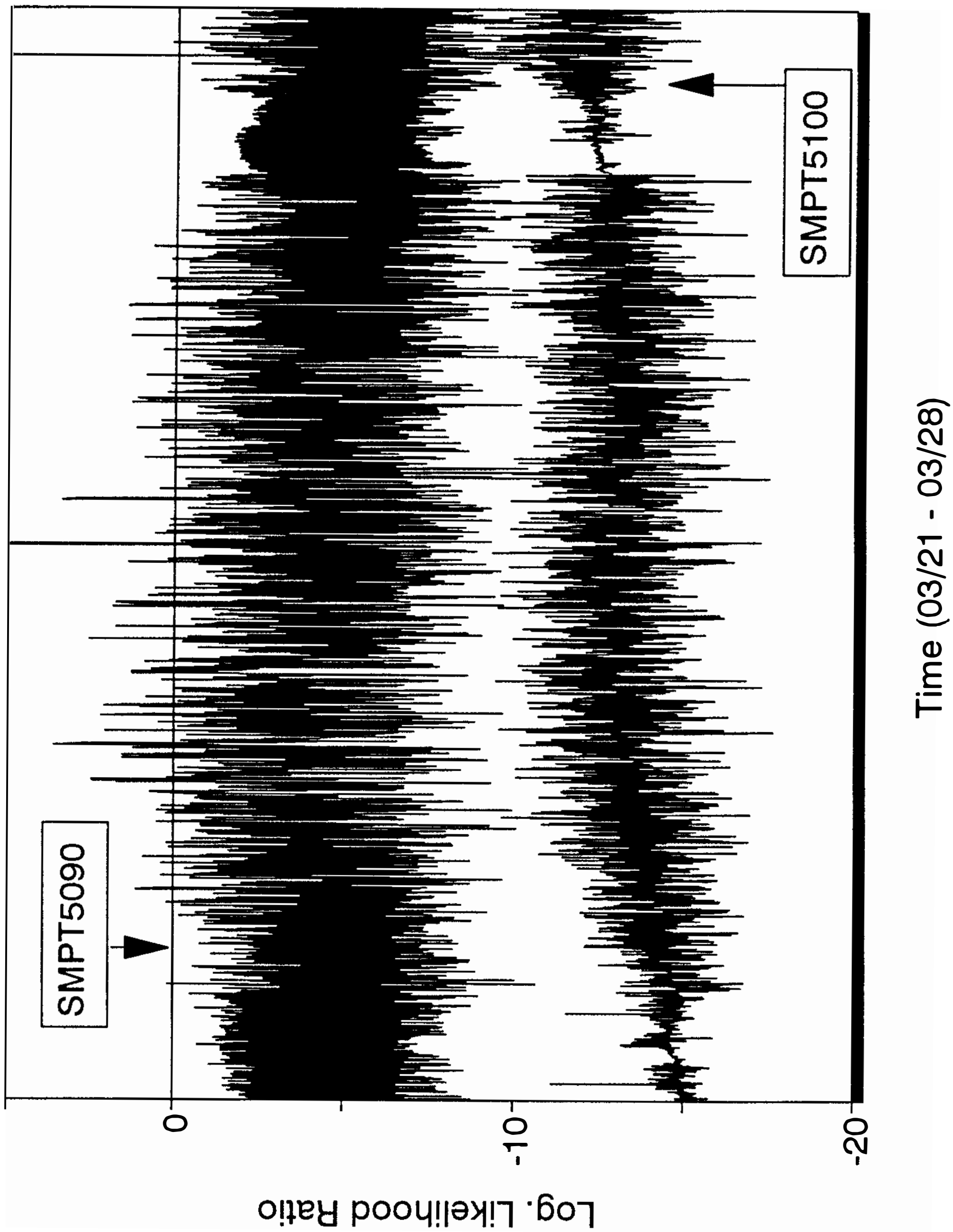

Figure 6.6: Log likelihood ratios computed by GCC for the steam generator pressure for PWR-1.

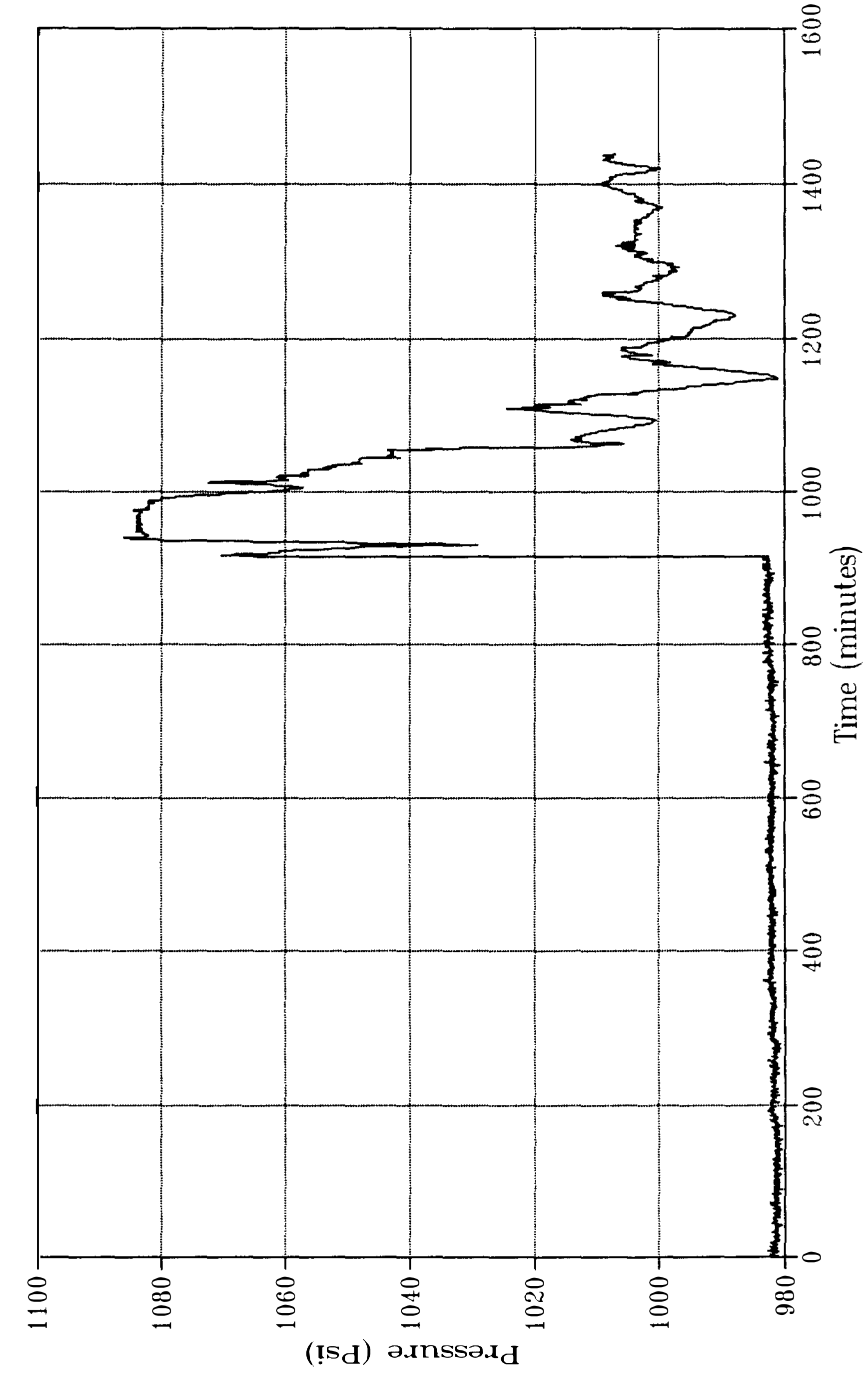

Figure 6.7: GCC estimate of steam generator pressure for PWR-2.

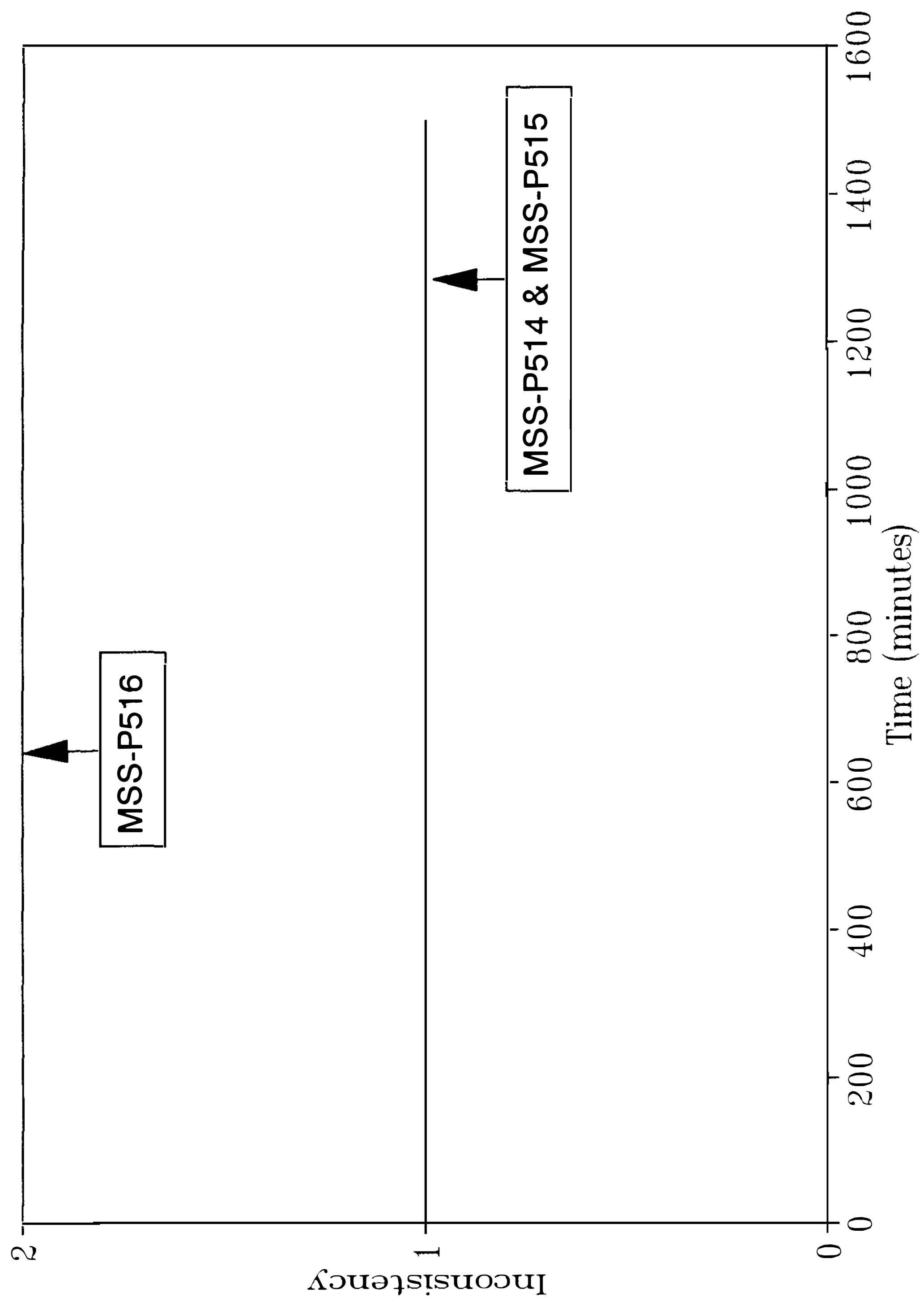

Figure 6.8: Inconsistency indices computed by GCC for the steam generator pressure for PWR-2.

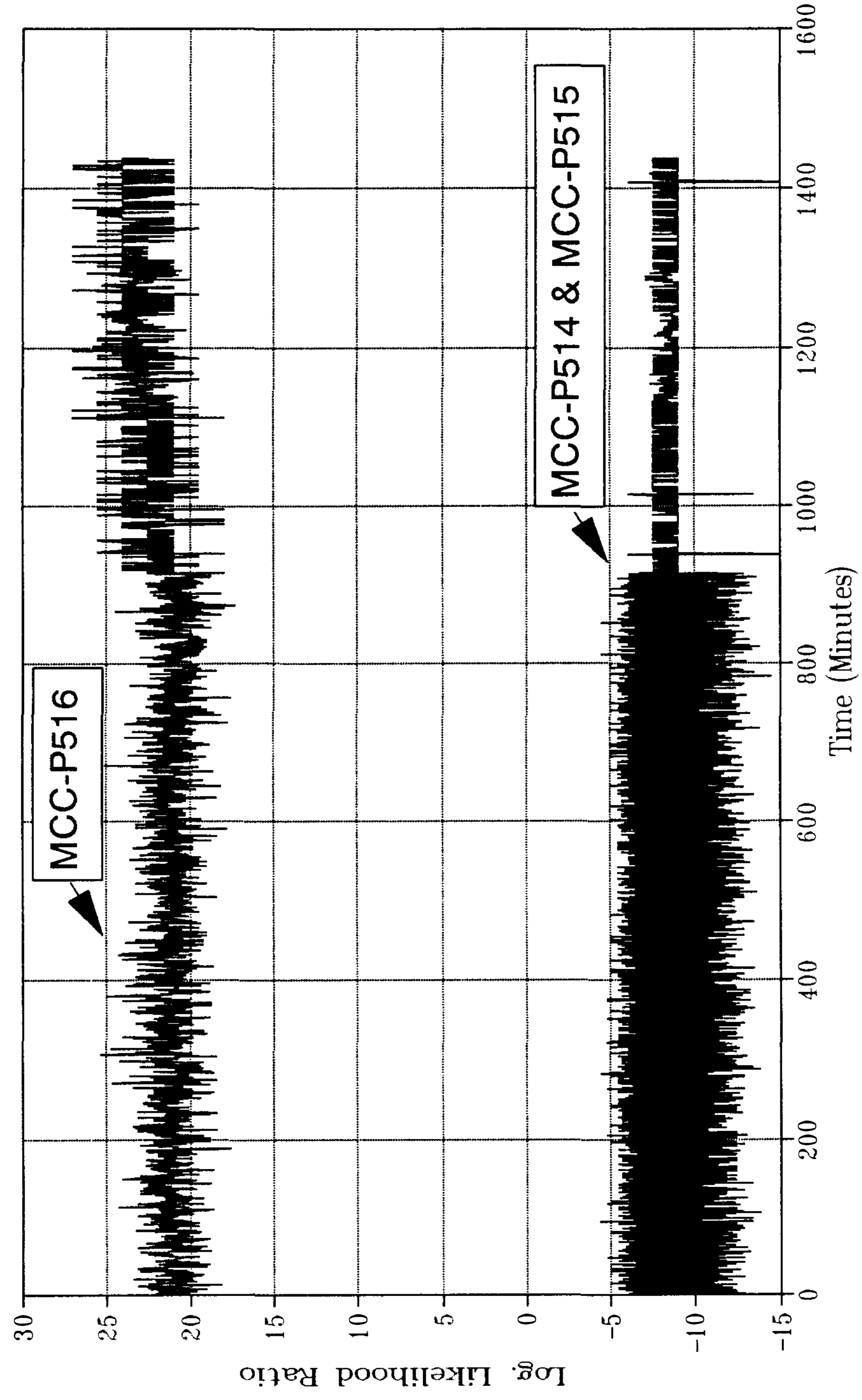

Figure 6.9: Log likelihood ratios computed by GCC of the steam generator pressure for PWR-2.

6.3 Process Empirical Modeling

The process empirical modeling (PEM) was performed for two variables: steam generator wide range water level and steam generator pressure. Data from PWR-1 and PWR-2 were used for developing empirical models. Tables 6.2 and 6.3 show functional forms and results of the PEM module for these two different data sets with the following input signals.

$x(1)$ = steam generator main feedwater flow rate,

$x(2)$ = steam generator wide range water level at previous time instant,

$x(3)$ = reactor coolant system (RCS) flow rate,

$x(4)$ = steam generator steam flow rate,

$x(5)$ = steam generator steam pressure at previous time instant,

$x(6)$ = hot leg temperature, and

$x(7)$ = cold leg temperature.

The models were created using 100 training patterns, which were sampled at regular intervals over the entire data interval. The PEM models (Appendix B) were incorporated into the PC-based signal validation system. As it is presented in the tables and figures of this section, dynamic models (model # 3, 4, 7 and 8) improved the PEM estimation. However, in some cases a static model was adequate for estimation (model # 6). The graphical representations of the estimations of the models are shown in Figures 6.10

Table 6.2: Process empirical models using PWR-1 data.

Model #	Figure	Modeled State Variable	Model	Constants	Modeling Error
1	6.10 & 6.11	Steam Generator Water Level	$c_1x(6)x(7)^2 + c_2x(1) + c_3x(1)^2 + c_4x(1)^3 + c_5$	$c_1 = -9.1x10^{-8}$ $c_2 = 0.074$ $c_3 = 0.002$ $c_4 = -2.0x10^{-5}$ $c_5 = 71.21$	0.84%
2	6.12 & 6.13	Steam Generator Pressure	$c_1x(3) + c_2x(7) + c_3x(1) + c_4x(4) + c_5$	$c_1 = 0.344$ $c_2 = 14.103$ $c_3 = -1.018$ $c_4 = -15.720$ $c_5 = -681.493$	0.54%
3	6.14 & 6.15	Steam Generator Water Level	$c_1x(1) + c_2x(1)x(7) + c_3x(2)^2 + c_4x(1) + c_5x(6)^2 + c_6$	$c_1 = 0.295$ $c_2 = -1.8x10^{-4}$ $c_3 = -9.1x10^{-4}$ $c_4 = 0.190$ $c_5 = -7.4x10^{-5}$ $c_6 = 56.866$	0.70%
4	6.16 & 6.17	Steam Generator Pressure	$c_1x(5) + c_2x(7) + c_3x(2) + c_4x(6) + c_5x(1) + c_6$	$c_1 = 0.810$ $c_2 = 1.820$ $c_3 = 0.587$ $c_4 = -0.411$ $c_5 = -0.060$ $c_6 = -611.886$	0.21%

Table 6.3: Process empirical models using PWR-2 data.

Model #	Figure	Modeled State Variable	Model	Constants	Modeling Error
5	6.18	Steam Generator Water Level	$c_1x(3)^2 + c_2x(7) + c_3x(7)^2 + c_4x(6) + c_5x(6)^2 + c_6$	$c_1 = -0.003$ $c_2 = 59.869$ $c_3 = -0.054$ $c_4 = -16.315$ $c_5 = 0.014$ $c_6 = -1176.855$	4.13%
6	6.19	Steam Generator Pressure	$c_1x(7) + c_2x(4) + c_3x(6) + c_4x(1) + c_5$	$c_1 = 8.311$ $c_2 = -13.888$ $c_3 = 0.121$ $c_4 = -0.016$ $c_5 = -3633.133$	0.31%
7	6.20 & 6.21	Steam Generator Water Level	$c_1x(2) + c_2x(7) + c_3x(6) + c_4x(1) + c_5x(4) + c_6$	$c_1 = 0.993$ $c_2 = -.040$ $c_3 = 0.041$ $c_4 = 0.001$ $c_5 = -2.266$ $c_6 = .331$	0.47%
8	6.22 & 6.23	Steam Generator Pressure	$c_1x(5) + c_2x(7) + c_3x(6) + c_4x(1) + c_5x(4) + c_6$	$c_1 = 1.019$ $c_2 = 0.190$ $c_3 = -0.368$ $c_4 = 0.003$ $c_5 = 4.348$ $c_6 = 78.404$	0.09%

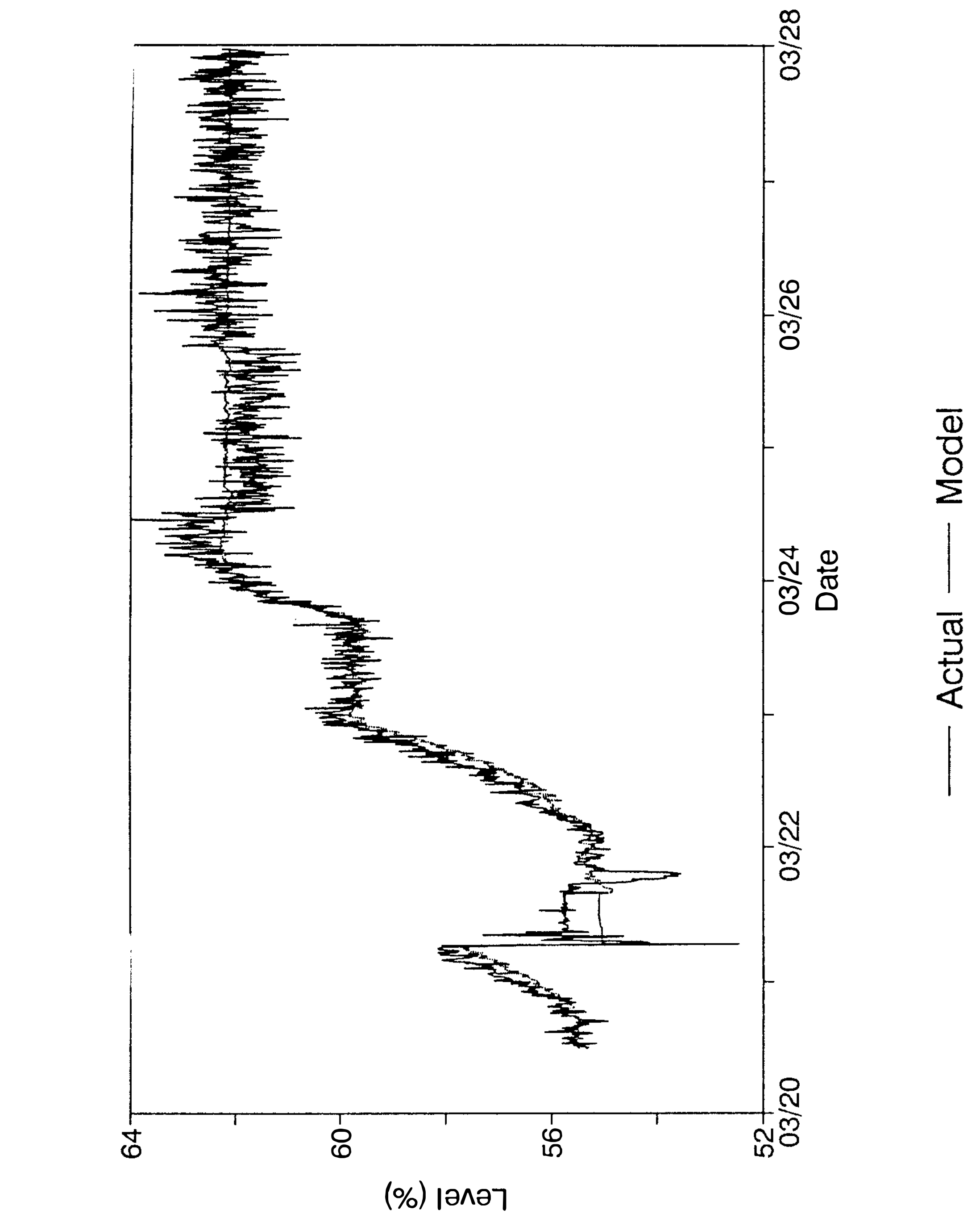

Figure 6.10: PEM estimate of steam generator wide range water level for PWR-1 using static modeling.

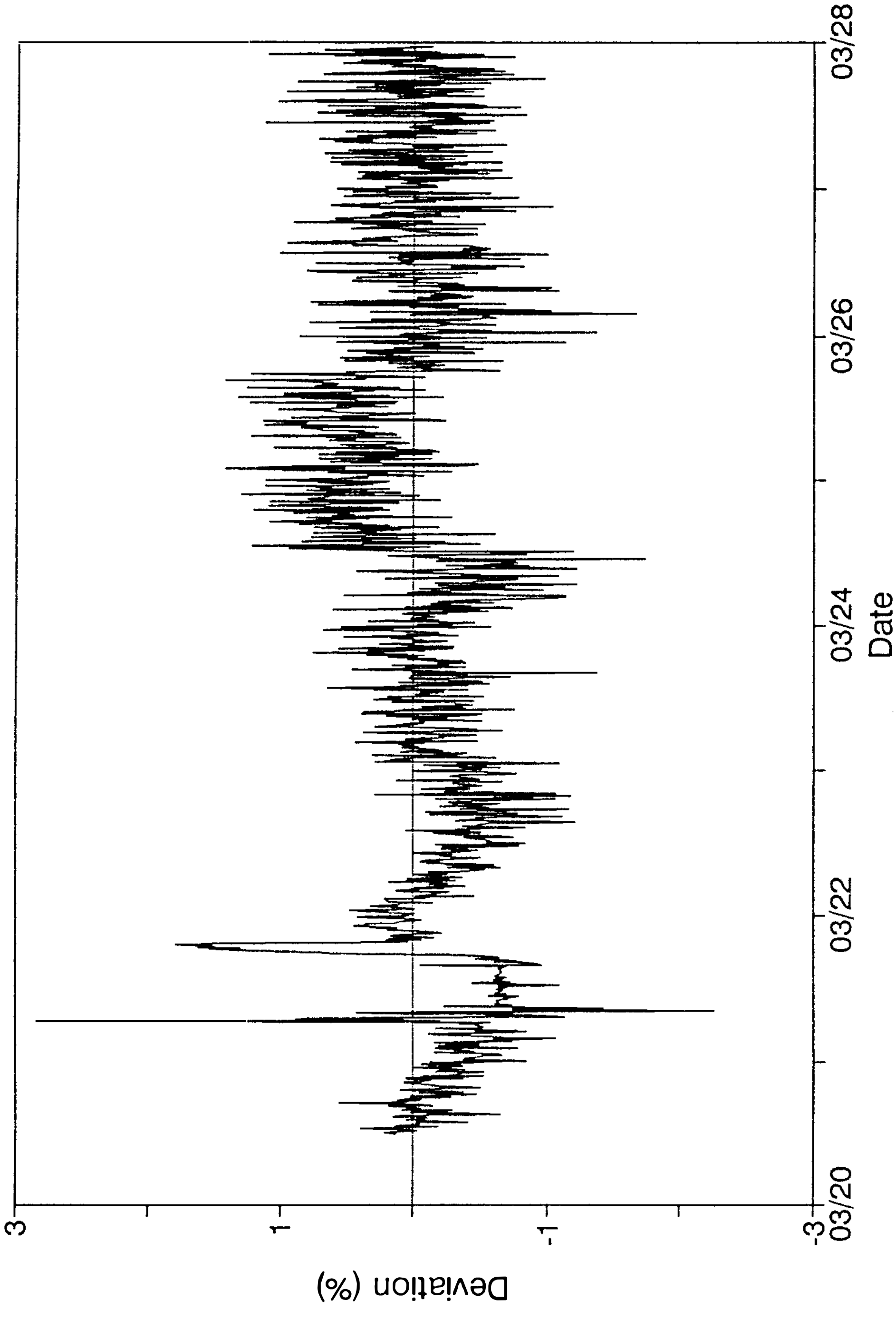

Figure 6.11: Error in PEM estimation shown in Figure 6.10.

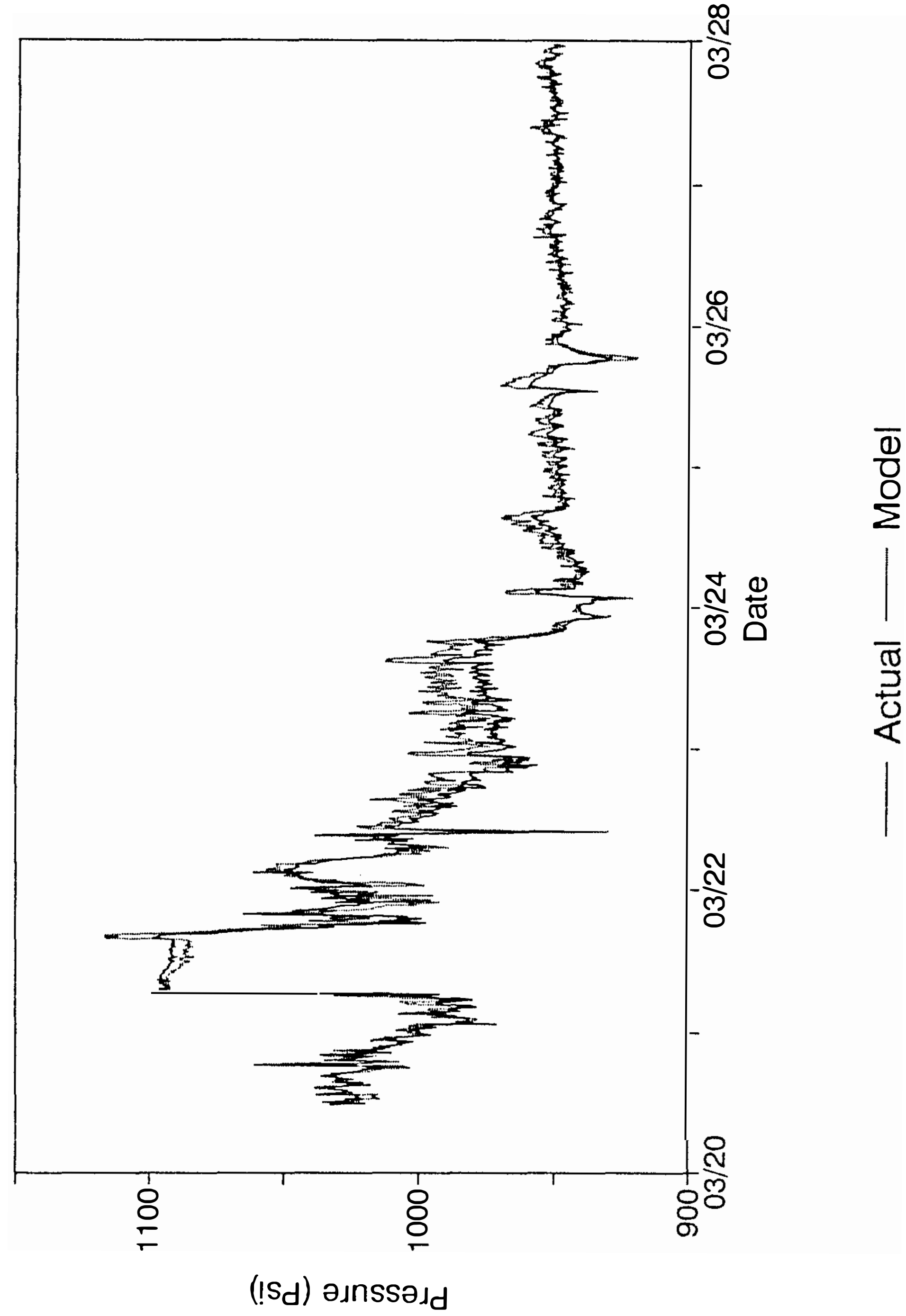

Figure 6.12: PEM estimation of steam generator pressure for PWR-1 using static modeling.

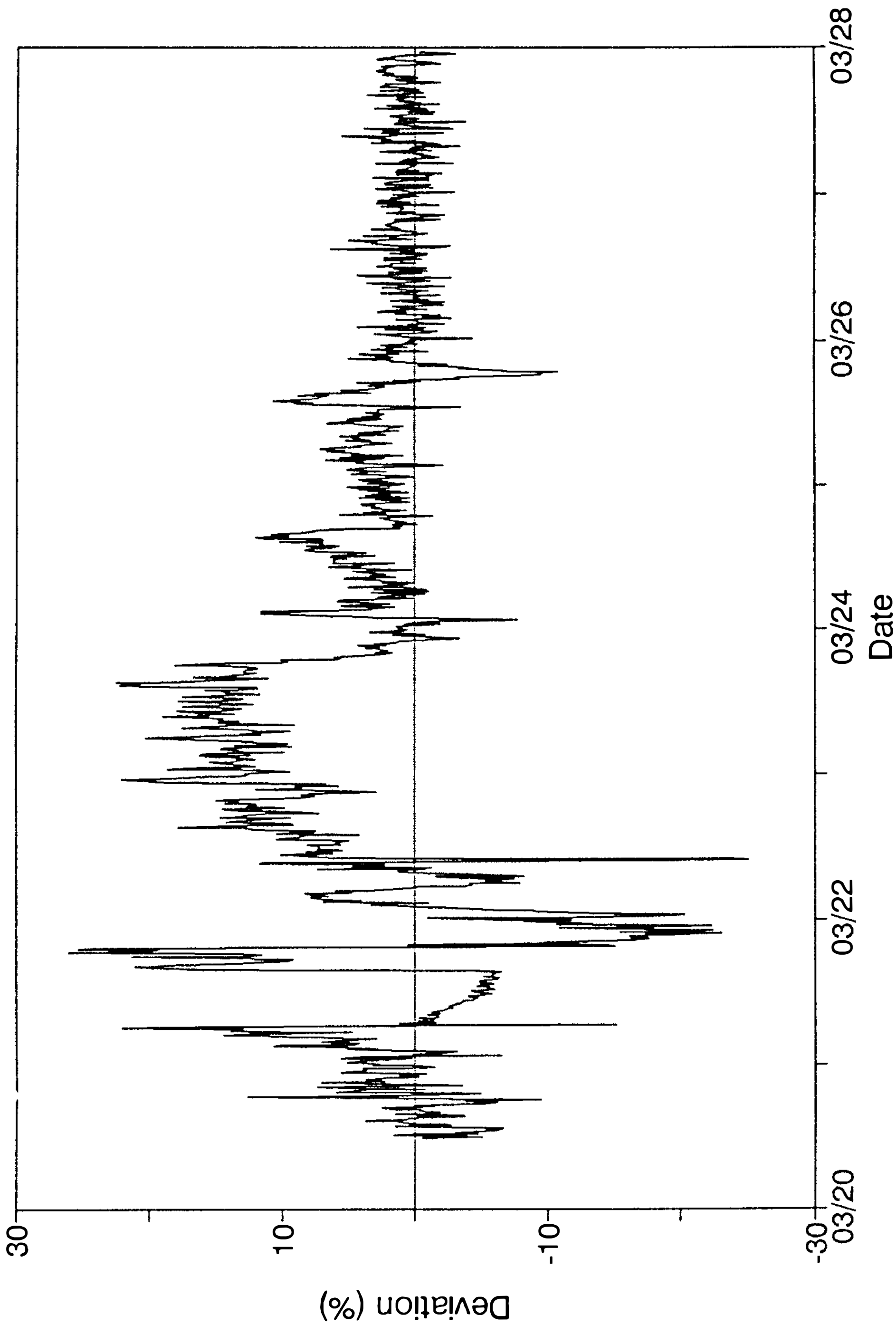

Figure 6.13: Error in PEM estimation shown in Figure 6.12.

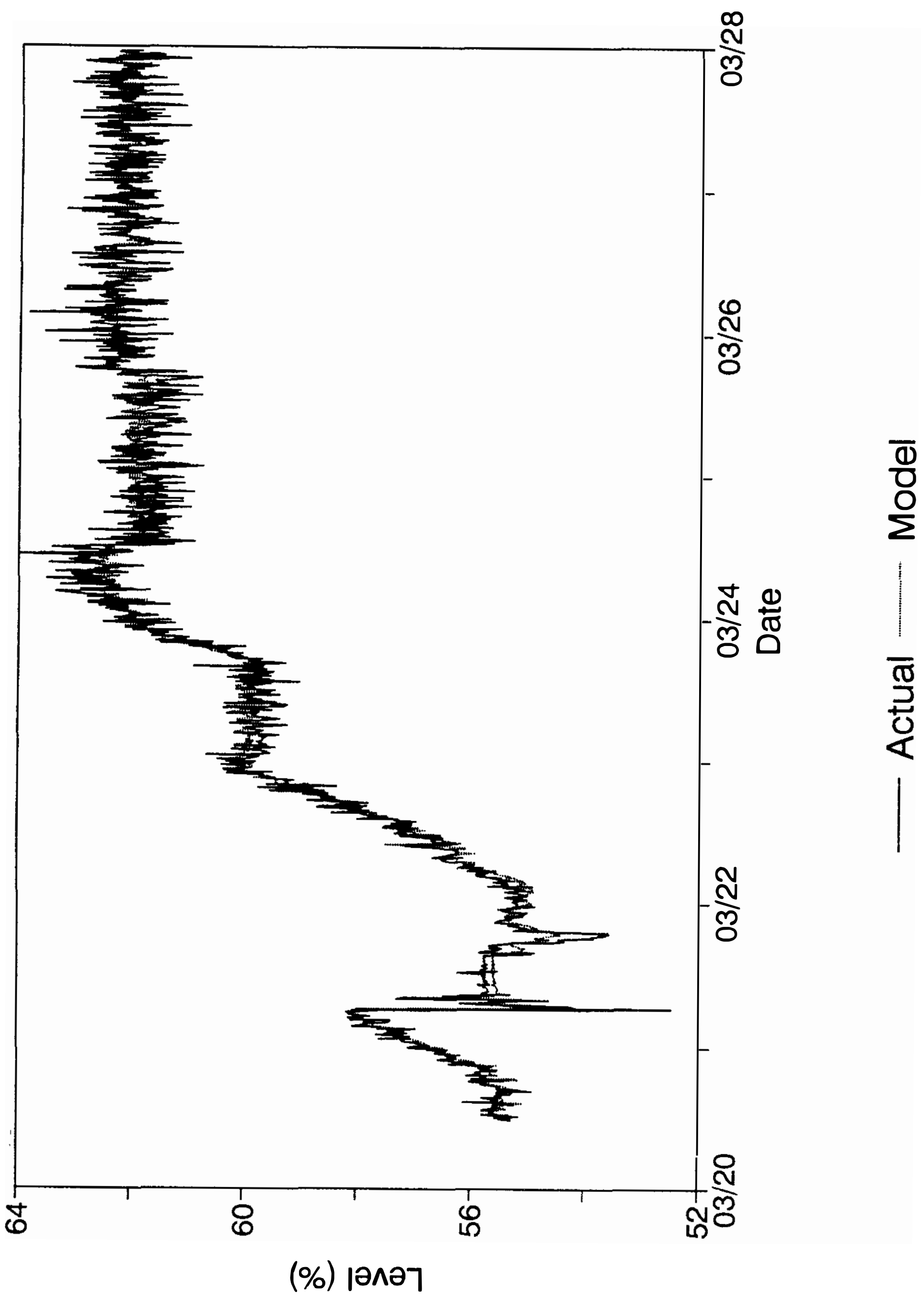

Figure 6.14: PEM estimate of steam generator wide range water level for PWR-1 using dynamic modeling.

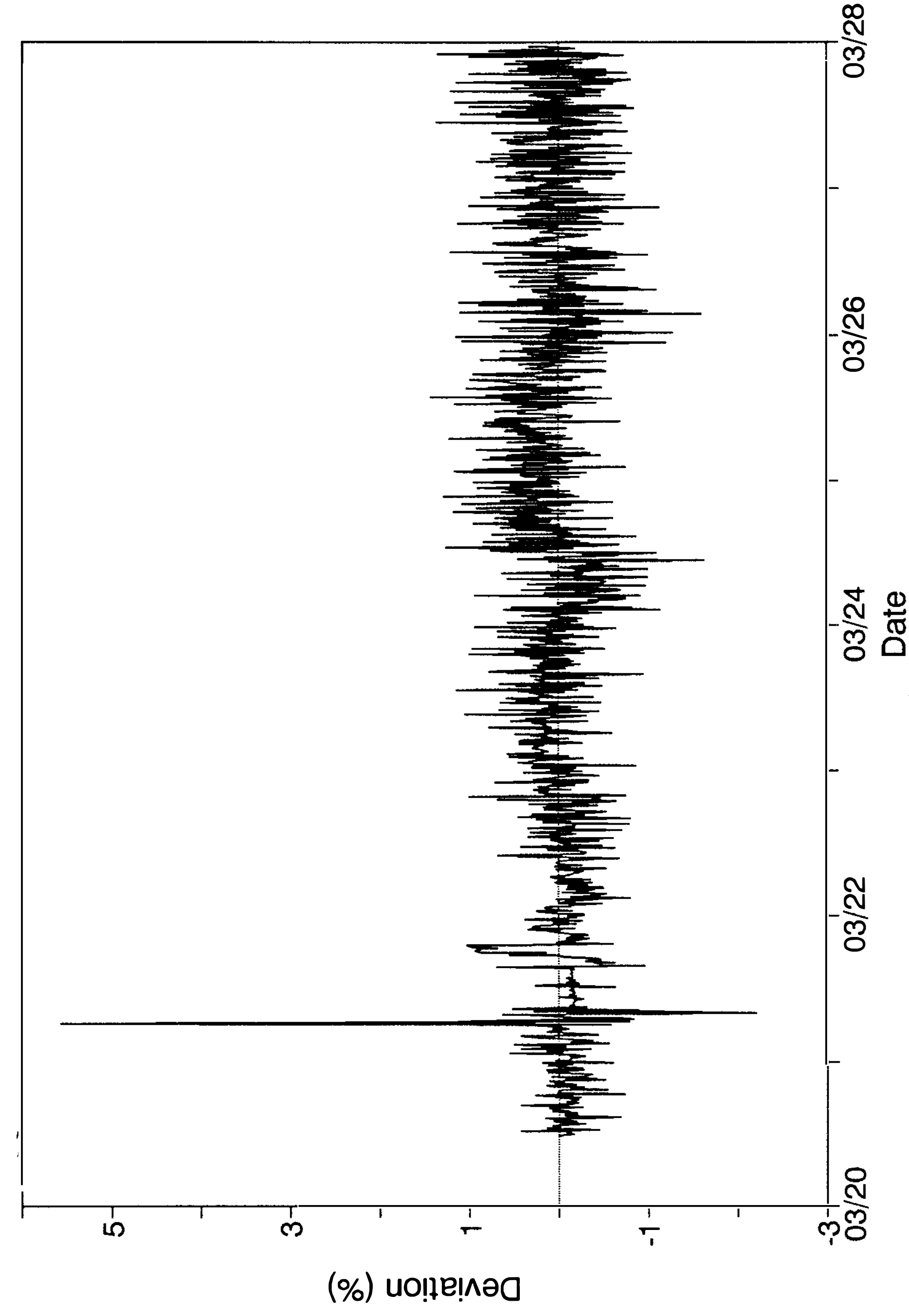

Figure 6.15: Error in PEM estimation shown in Figure 6.14.

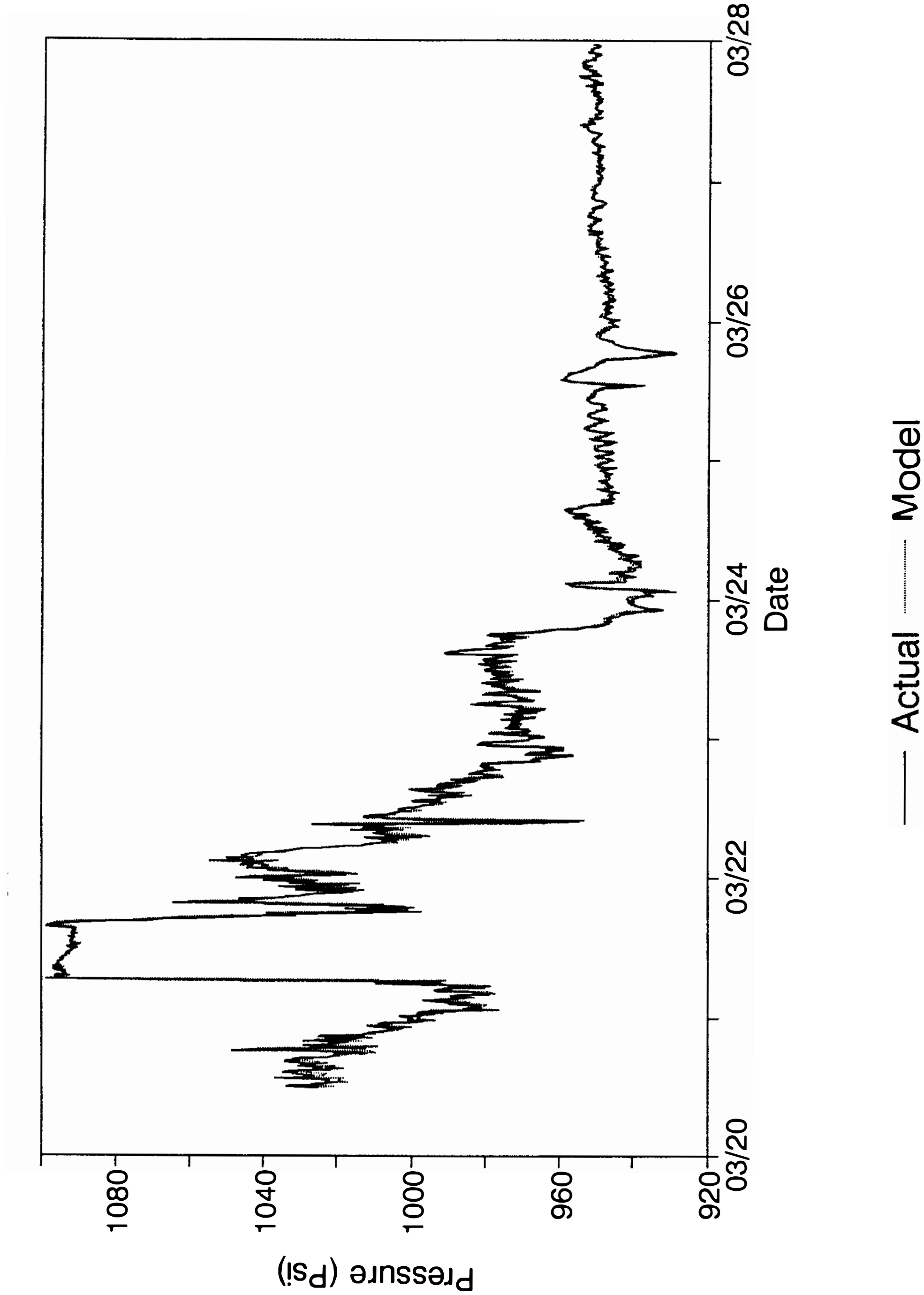

Figure 6.16: PEM estimation of steam generator pressure for PWR-1 using dynamic modeling.

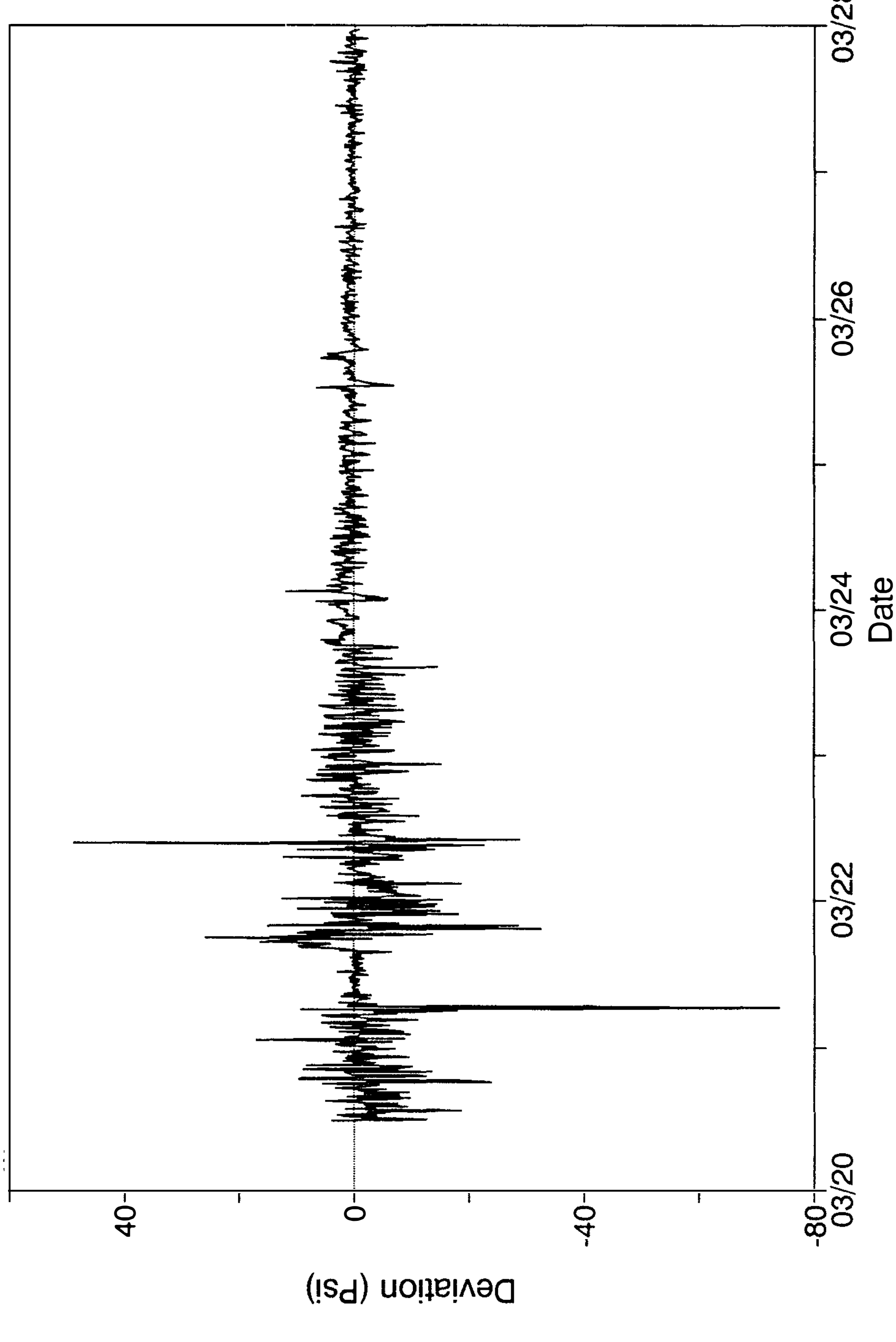

Figure 6.17: Error in PEM estimation shown in Figure 6.16.

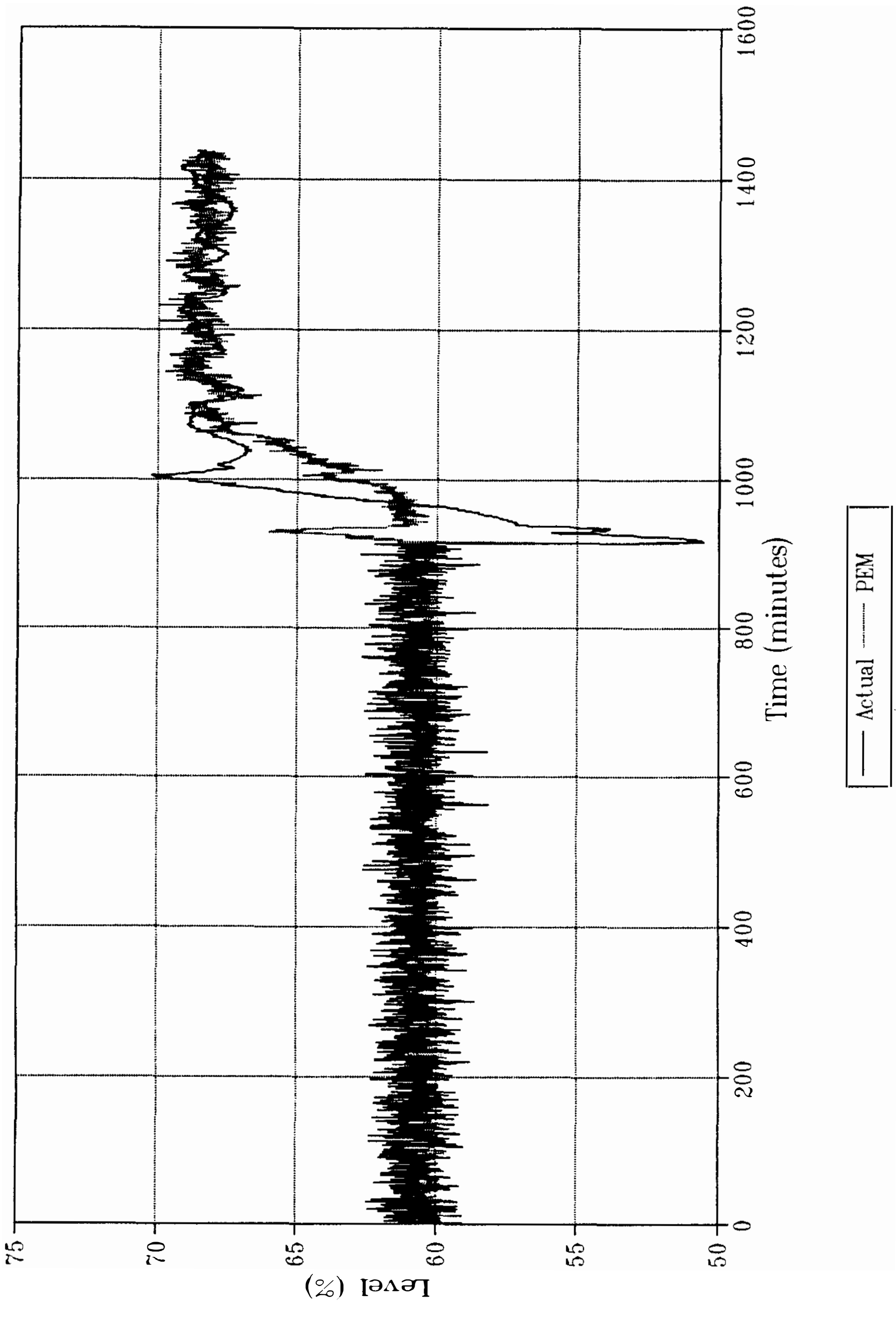

Figure 6.18: PEM estimate of steam generator wide range water level for PWR-2 using static modeling.

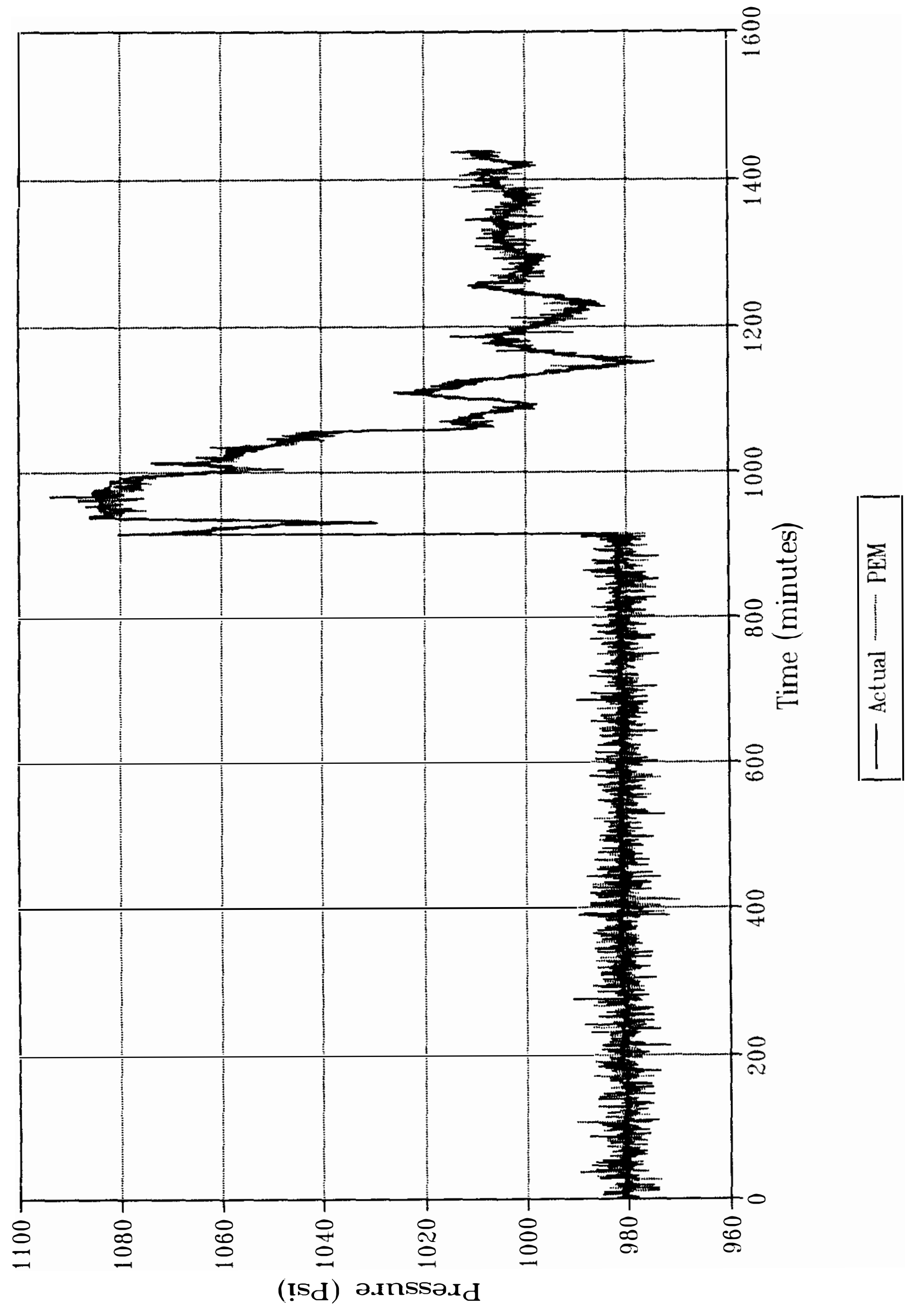

Figure 6.19: PEM estimate of steam generator pressure for PWR-2 using static modeling.

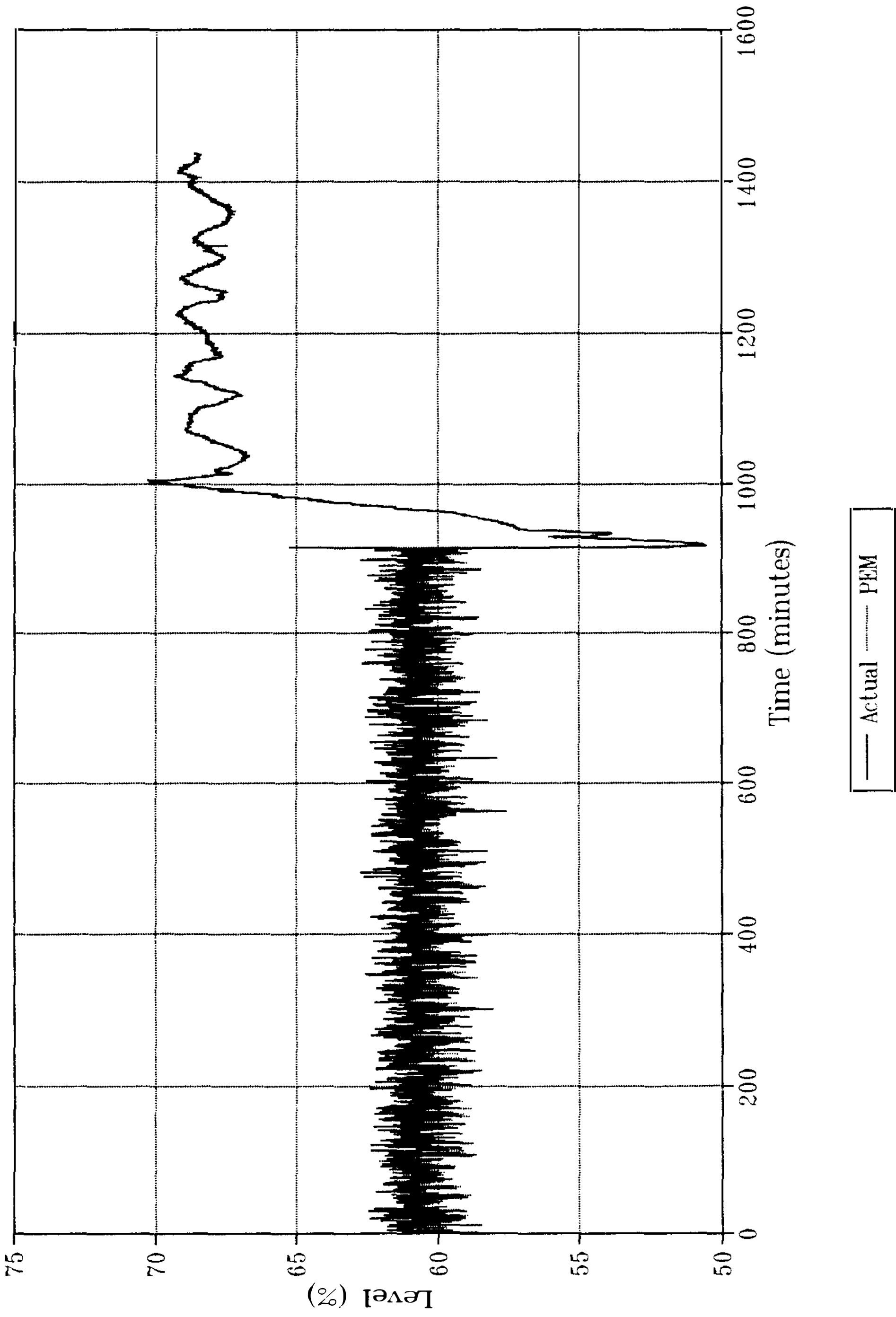

Figure 6.20: PEM estimate of steam generator wide range water level for PWR-2 using dynamic modeling.

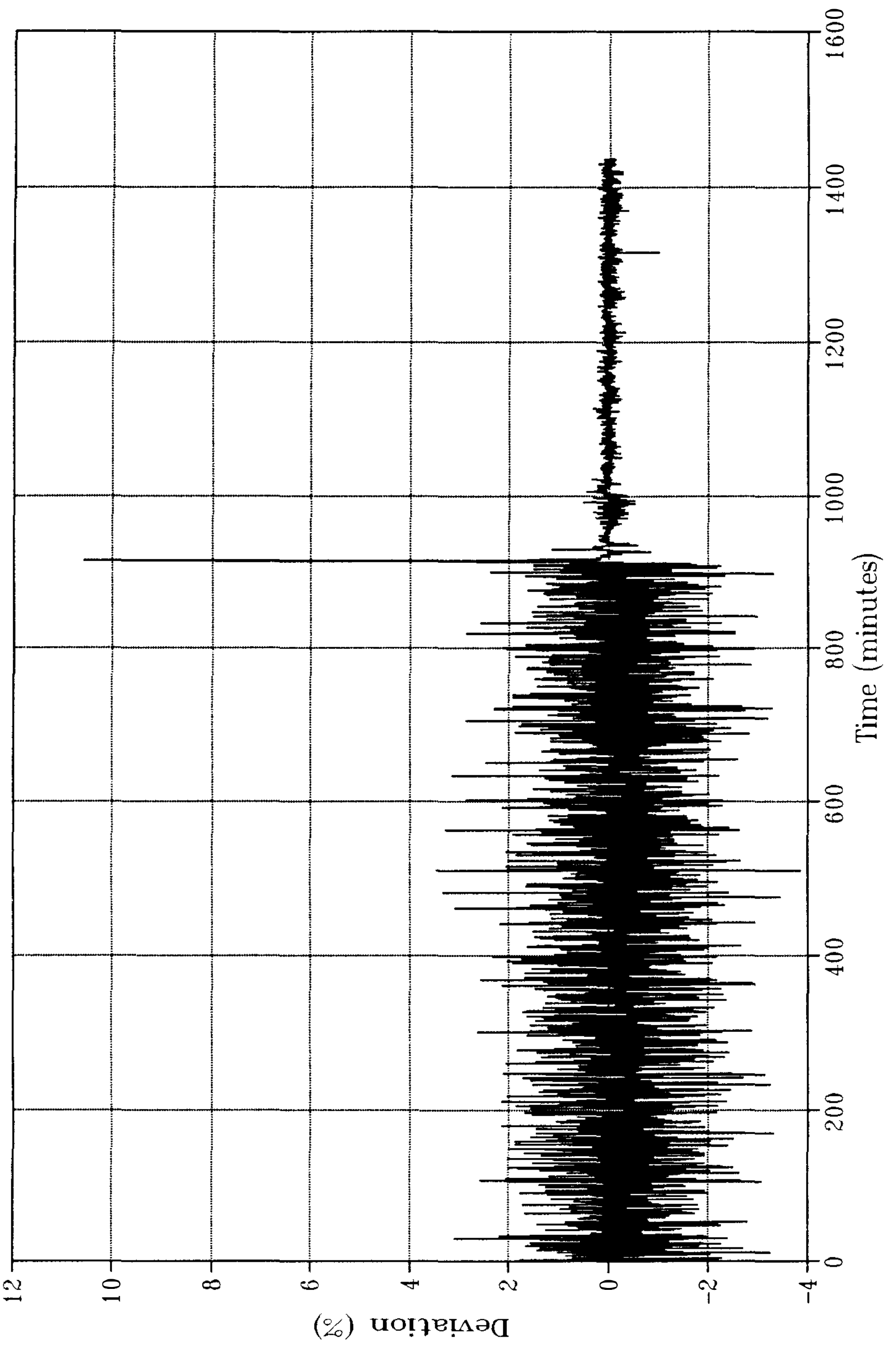

Figure 6.21: Error in PEM estimation shown in Figure 6.20.

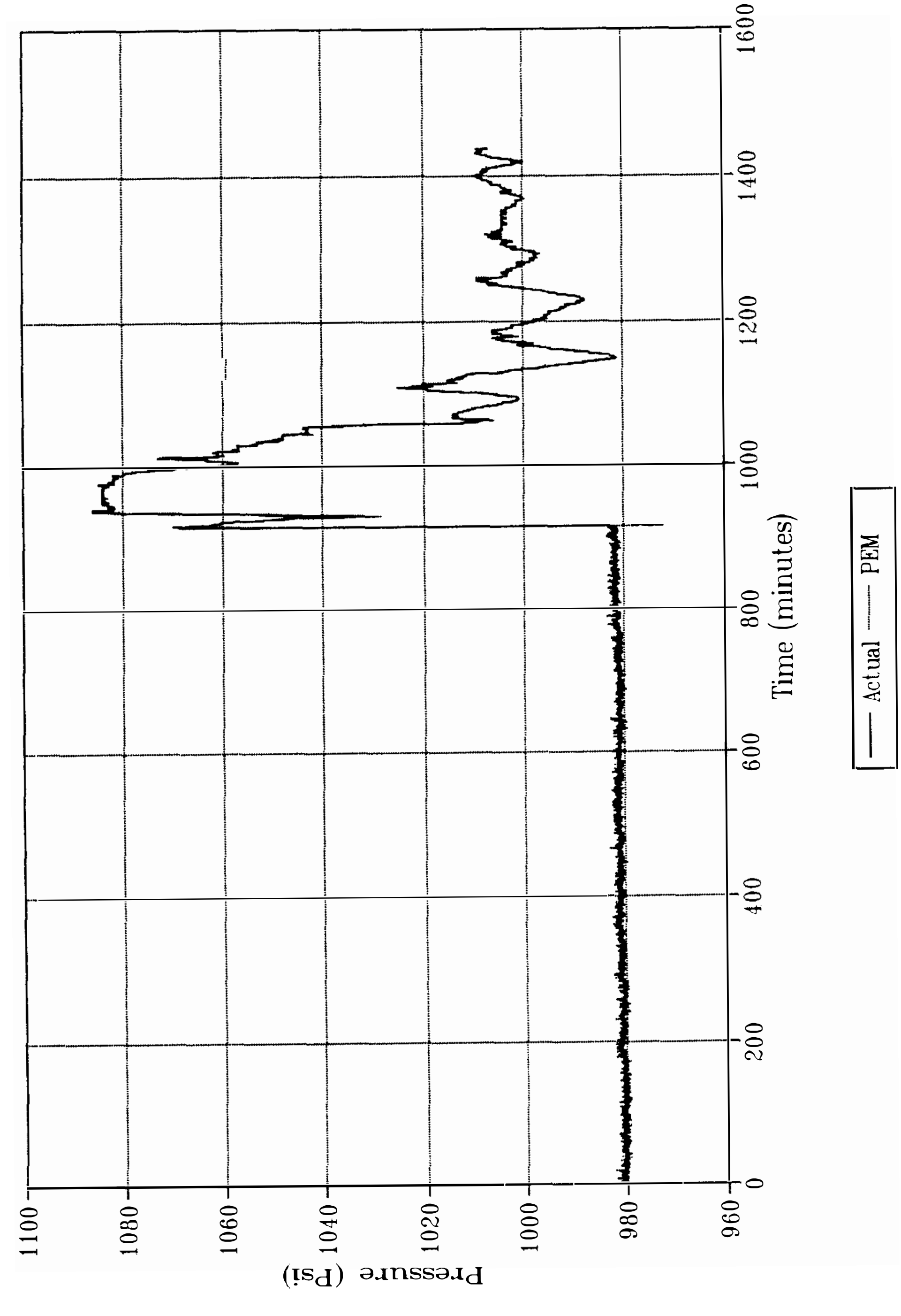

Figure 6.22: PEM estimate of steam generator pressure for PWR-2 using dynamic modeling.

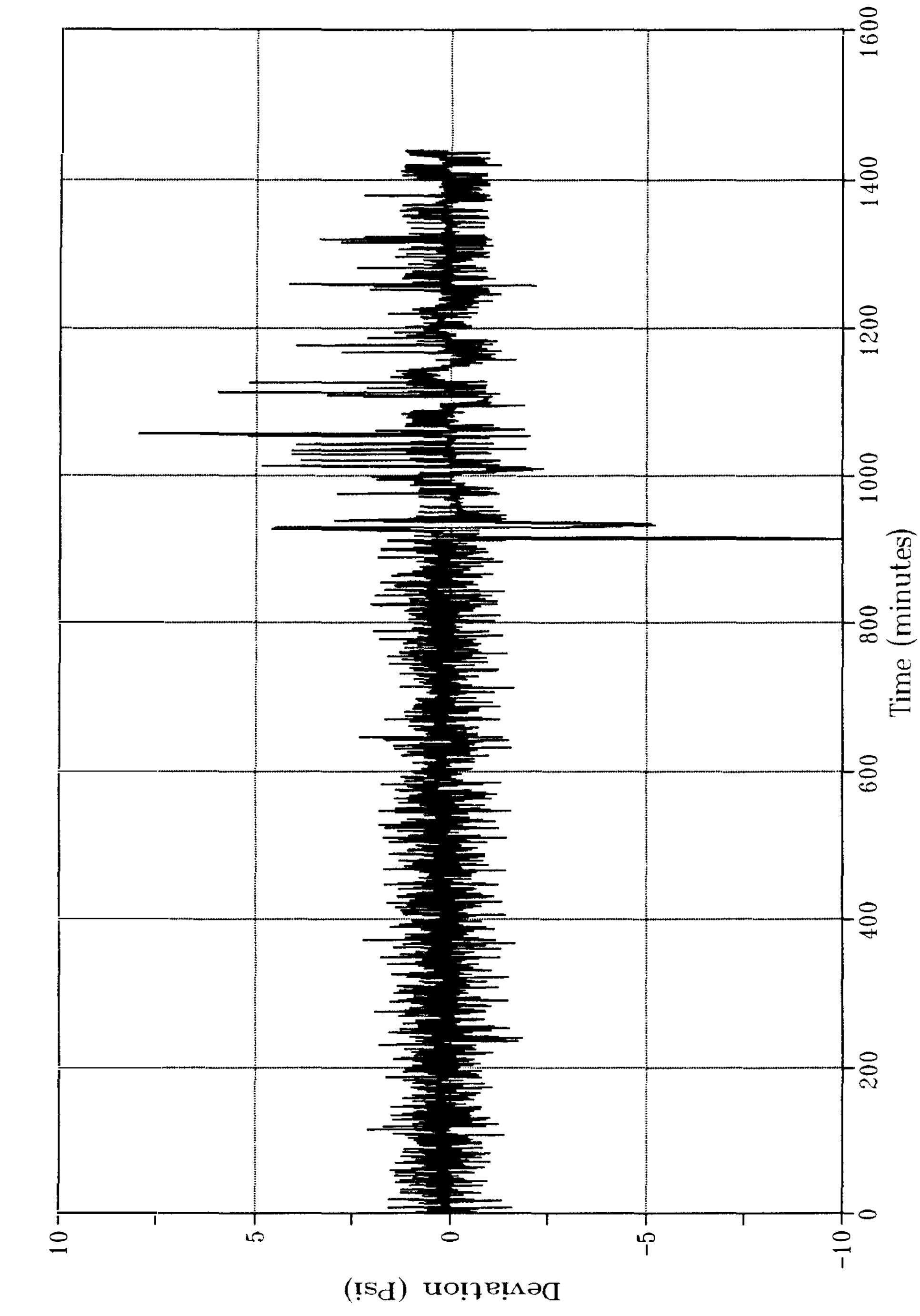

Figure 6.23: Error in PEM estimation shown in Figure 6.22.

through 6.23 (also see Tables 6.2 and 6.3).

A sensitivity analysis of PEM estimates can be performed easily since an analytical equation is available. Sensitivity analysis for the dynamic PEM indicates that the most important signal is the hot leg temperature for the steam generator wide range water level estimate, and the cold leg temperature for the steam generator pressure estimate.

The relative sensitivities for model #7 and model #8 were found as

$$\frac{\partial Level}{\partial x(1)} = 0.05 \tag{6.1}$$

$$\frac{\partial Level}{\partial x(2)} = 0.40 \tag{6.2}$$

$$\frac{\partial Level}{\partial x(6)} = 0.94 \tag{6.3}$$

$$\frac{\partial Level}{\partial x(7)} = -0.21 \tag{6.4}$$

$$\frac{\partial Pressure}{\partial x(1)} = 0.01 \tag{6.5}$$

$$\frac{\partial Pressure}{\partial x(6)} = -0.27 \tag{6.6}$$

$$\frac{\partial Pressure}{\partial x(2)} = 0.04 \tag{6.7}$$

$$\frac{\partial Pressure}{\partial x(5)} = 0.81 \tag{6.8}$$

$$\frac{\partial Pressure}{\partial x(7)} = 1.07 \tag{6.9}$$

The sensitivity results were obtained by using partial derivatives of the polynomial model with respect to the input variables, and then substituting the nominal values of the variables.

6.4 Artificial Neural Networks

The estimation of process variables using artificial neural networks (ANN) modeling was peformed for steam generator wide range water level and steam generator pressure signals. Data from PWR-1 and PWR-2 were used for the ANN estimates.

Table 6.4 shows the input variables used for static and dynamic ANN models used in this study. All ANN models were developed using the commercial software package NeuralWorks. The ANN models were constructed by using the fast back-propagation algorithm with a three-layer topology. Although initial research of this study also included an auto-associative ANN, hetero-associative ANN's were more successful in training speed and recall precision, so that this study focused only on hetero-associative ANN's having only one processing element (PE) in the output layer. The number of PE's in the hidden layer was twice the number of PE's in the input layer to memorize specific transient patterns. Training was stopped when the root-mean-square error (RMSE) was reduced to approximately 0.05 (this corresponds approximately to 20000 iterations for dynamic networks and 100000 iterations for static networks). The generated network was exported to C and incorperated in the PC-based signal validation program (Appendix C).

Figures 6.24 - 6.40 show the results obtained using ANN modeling. According to these results the best estimates were obtained using type 1 dynamic ANN's in which the value of the output variable at previous sampling time was used as an input to the ANN (Figures 6.28, 6.30, 6.37 and 6.39). However, in some cases having a static ANN was adequate for a good estimate (Figure 6.36).

6.5 Implementation of the Kalman Filtering Technique

The Kalman filtering technique (KFT) uses the UTSG model described in Chapter 4. The model consists of 19 state variables for the steam generator and 4 state variables for the controller, for a total of 23 state variables. The measurement vector includes

- steam generator wide range water level,
- steam generator pressure,
- steam generator main feedwater flow,
- steam generator steam flow,
- RCS flow,
- hot leg temperature, and
- cold leg temperature.

Table 6.4: Input variables used in various artificial neural network models.

Estimated Variable	**Steam Generator Main Feedwater Flow**	**RCS Flow**	**Steam Generator Steam Flow**	**Hot Leg Temperature**	**Cold Leg Temperature**	**Steam Generator Water Level**	**Steam Generator Pressure**
Steam Generator Water Level or Pressure for Static Modeling (Figure 6.24 - 6.27, 6.34 - 6.36)	t	t	t	t	t	N/A	N/A
Steam Generator Water Level for Type 1 Dynamic Modeling (Figure 6.28, 6.29, 6.37, 6.38)	t	t	t	t	t	t-1	N/A
Steam Generator Pressure for Type 1 Dynamic Modeling (Figure 6.30, 6.31, 6.39, 6.40)	t	t	t	t	t	N/A	t-1
Steam Generator Water Level or Pressure for Type 2 Dynamic Modeling (Figure 6.32, 6.33)	t, t-1	t, t-1	t, t-1	t, t-1	t, t-1	N/A	N/A

The UTSG model equations were discretized as follows:

The inlet plenum temperature is modeled as

$$\frac{dT_{pi}}{dt} = \frac{W_{pi}}{M_{pi}}\left(\theta_i - T_{pi}\right) \tag{6.10}$$

where the variables are defined in Table 4.2. Using the forward difference technique, the differential can be approximated as

$$\frac{T_{pi}(t+1) - T_{pi}(t)}{\Delta t} = \frac{W_{pi}}{M_{pi}}\left(\theta_i - T_{pi}(t)\right) \tag{6.11}$$

where Δt denotes the sampling time. Equation (6.11) simplifies to

$$T_{pi}(t+1) = \frac{W_{pi}}{M_{pi}}\left(\theta_i - T_{pi}(t)\right)\Delta t + T_{pi}(t) \tag{6.12}$$

which has the same form given in Equation (3.13). This final form is used in the KFT calculations.

Figures 6.41 - 6.46 and Figures 6.49 - 52 show results obtained using the extended KFT for PWR-1 and PWR-2. The results in these figures indicate that the estimations of the KFT module are quite close to the actual *good* measurements. The use of measurements provides high accuracy in estimating these variables. This is also one of the main reasons why the KFT module gives better estimations than other signal validation modules.

Defining the Kalman filtering correction as

$$\begin{aligned} \textit{Kalman filtering correction} &= \textit{Kalman gain} \times \textit{innovation sequence} \\ &= G(t) \times e(t) \end{aligned} \tag{6.13}$$

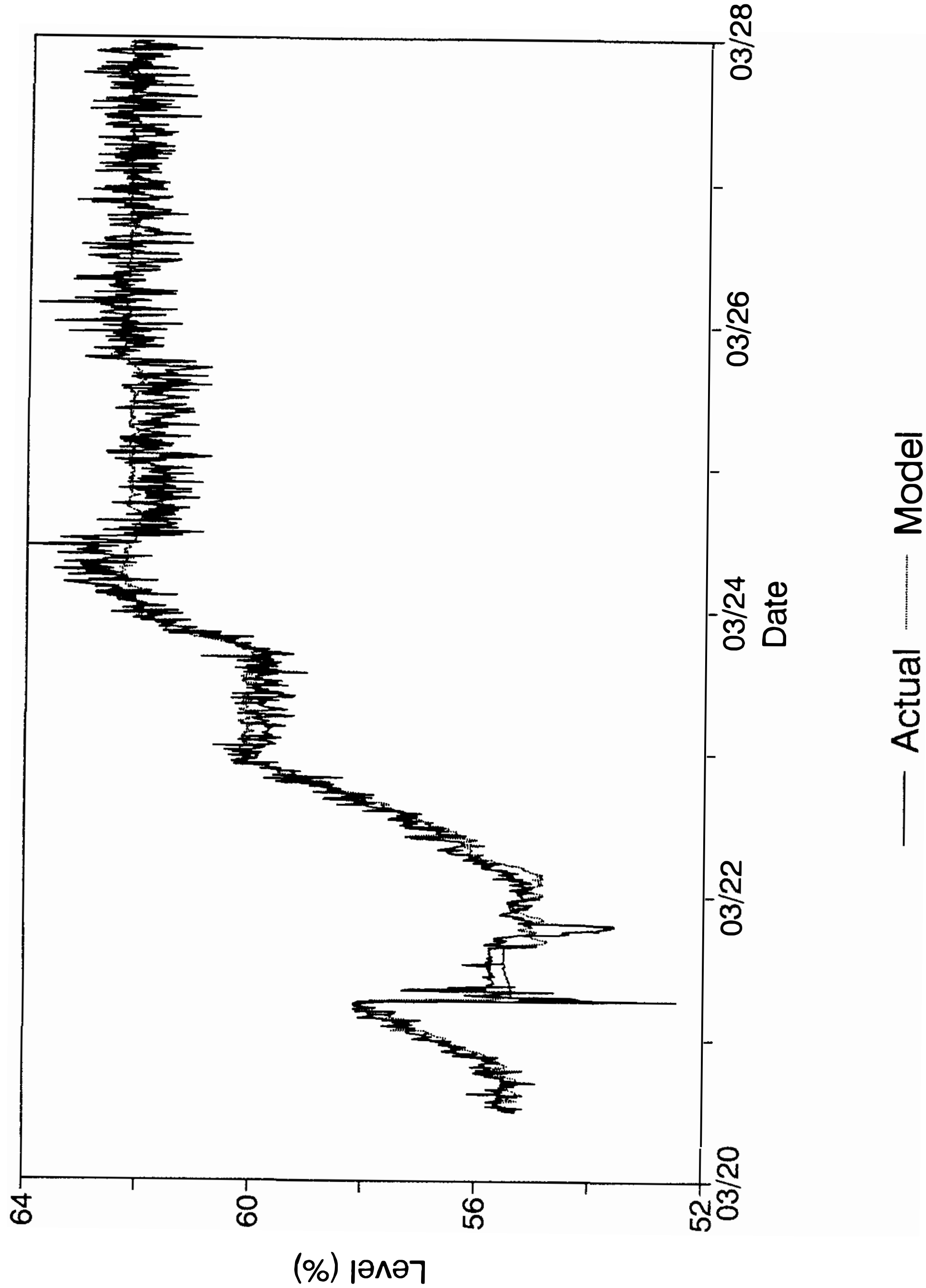

Figure 6.24: ANN estimate of steam generator wide range water level for PWR-1 using static modeling.

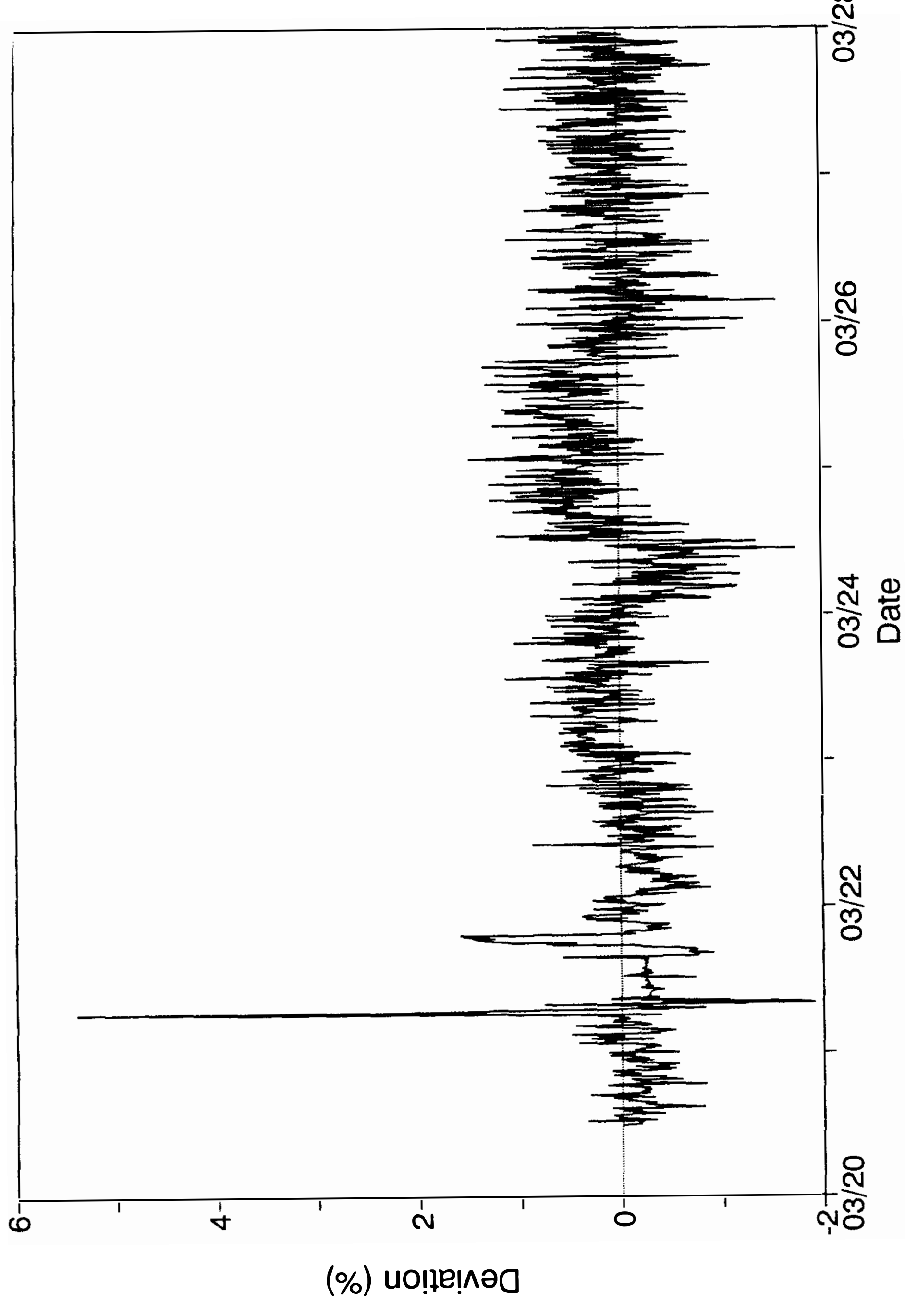

Figure 6.25: Error in ANN estimation shown in Figure 6.24.

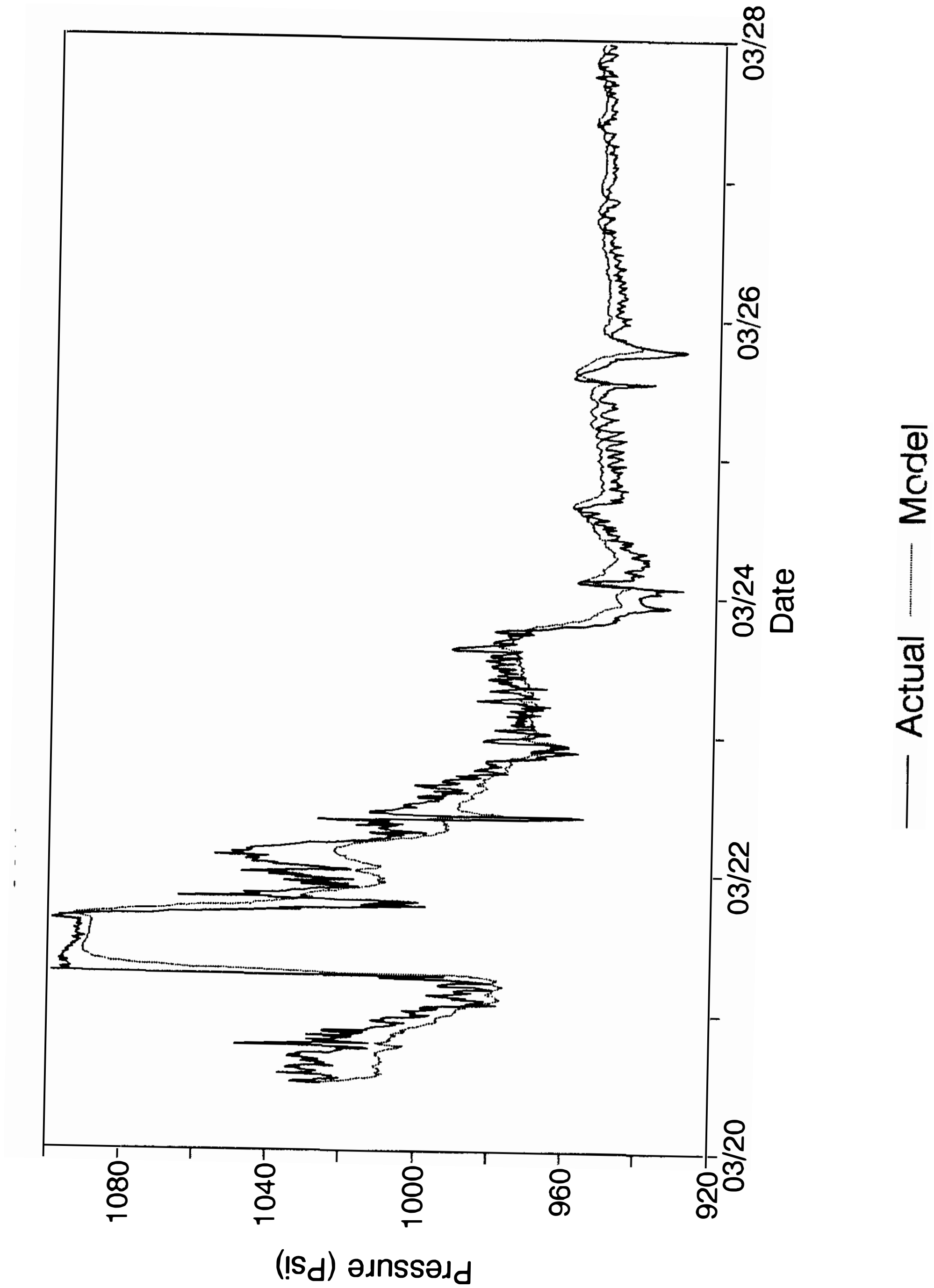

Figure 6.26: ANN estimate of steam generator pressure for PWR-1 using static modeling.

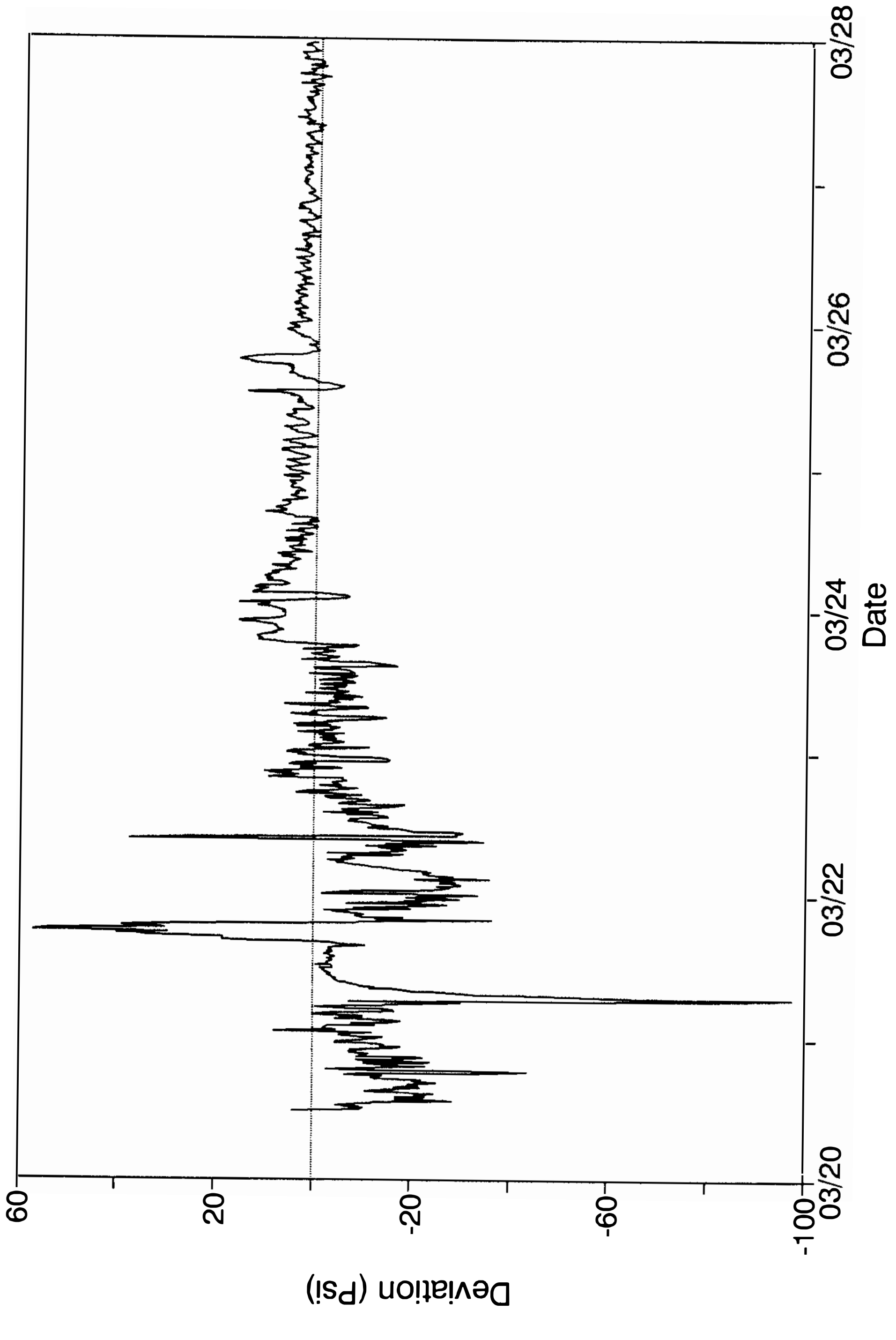

Figure 6.27: Error in ANN estimation shown in Figure 6.26.

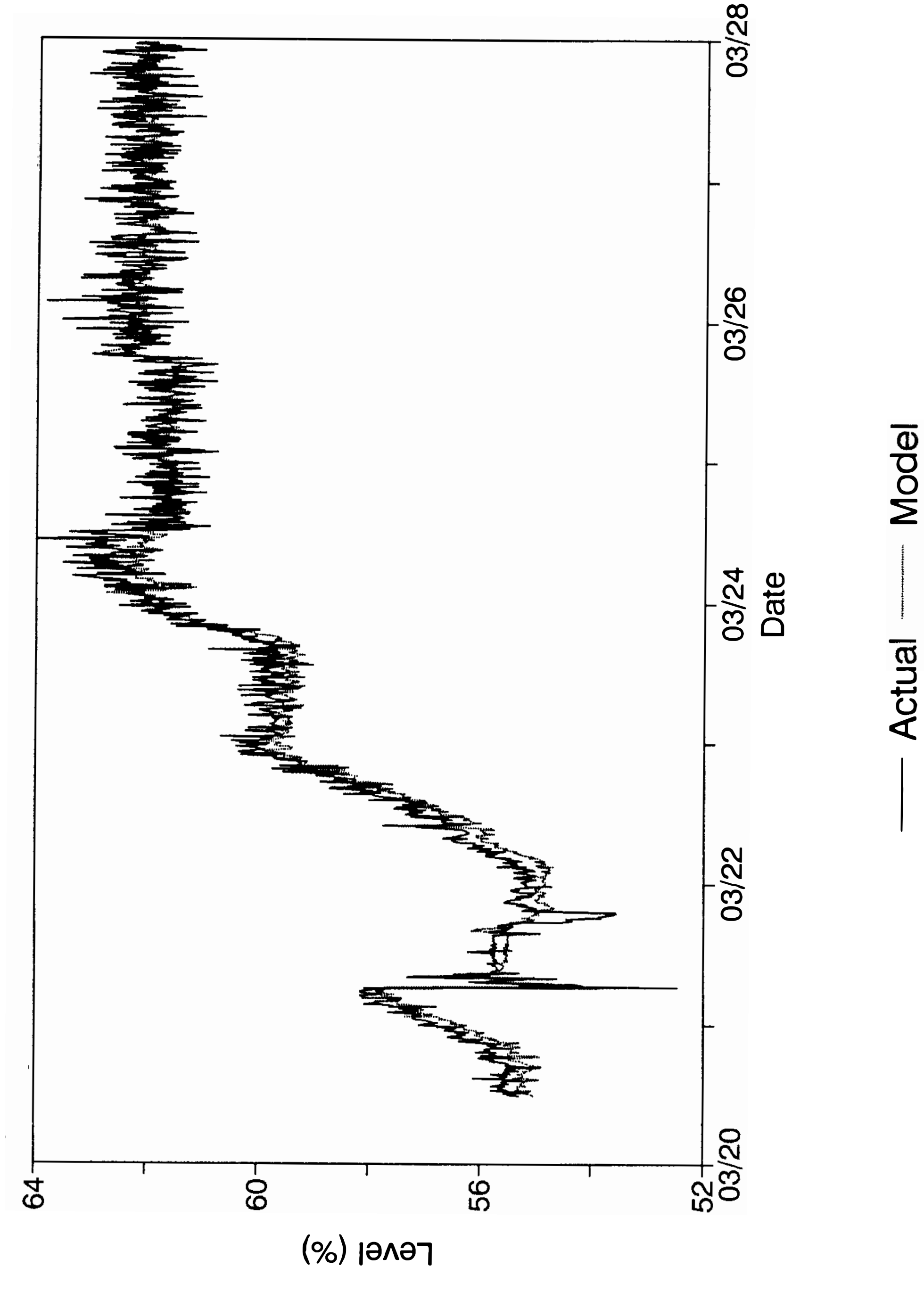

Figure 6.28: ANN estimate of steam generator wide range water level for PWR-1 using type 1 dynamic modeling.

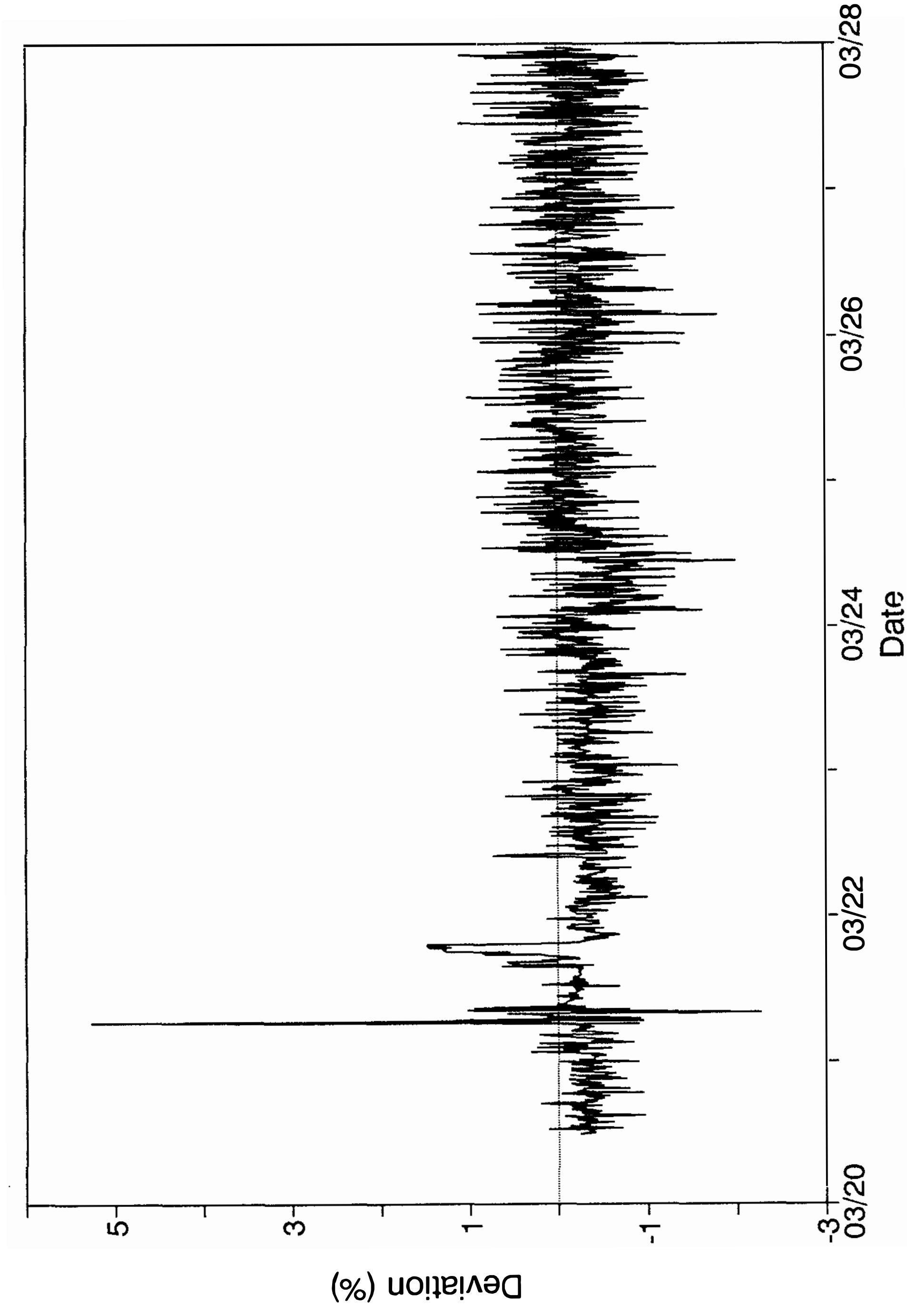

Figure 6.29: Error in ANN estimation shown in Figure 6.28.

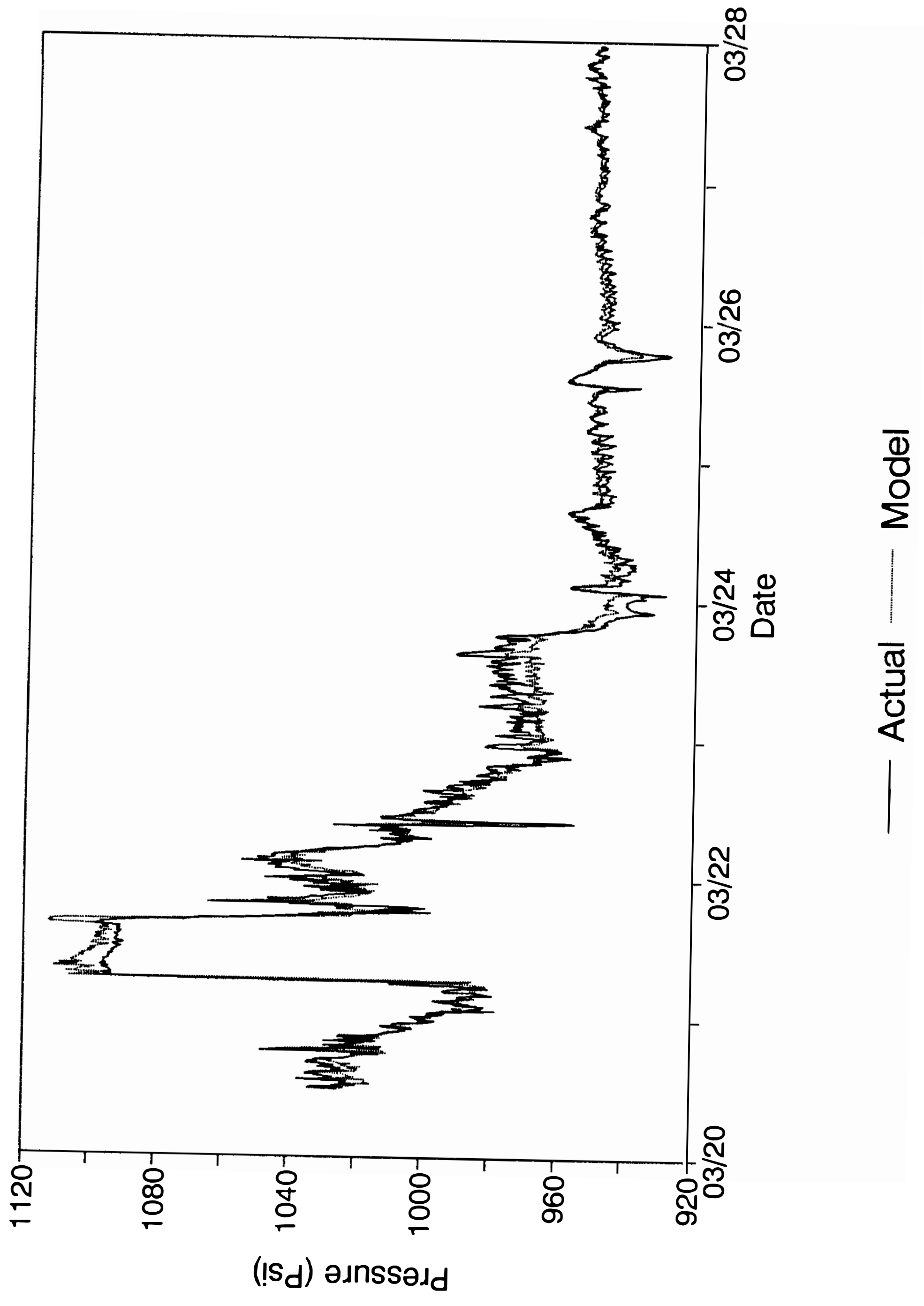

Figure 6.30: ANN estimate of steam generator pressure for PWR-1 using type 1 dynamic modeling.

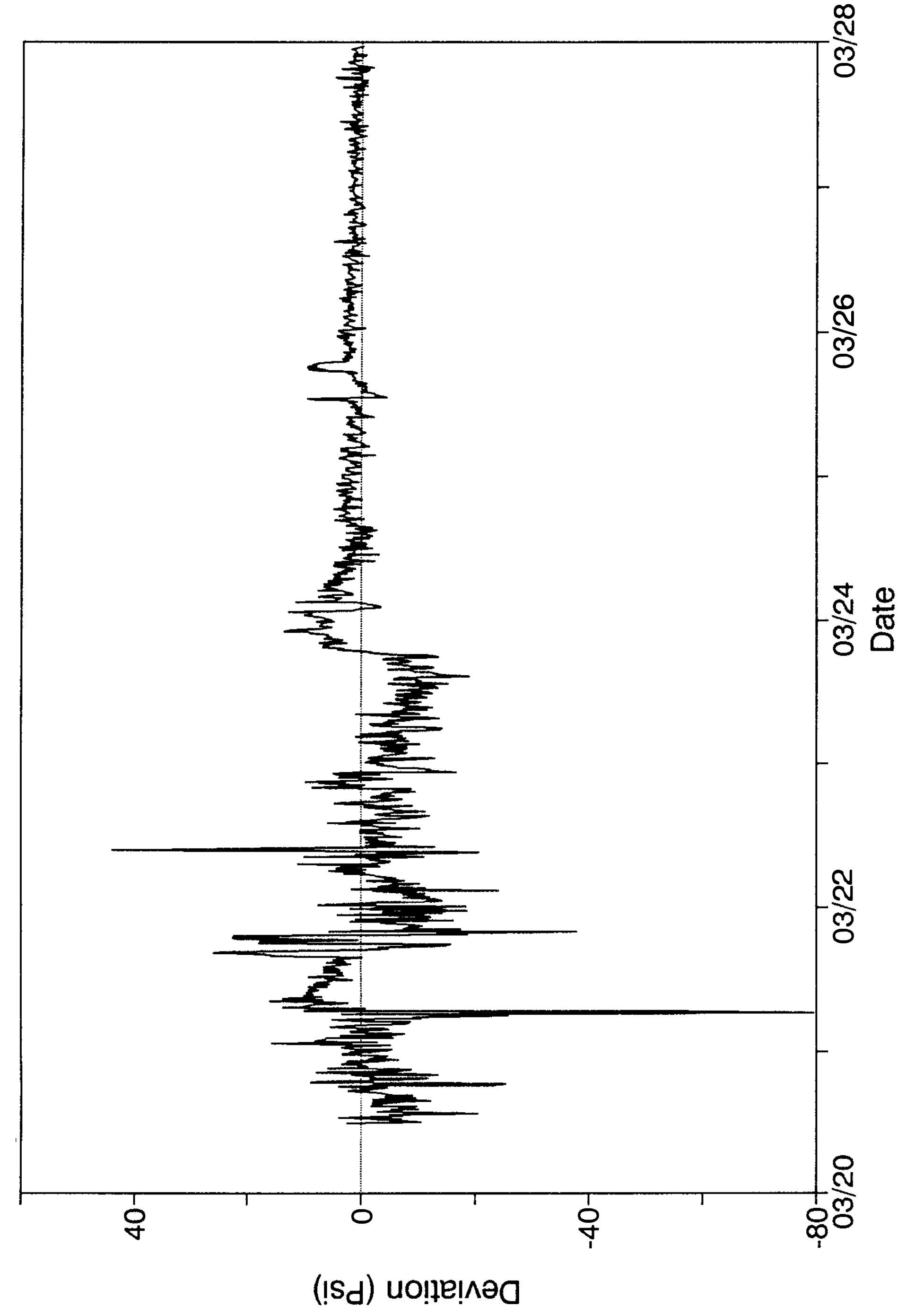

Figure 6.31: Error in ANN estimation as shown in Figure 6.30.

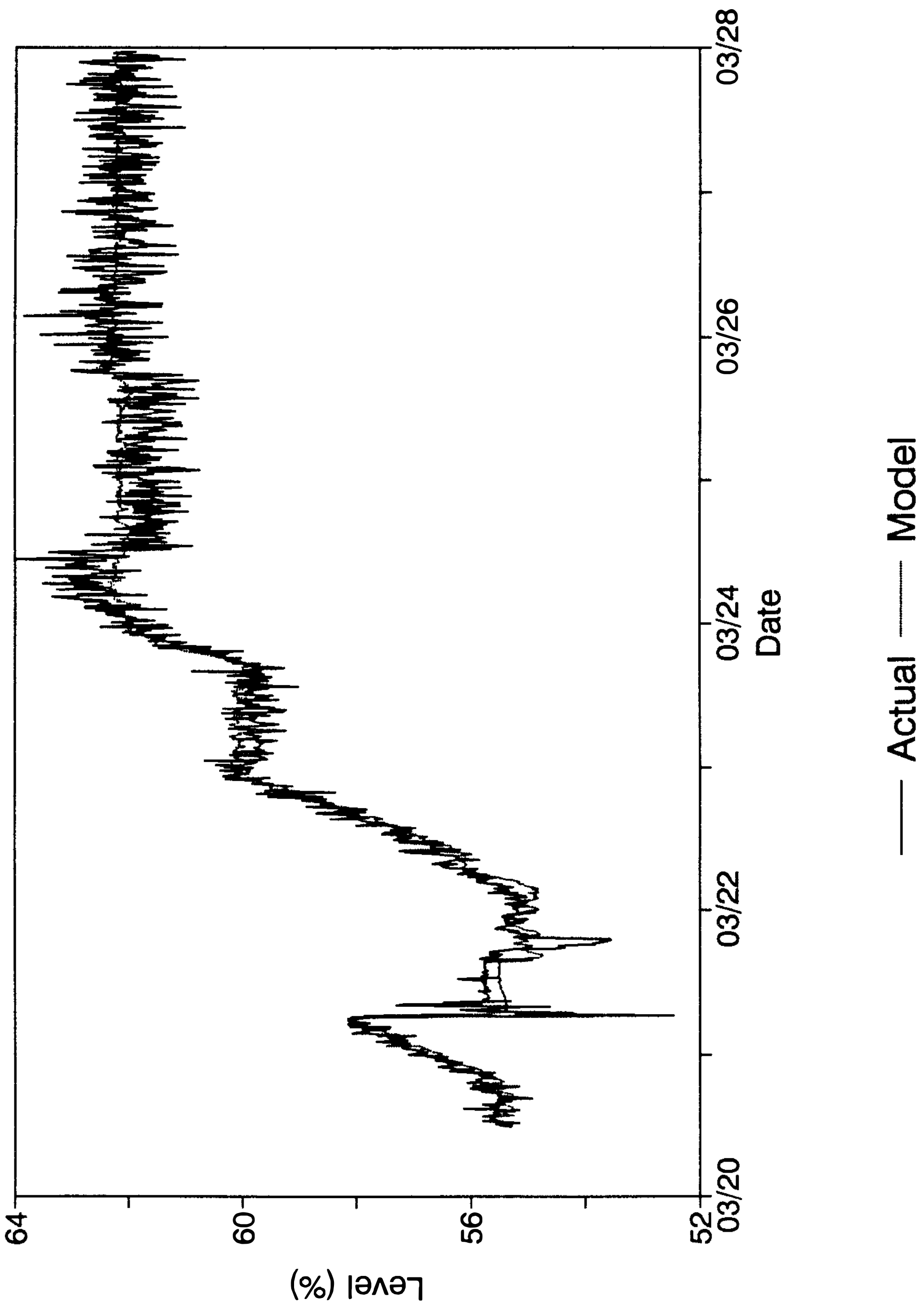

Figure 6.32: ANN estimate of steam generator wide range water level for PWR-1 using type 2 dynamic modeling.

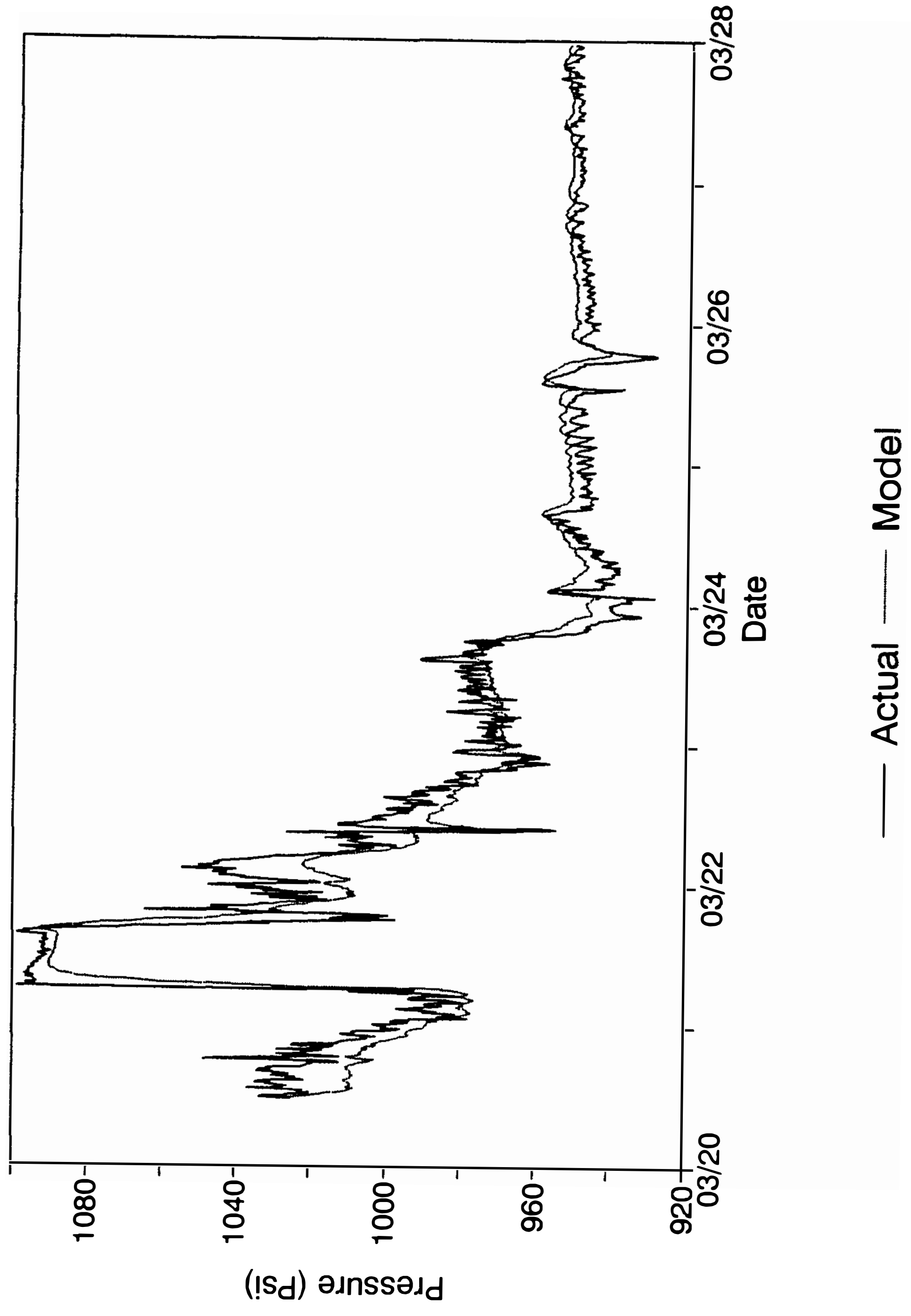

Figure 6.33: ANN estimate of steam generator pressure for PWR-1 using type 2 dynamic modeling.

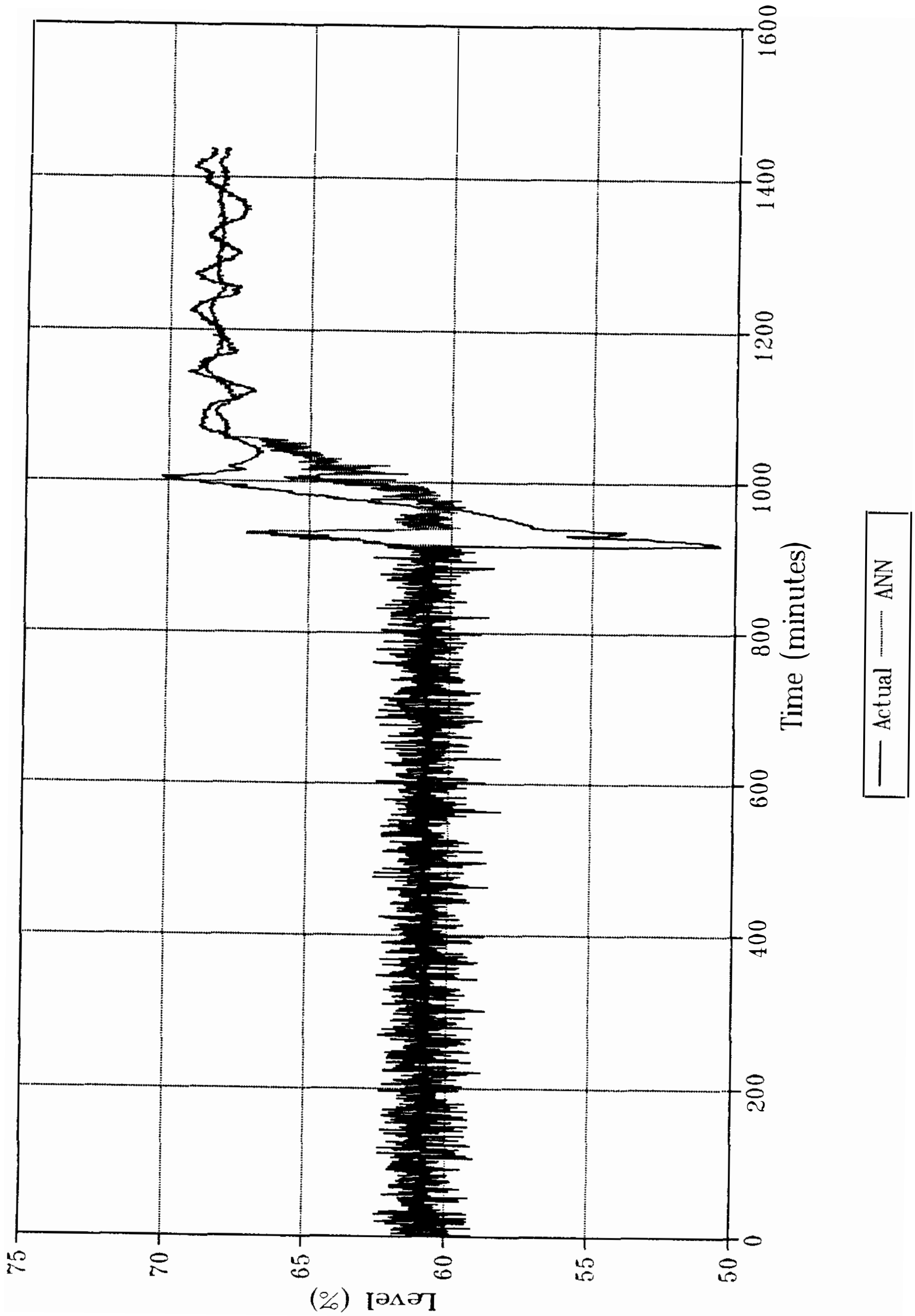

Figure 6.34: ANN estimate of steam generator wide range water level for PWR-2 using static modeling for steady-state and semi-transient operating conditions.

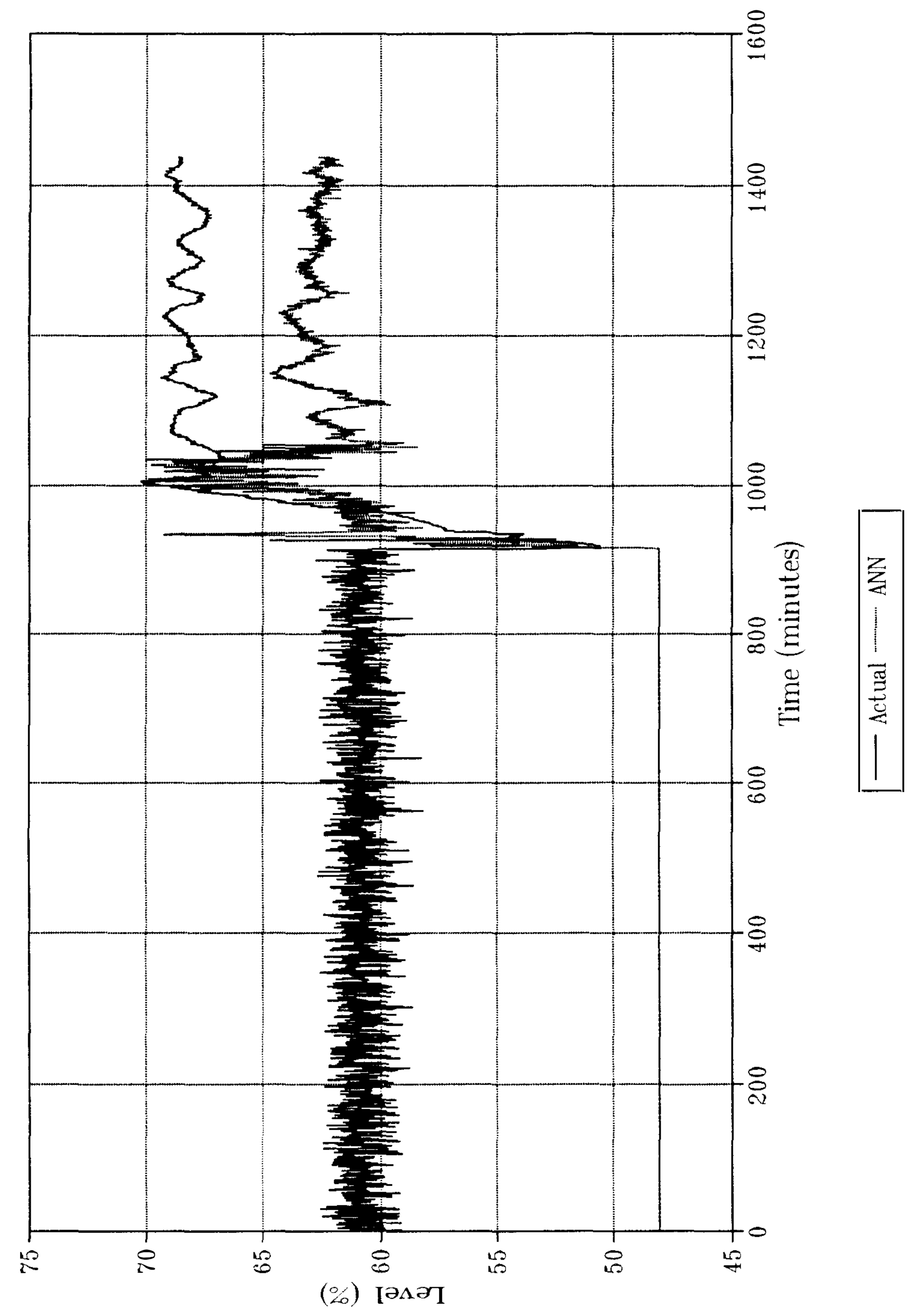

Figure 6.35: ANN estimate of steam generator wide range water level for PWR-2 using static modeling for transient operating conditions.

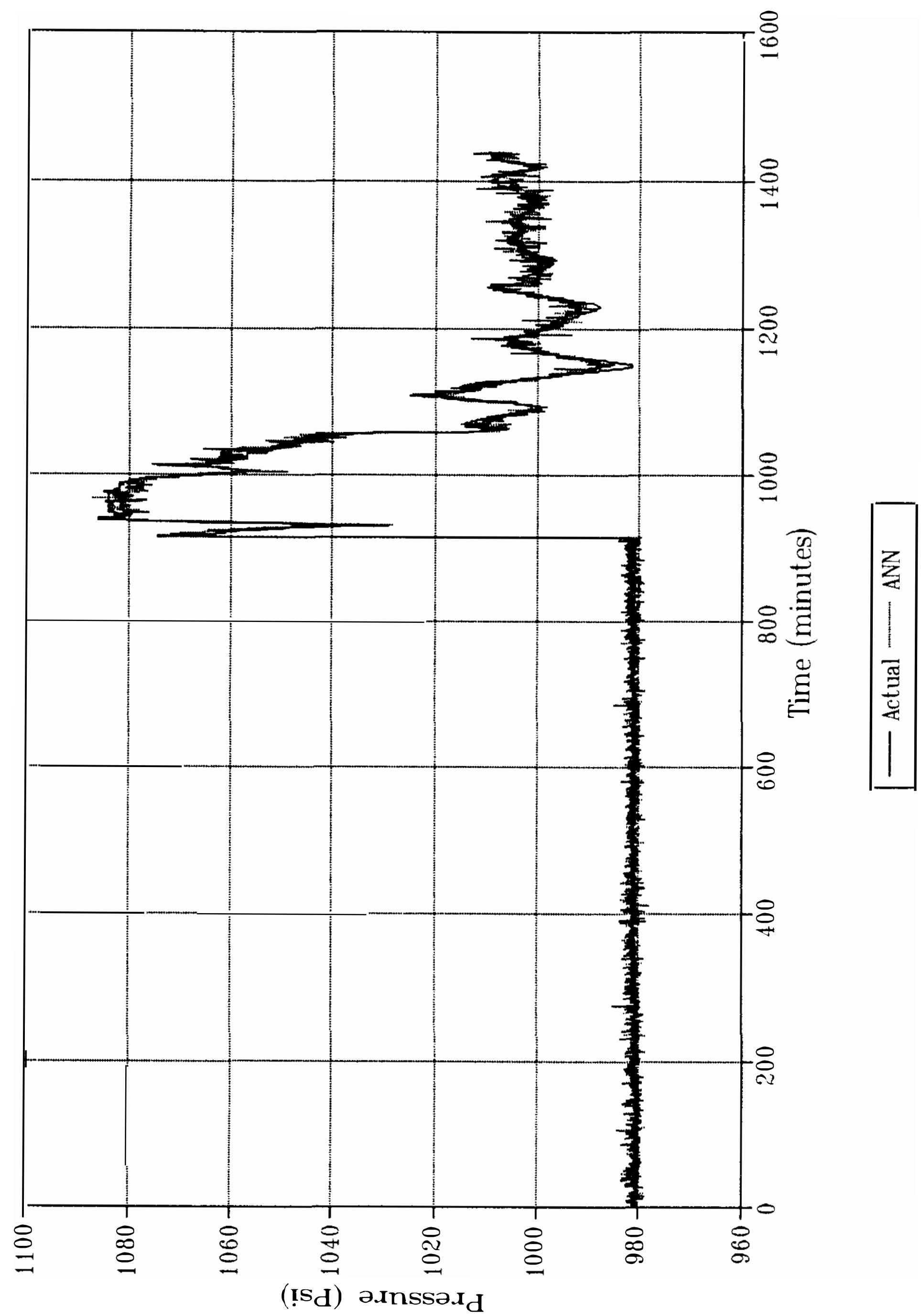

Figure 6.36: ANN estimate of steam generator wide pressure for PWR-2 using static modeling.

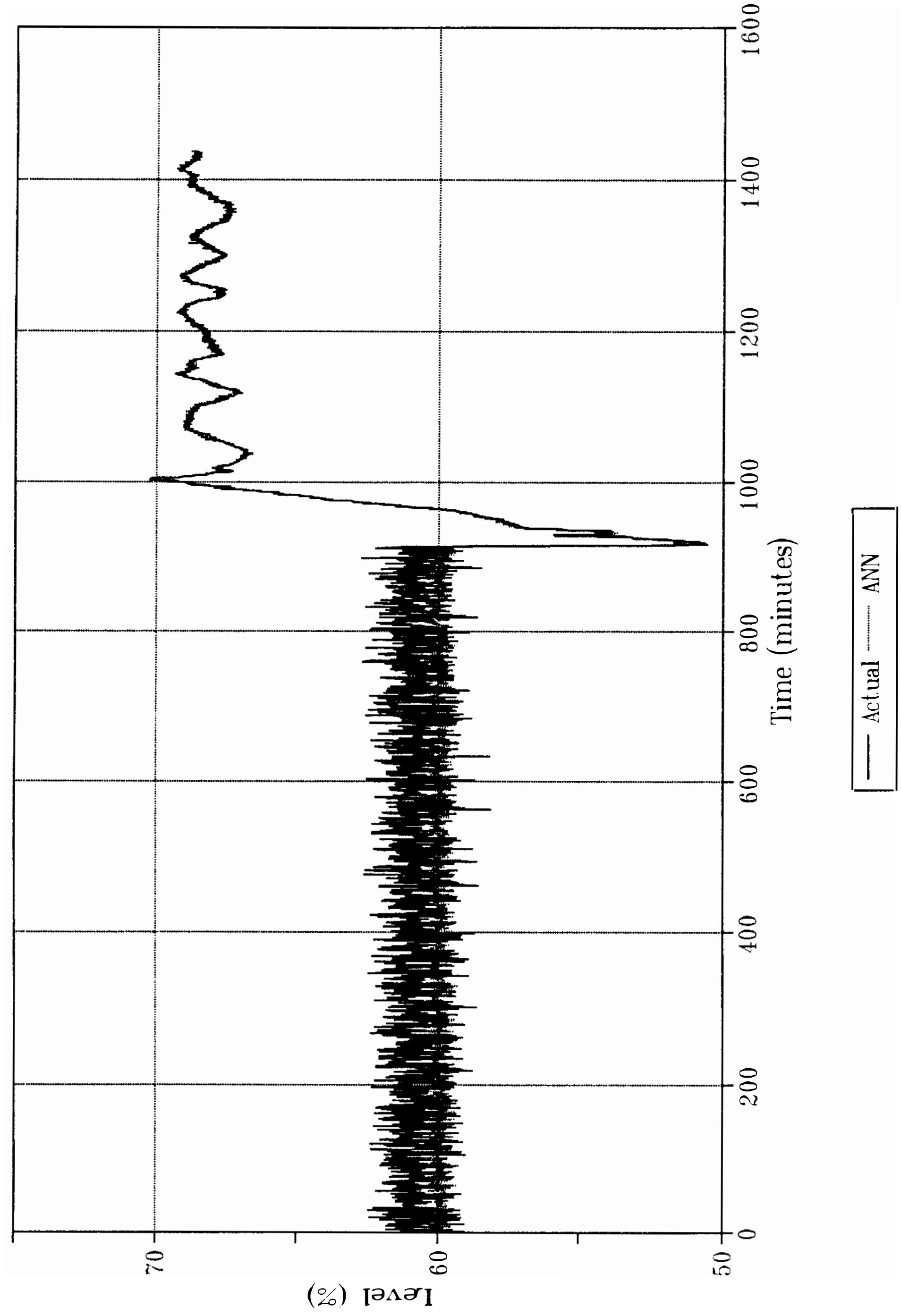

Figure 6.37: ANN estimate of steam generator wide range water level for PWR-2 using type 1 dynamic modeling.

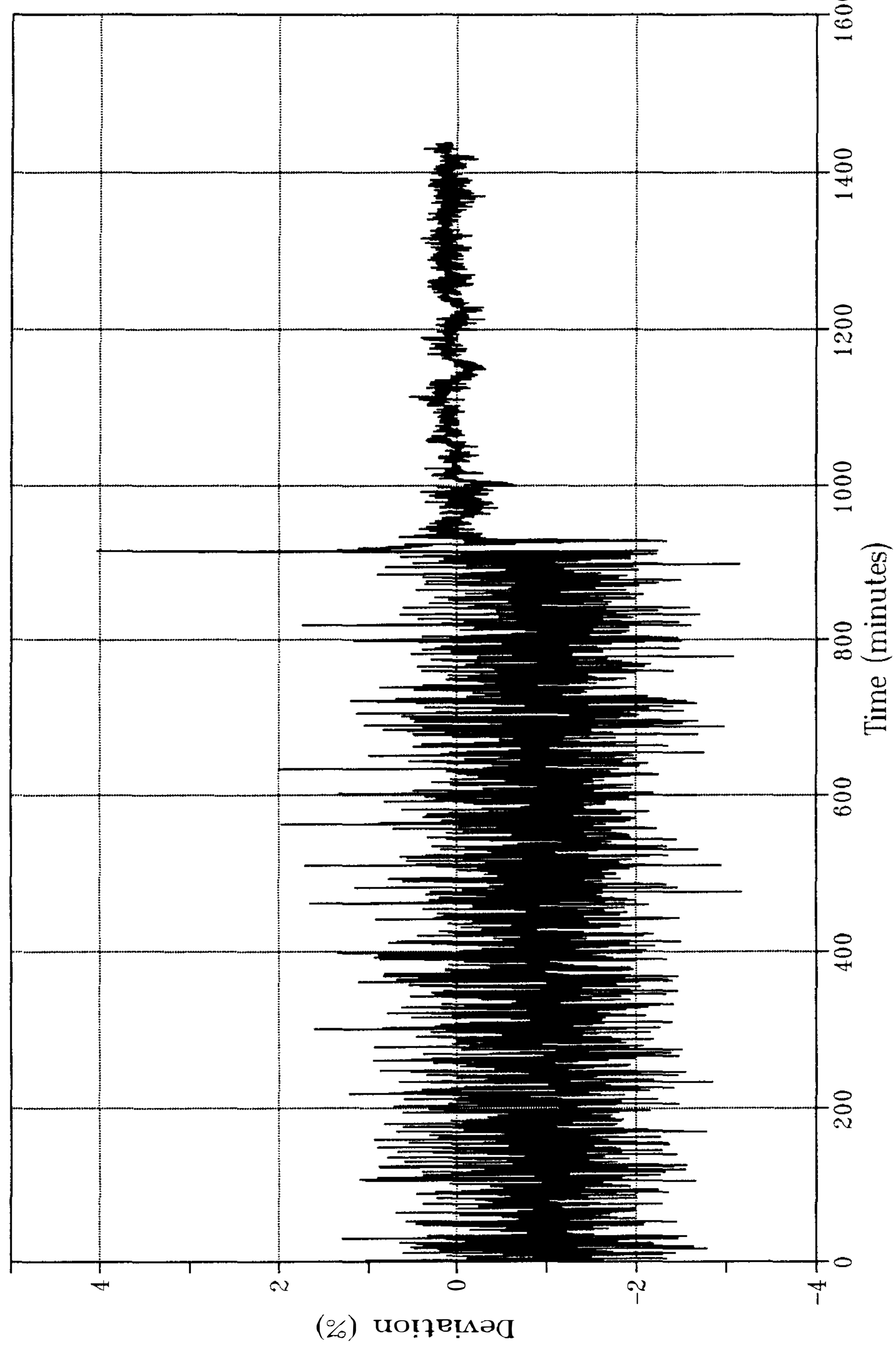

Figure 6.38: Error in ANN estimation as shown in Figure 6.37.

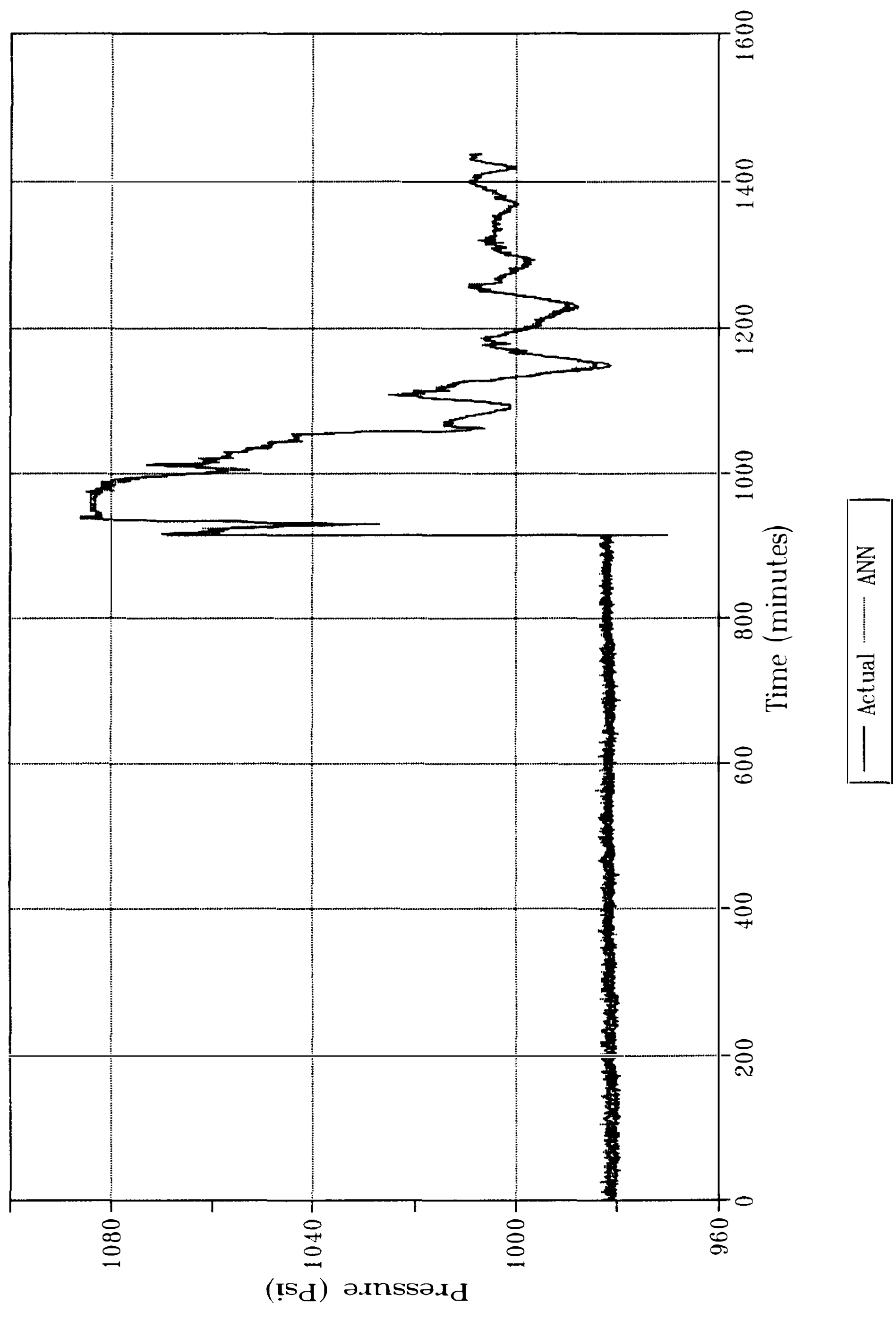

Figure 6.39: ANN estimate of steam generator pressure for PWR-2 using type 1 dynamic modeling.

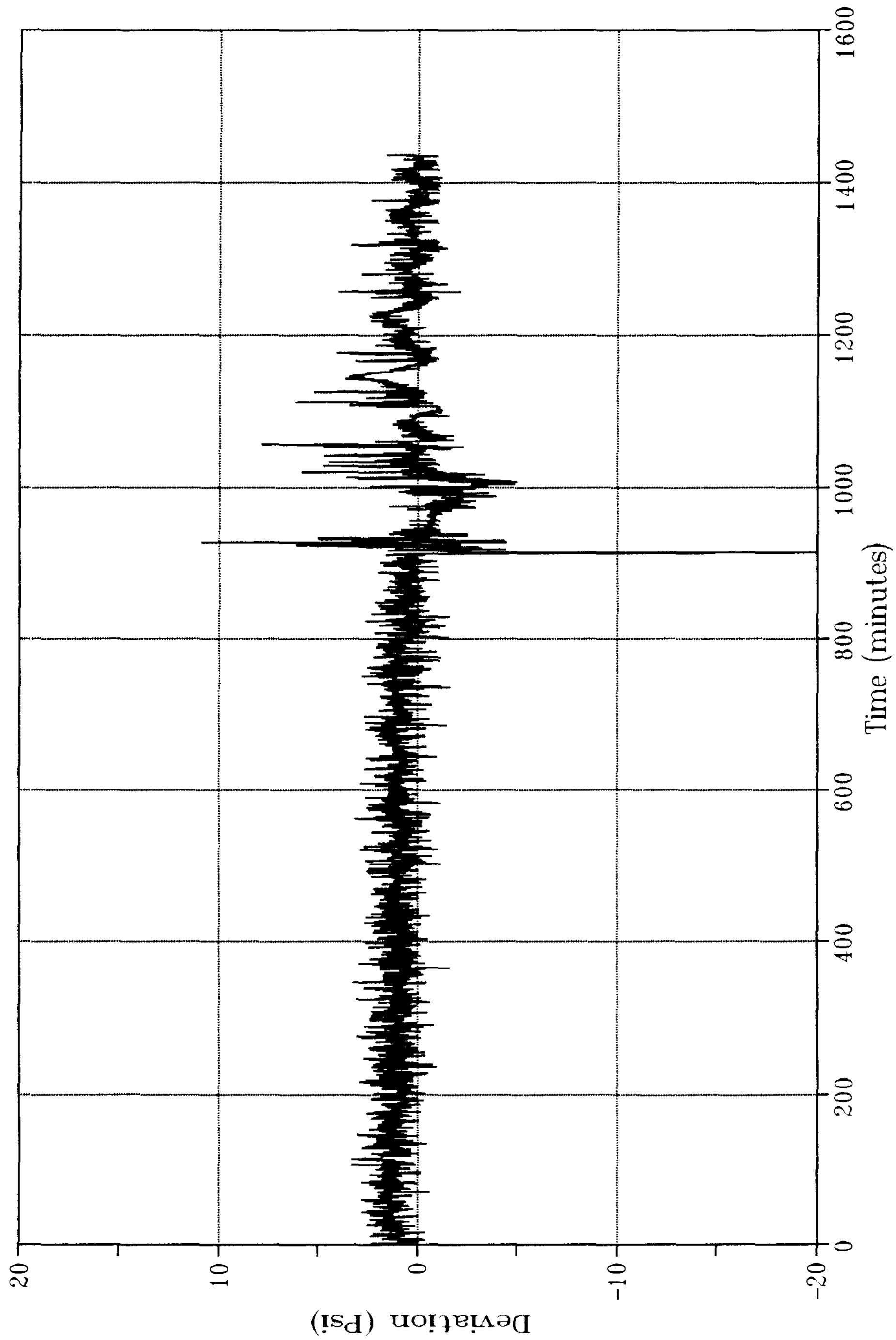

Figure 6.40: Error in ANN estimation as shown in Figure 6.39.

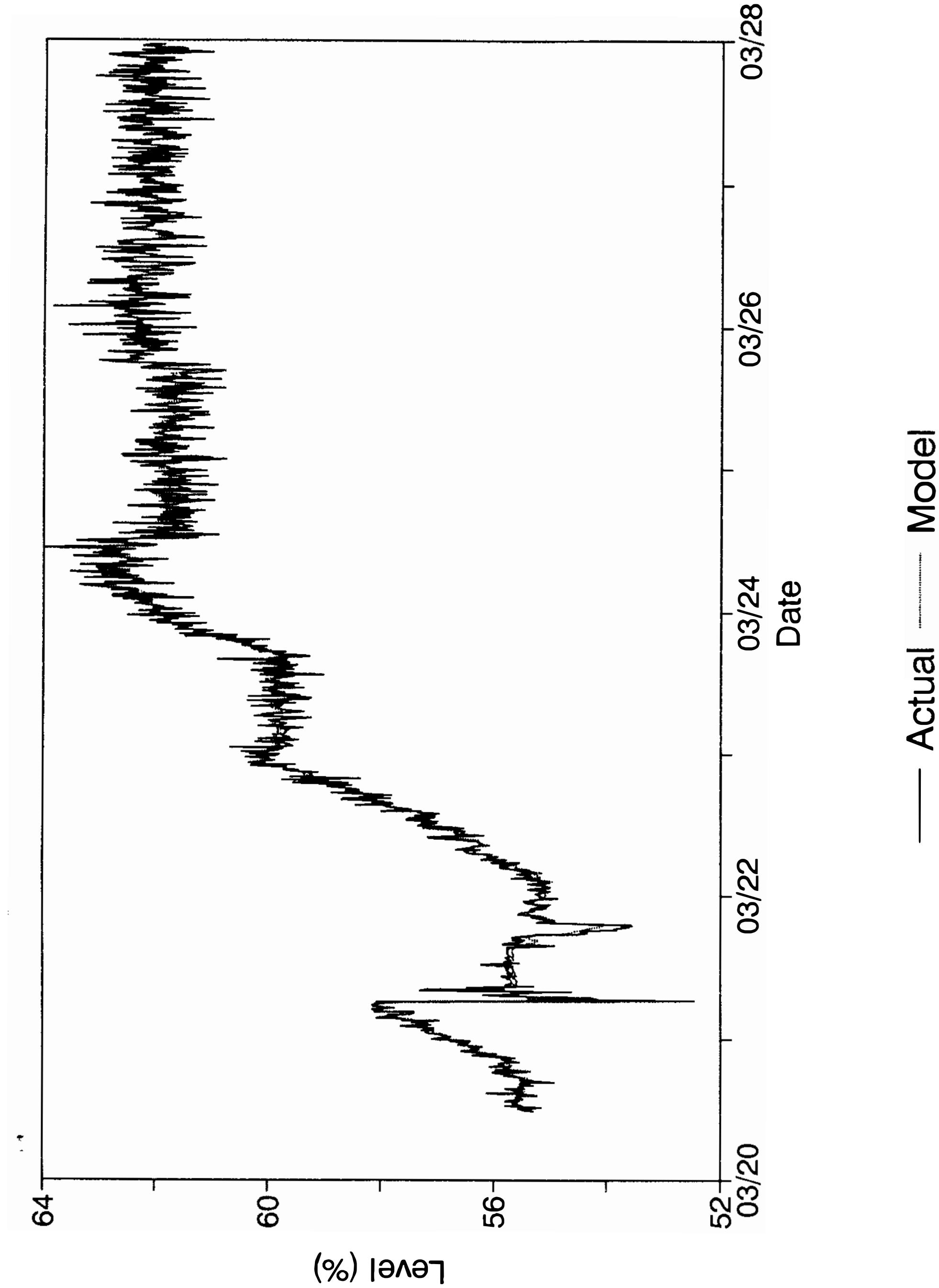

Figure 6.41: KFT estimation of steam generator wide range water level for PWR-1 with level and pressure measurements included.

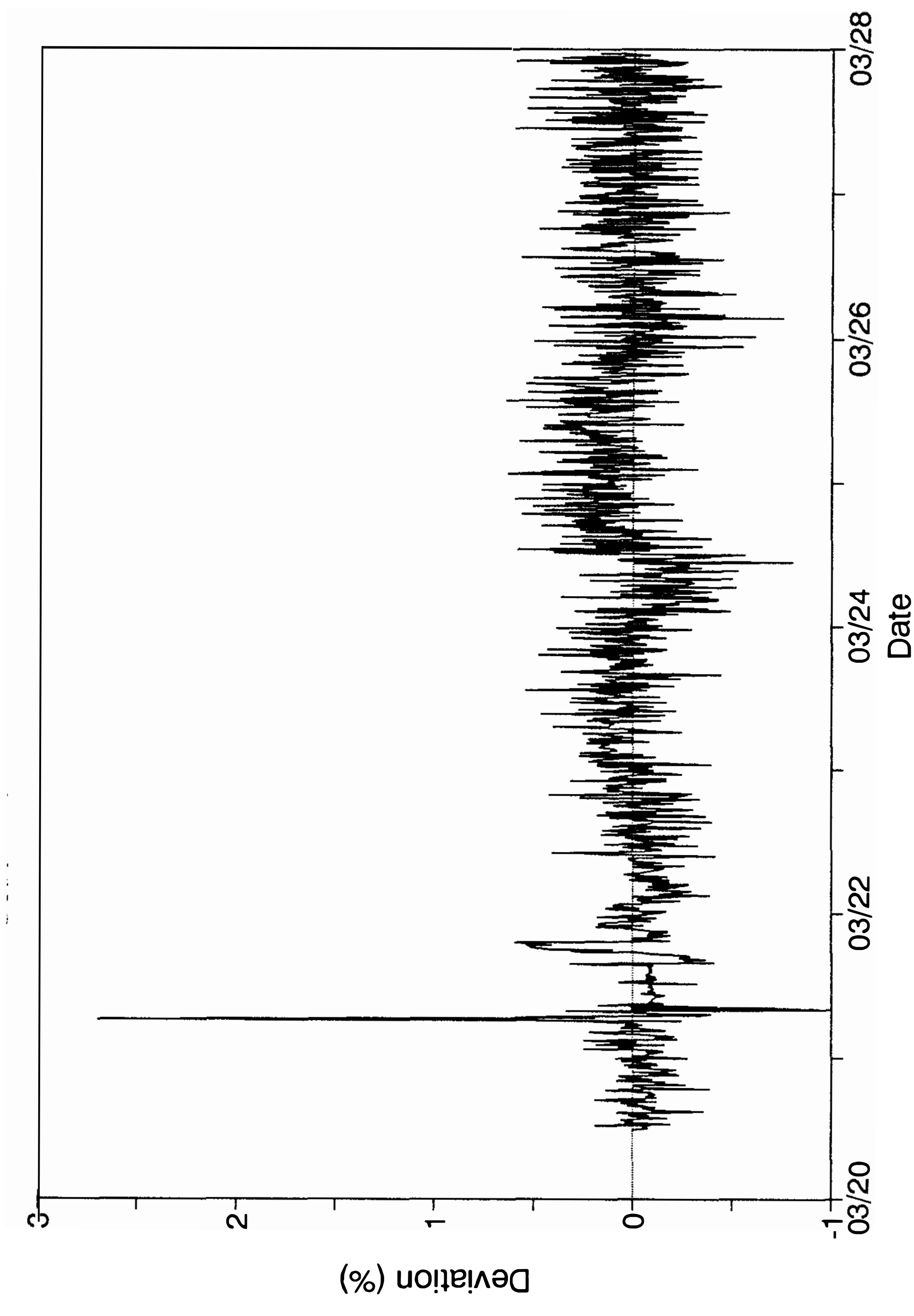

Figure 6.42: Error in KFT estimation shown in Figure 6.41.

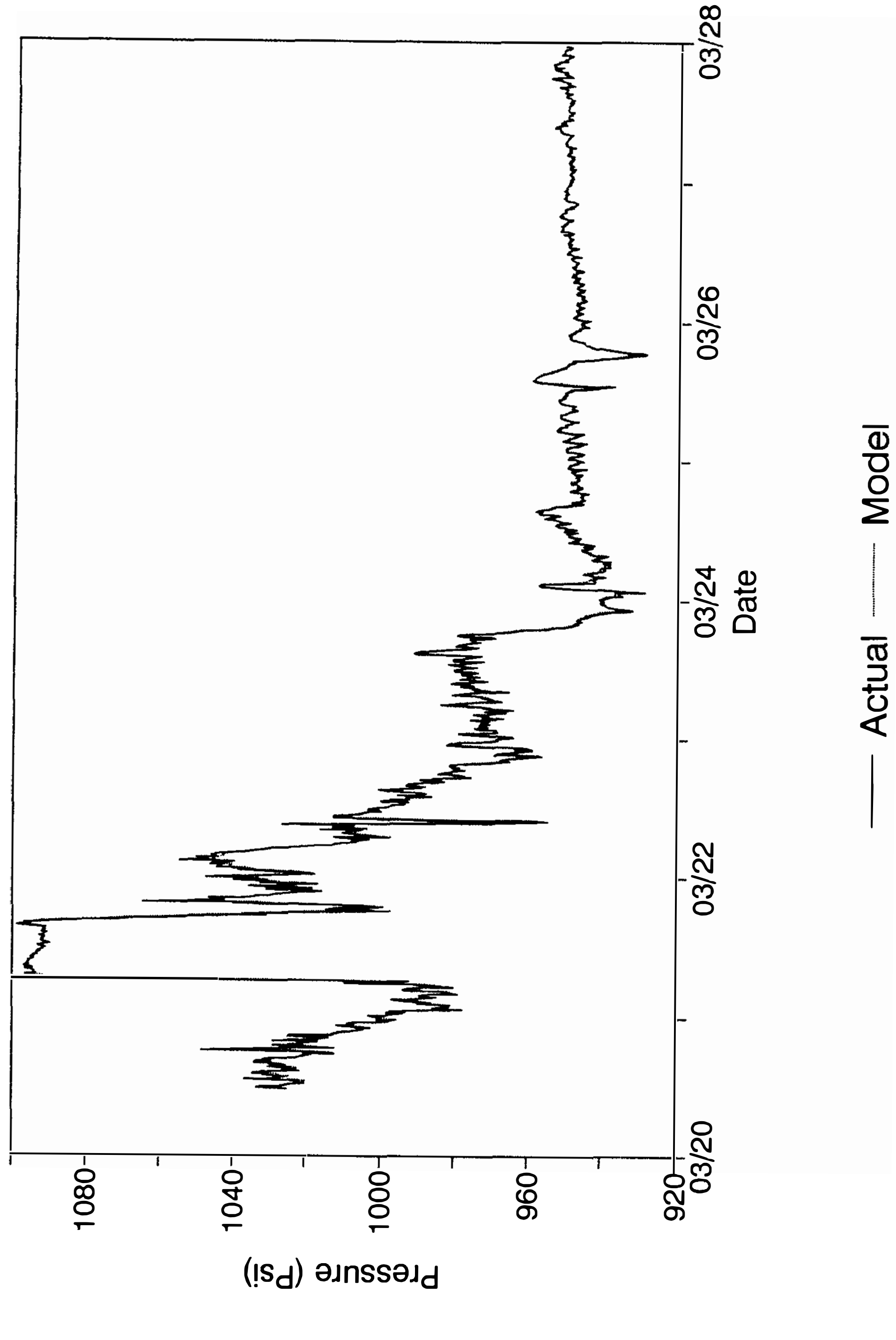

Figure 6.43: KFT estimation of steam generator pressure for PWR-1 with level and pressure measurements included.

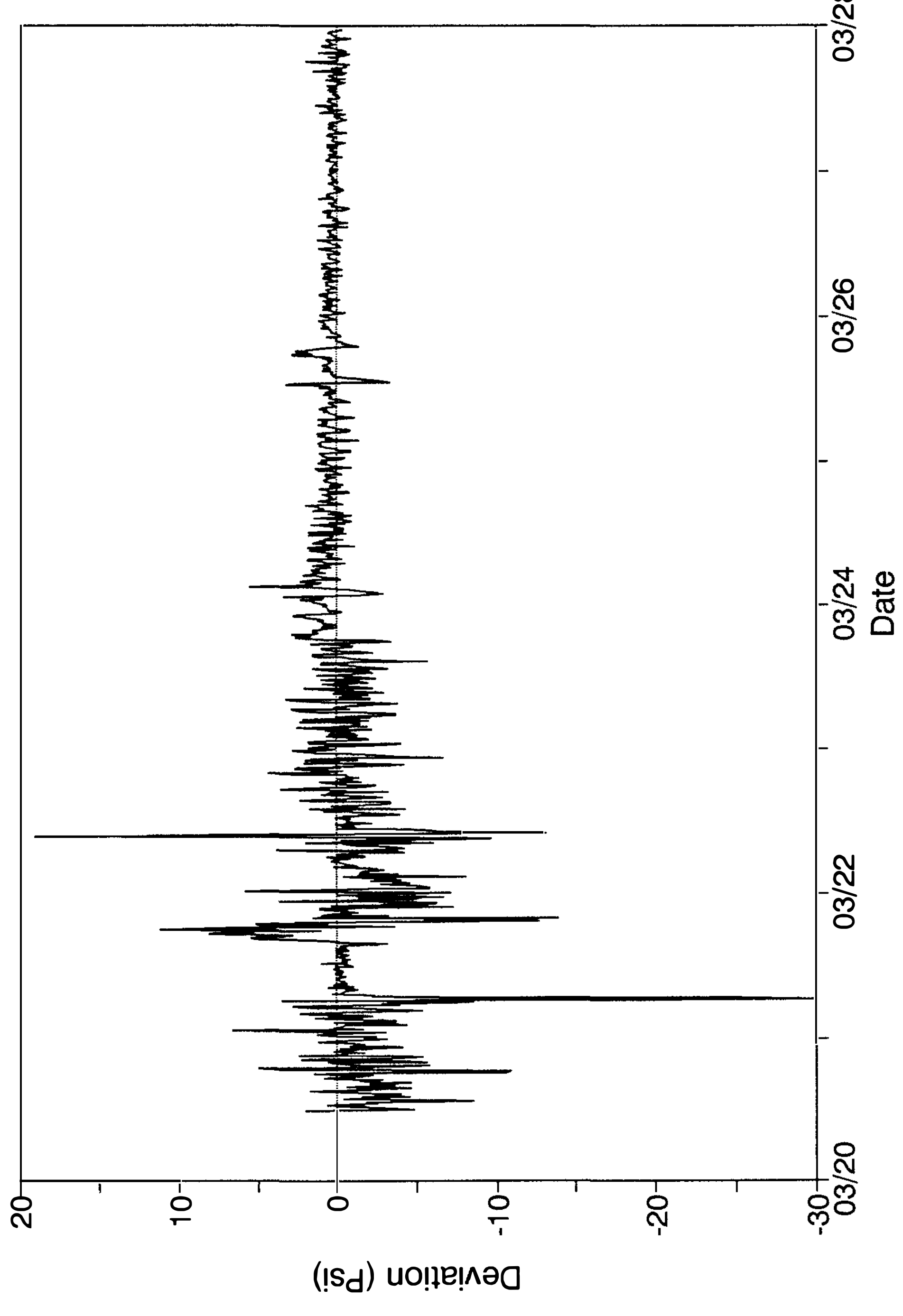

Figure 6.44: Error in KFT estimation shown in Figure 6.43.

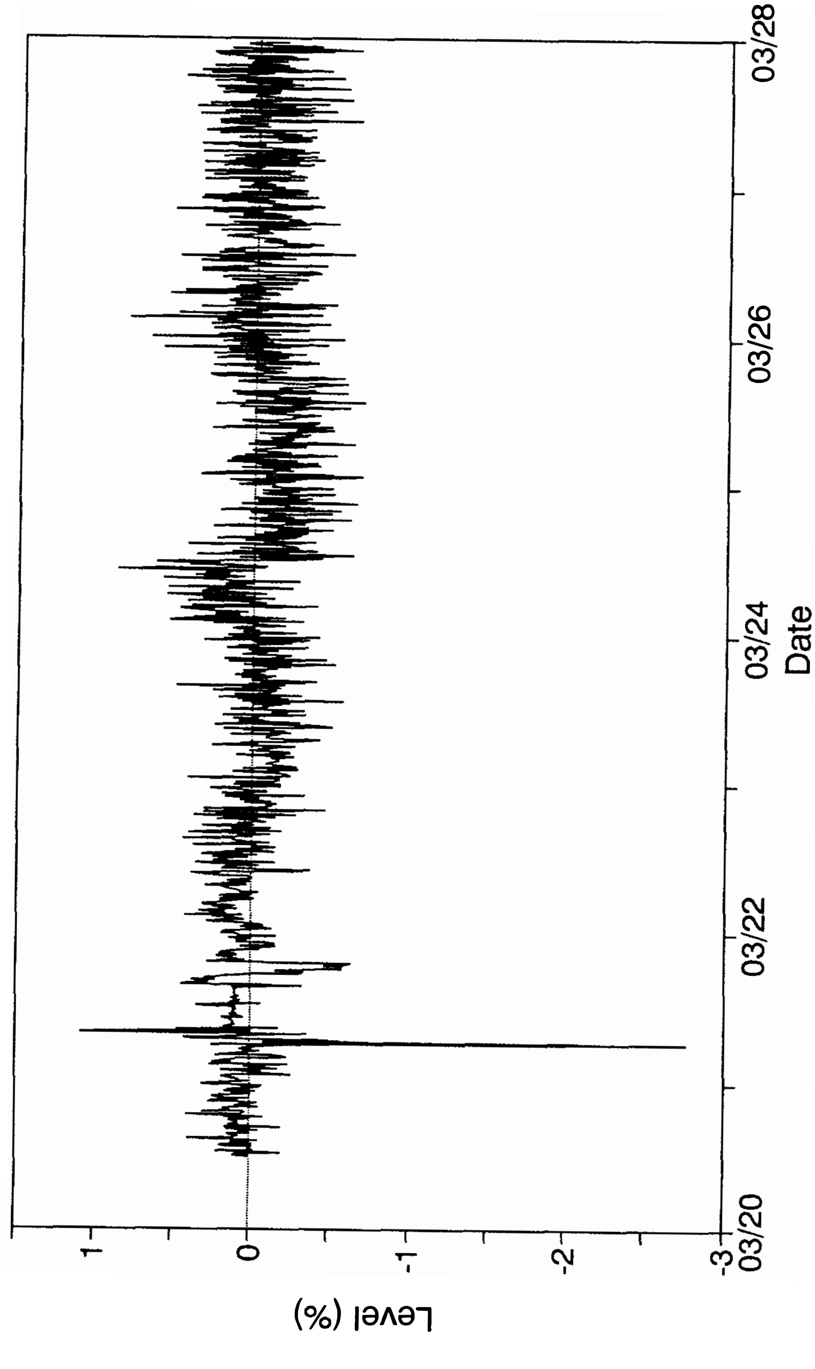

Figure 6.45: Kalman filtering correction to the estimate given in Figure 6.41.

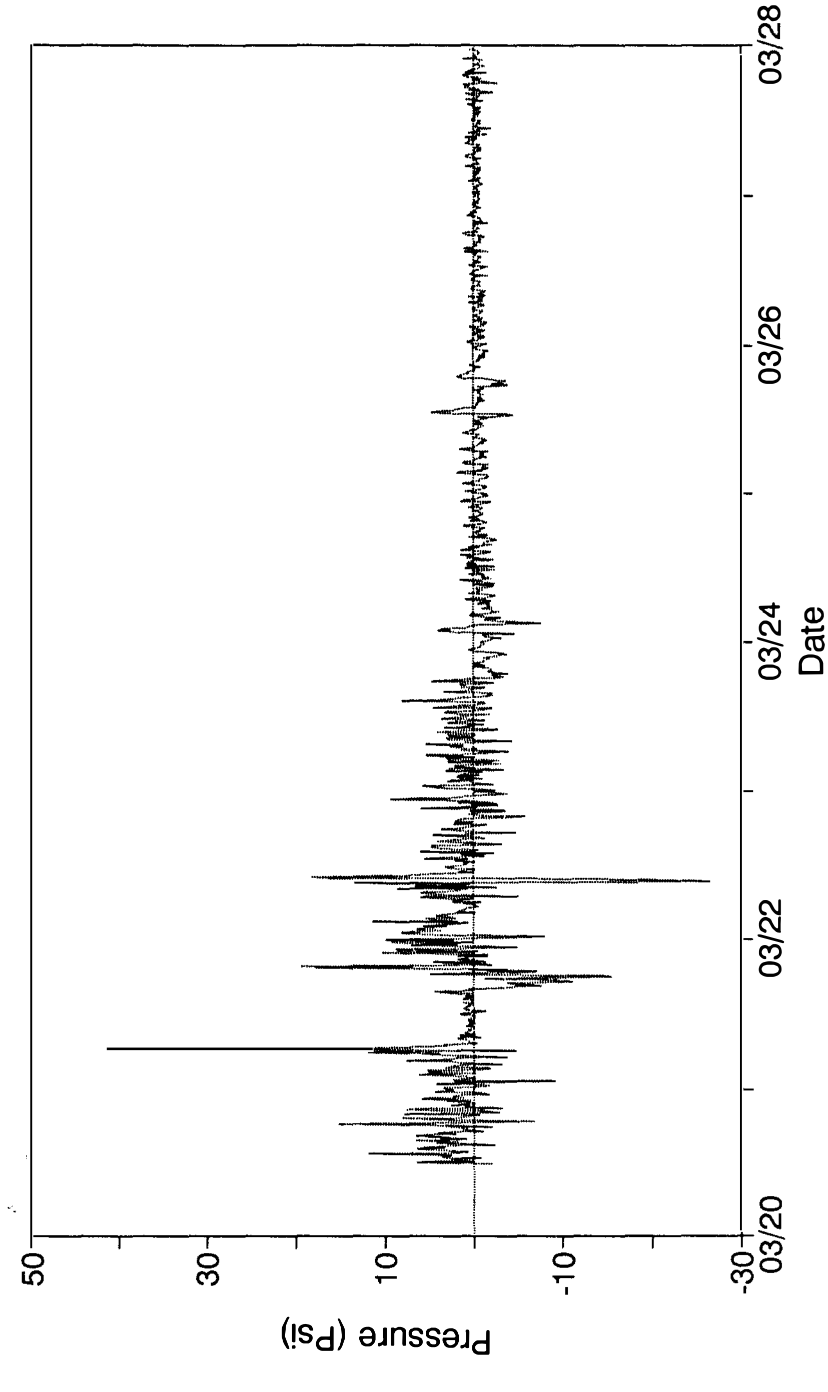

Figure 6.46: Kalman filtering correction to the estimate given in Figure 6.43.

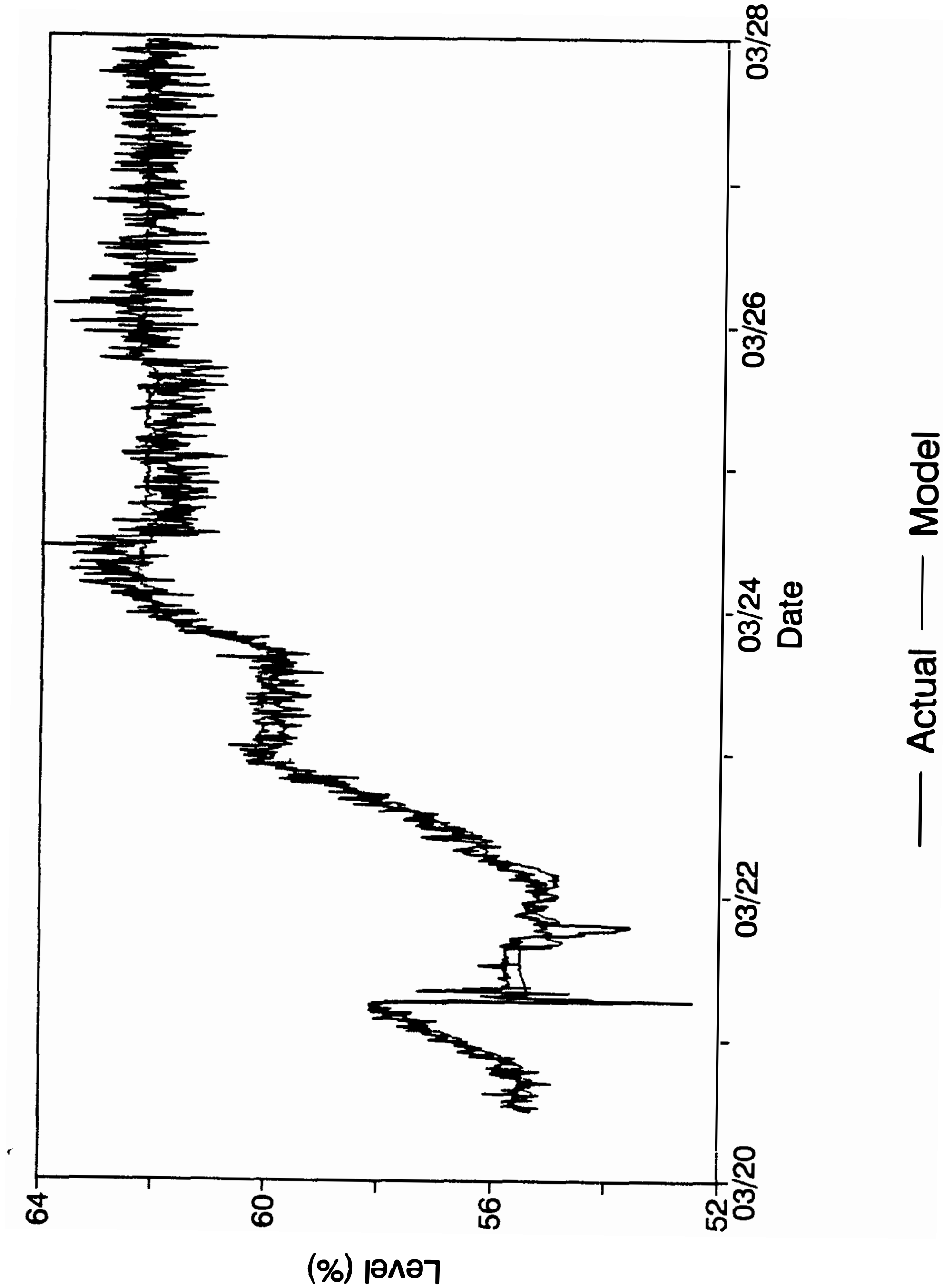

Figure 6.47: KFT estimation of steam generator wide range water level for PWR-1 with level and pressure measurements excluded.

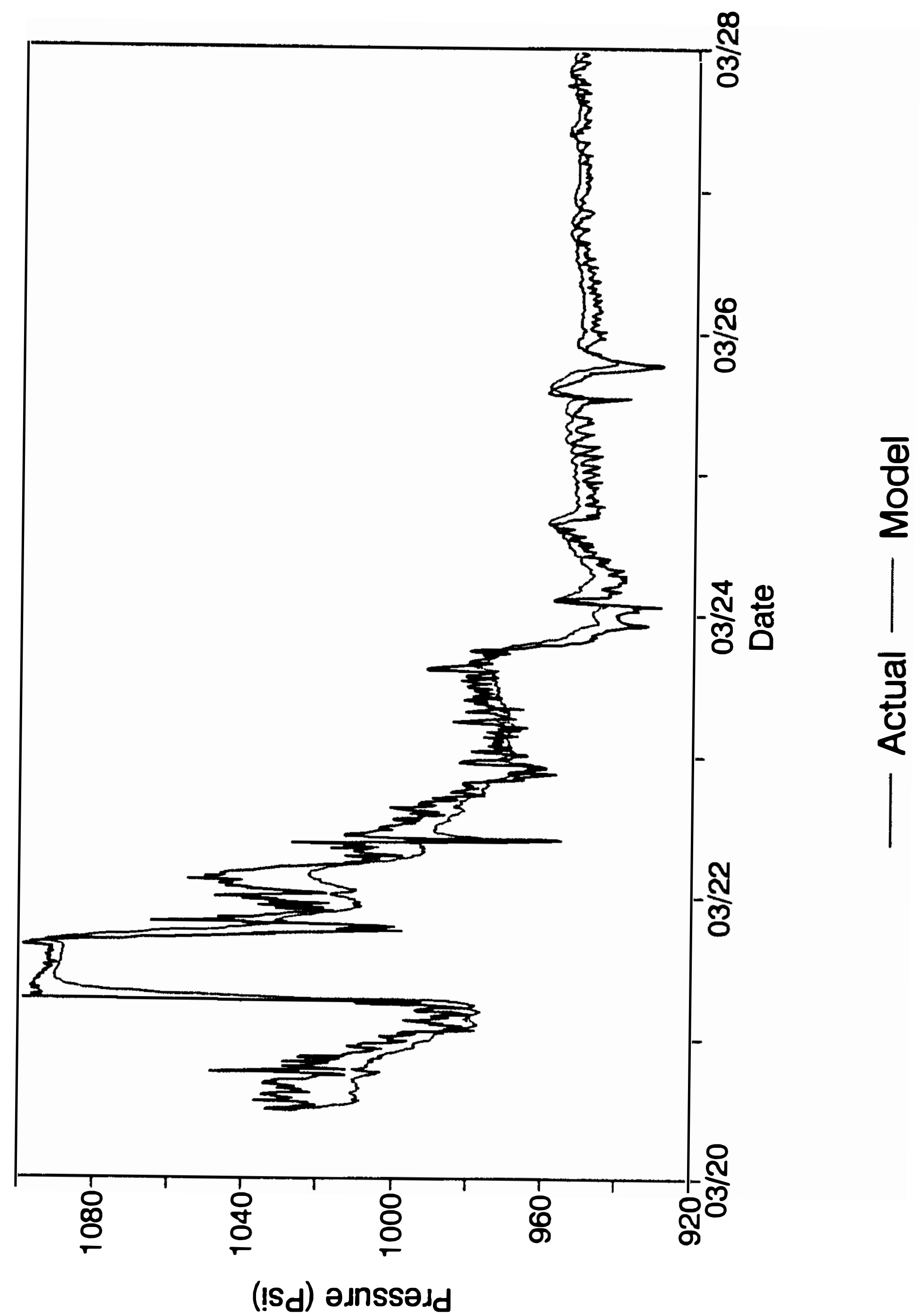

Figure 6.48: KFT estimation of steam generator pressure for PWR-2 with level and pressure measurements excluded.

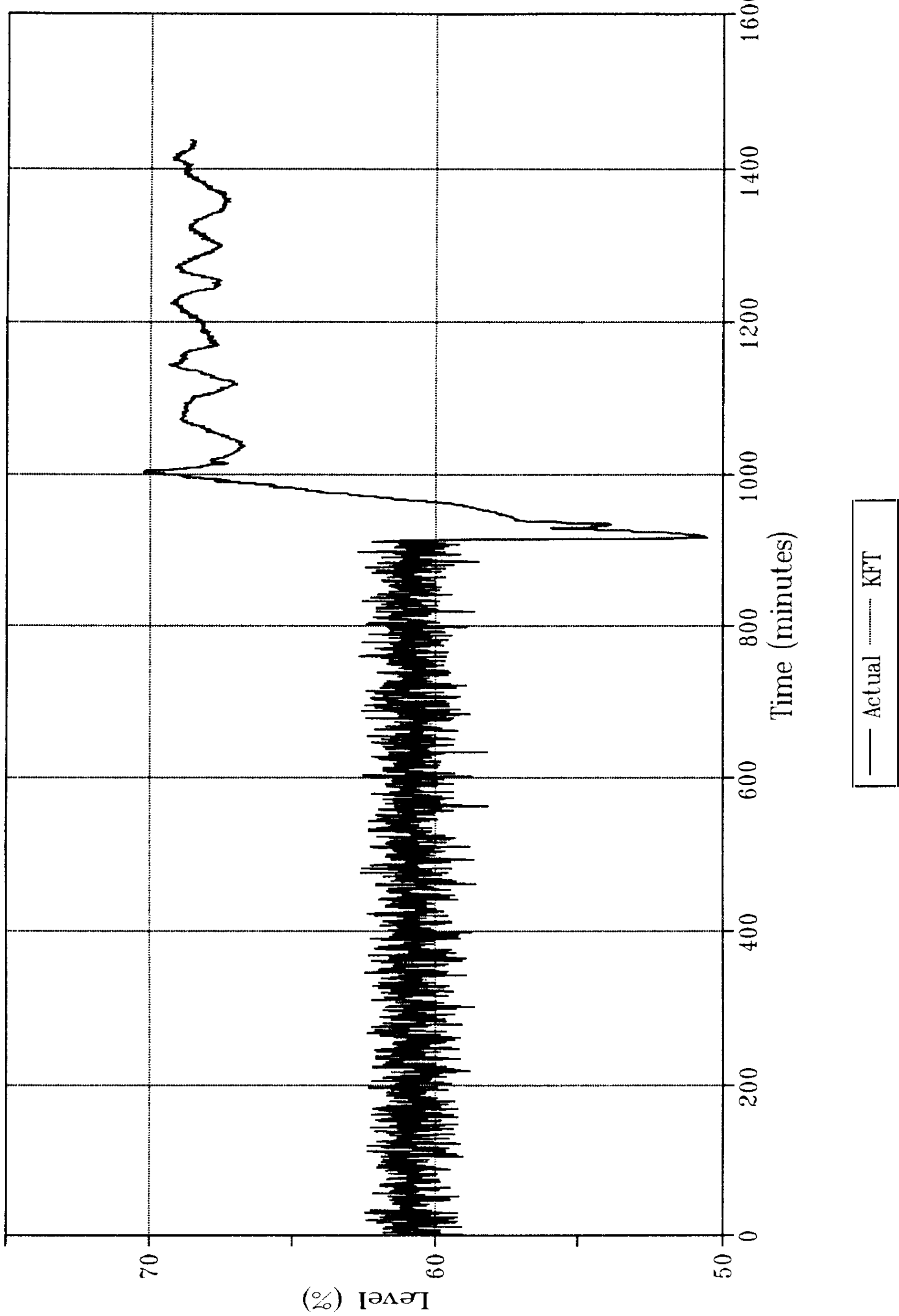

Figure 6.49: KFT estimation of steam generator wide range water level for PWR-2 with level and pressure measurements included.

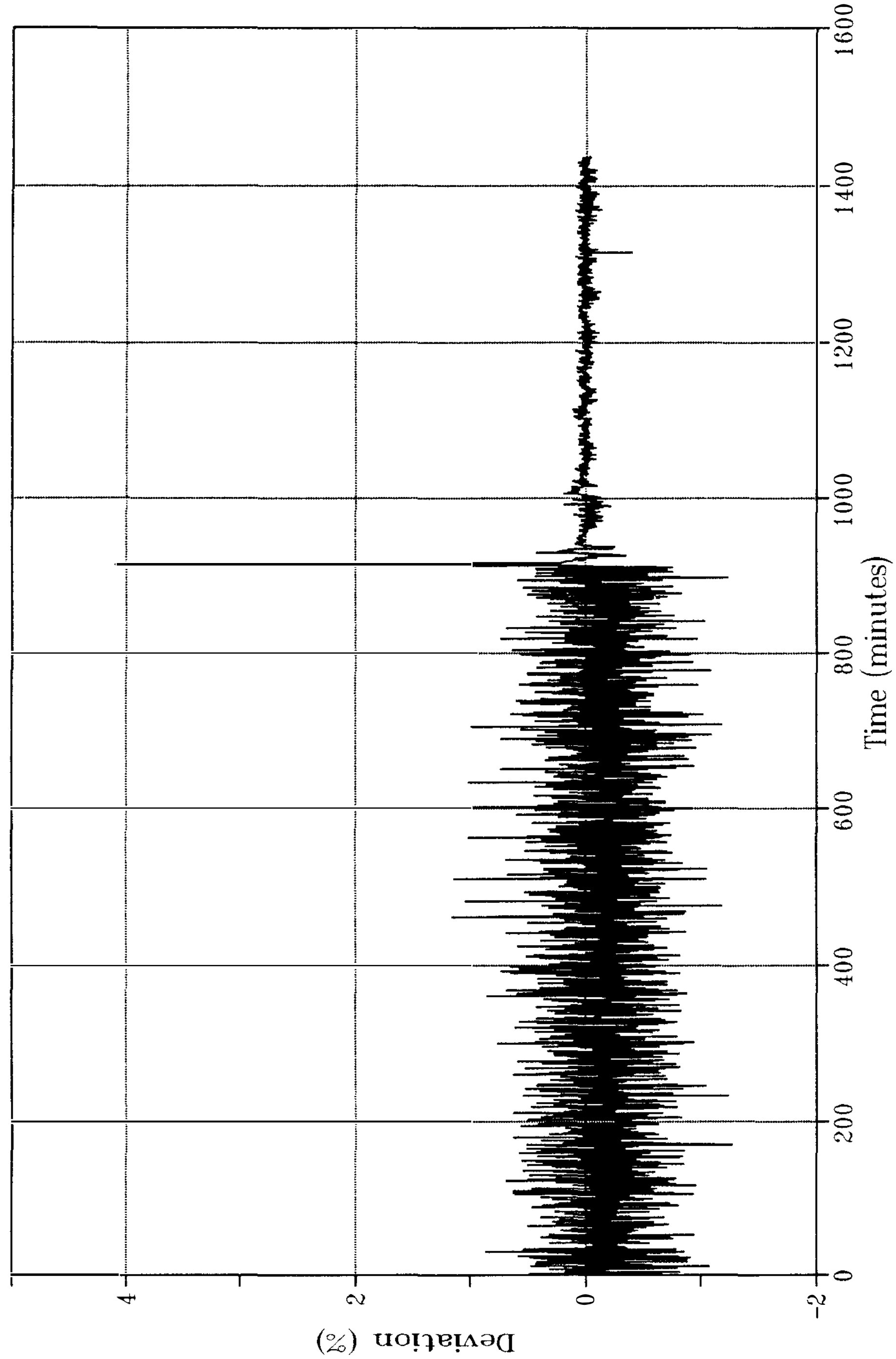

Figure 6.50: Error in KFT estimation shown in Figure 6.49.

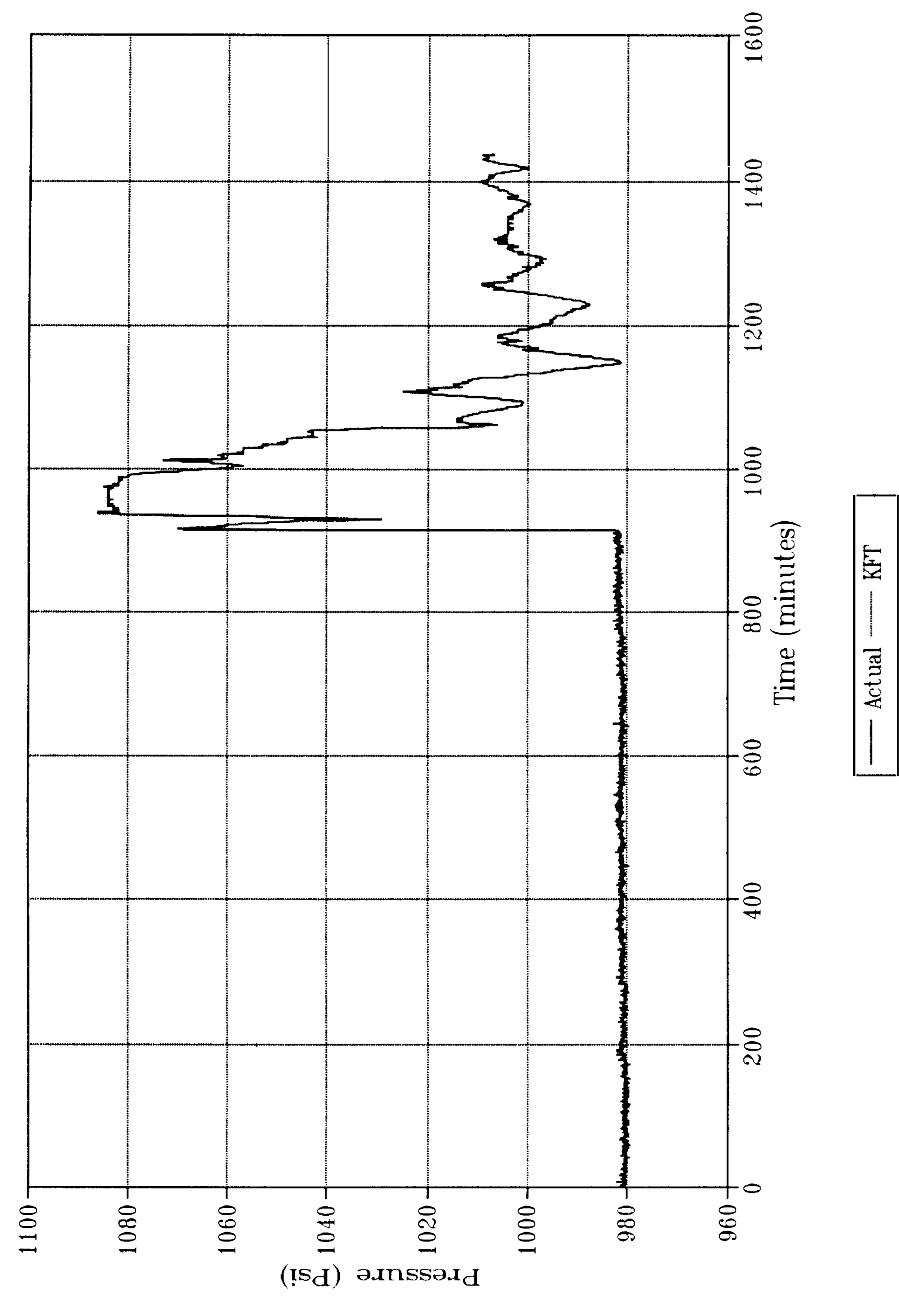

Figure 6.51: KFT estimation of steam generator pressure for PWR-2 with level and pressure measurements included.

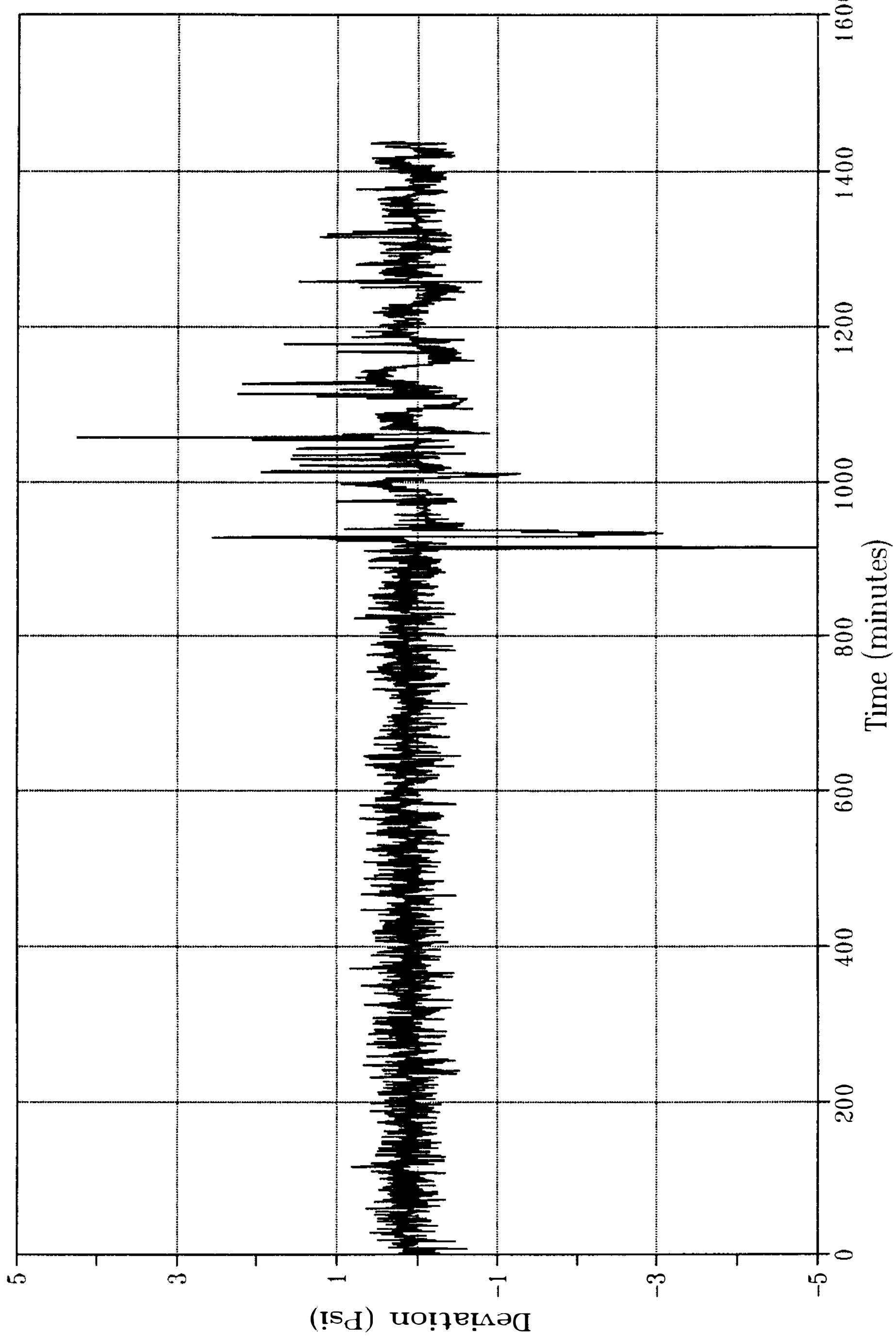

Figure 6.52: Error in KFT estimation shown in Figure 6.51.

Figures 6.45 and 6.46 show the Kalman filtering correction for steam generator wide range water level and steam generator pressure estimates. These figure indicate that the KFT corrects the inaccuracies of the model during steady-state and transient operating conditions.

Steam generator water level and pressure estimations are also computed without including their measurements as shown in Figure 6.47 and Figure 6.48 respectively. However, the use of measurements to be validated provides a higher accuracy in estimating these variables. One of the reasons that the KFT could not track the steady-state fluctuations, may be that the UTSG model has inaccuracies and certain assumptions (e.g. critical flow assumption), thus limiting the estimation accuracy. It is important to note that even though the UTSG model was developed using design data for the Sequoyah Nuclear Plant, it performed very well for PWR-1 and PWR-2 data analysis.

The noise covariance matrix Q is computed with the system noise variances of the computed state variables, whereas the noise covariance matrix R is computed with the noise variance of the sensors (e.g. steam generator pressure sensor noise sensor was 2 psig).

The KFT was incorporated as a DLL subroutine and is given in Appendix D.

6.6 System Executive and Graphical User Interface

The system executive was designed using Microsoft Visual Basic, which creates applications executable in Microsoft Windows 3.1™. The decision-making and the I/O scheduling were programmed in Visual Basic (Appendix E). The signal validation (SV) modules, written in FORTRAN and C, were compiled and added as a DLL library to the PC-based signal validation system.

The main navigational window, appears during the initial execution of the program. This window, shown in Figure 6.53, has hypertext buttons, which were created by Visual Basic graphical objects, thus adding virtual realism. Through this window, the user may access several other information windows, including instant data and historical trend plots windows of sensor measurements and signal estimates.

The information to be displayed was categorized mainly into two groups:

- Steam generator wide range level, and
- Steam generator pressure.

Different information windows were created for each variable to reduce the confusion in information display. Figures 6.54 and 6.55 show information windows for the steam generator narrow range water level and the steam generator pressure respectively. The information displayed in these windows are instantaneous displays of measured values, results of SV module estimates and fuzzy logic based decision-making results.

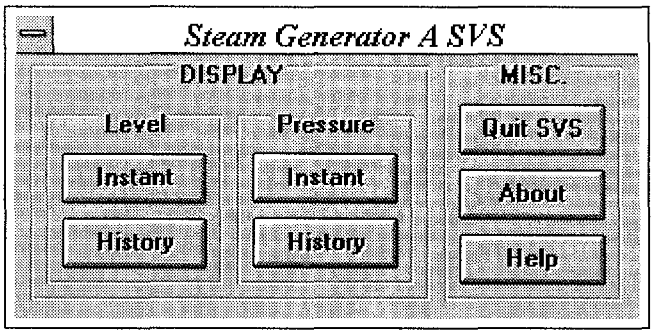

Figure 6.53: Main window for navigation through the signal validation information space.

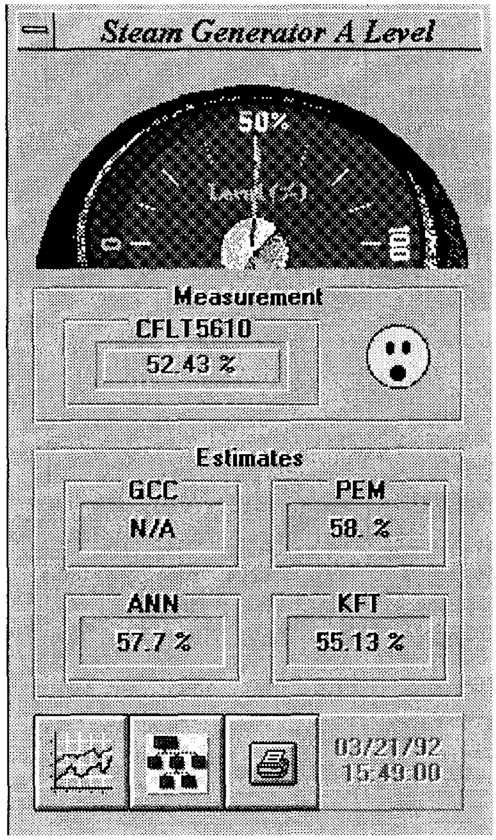

Figure 6.54: Information window for instantaneous steam generator wide range water level measurement and signal validation results.

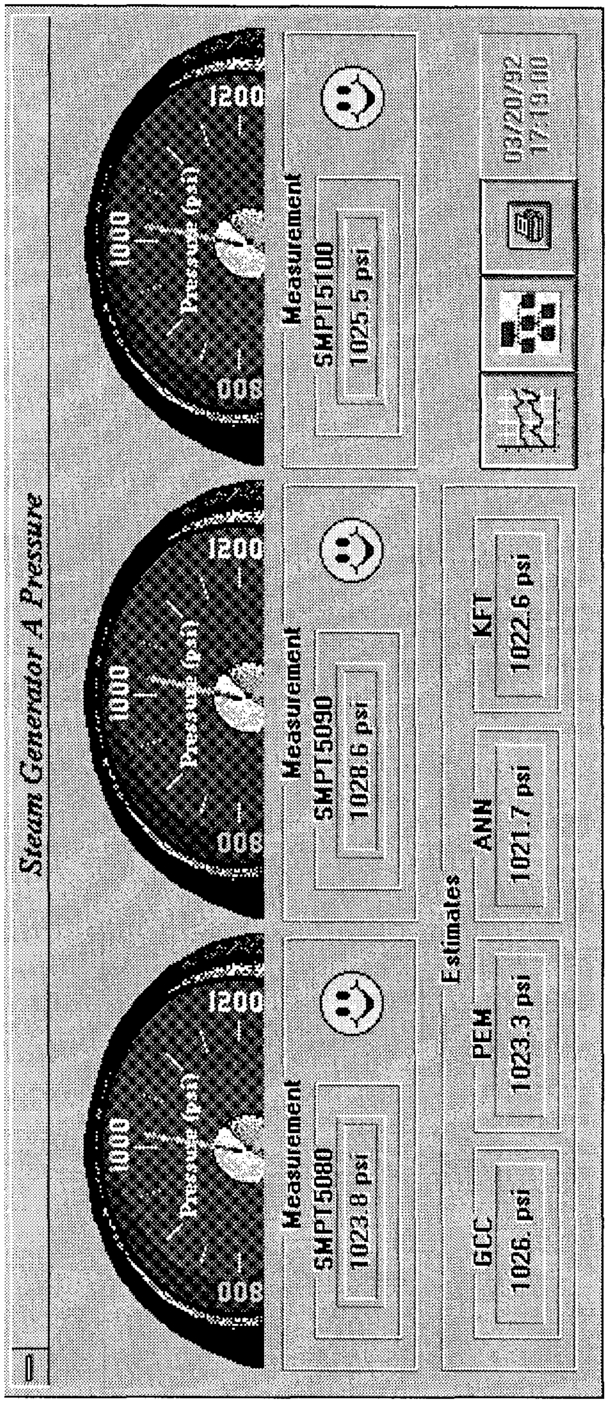

Figure 6.55: Information window of instantaneous steam generator pressure measurements and signal validation results.

A virtual realism was added to display measured sensor values in the analog form as well as in the digital form. The analog display of the sensor measurements may help plant operators to recognize SV information quickly and therefore take necessary preventive actions. On the other hand, digital displays are convenient ways to display information to plant engineers. Historical trend plots are also provided by the system executive. These information are incorporated in separate windows and are illustrated in Figures 6.56 - 6.58. Each of the instantaneous information windows displays results of decision-making in the form of the icon "smiley." Hypertext buttons are provided for each information window to navigate to the historical trend plot window, to the main window, or to print the current information window to obtain a hard copy.

Each of the historical trend plots includes point-by-point comparison of actual measurement and the signal estimates from all the four signal validation modules. This display can be selected using the bottom hypertext button of each window. In addition an historical plot of the quality index is also shown as a function of time. The quality index has the following meanings.

0 = safe,

1 = no fault,

2 = fault warning,

3 = fault,

4 = severe fault.

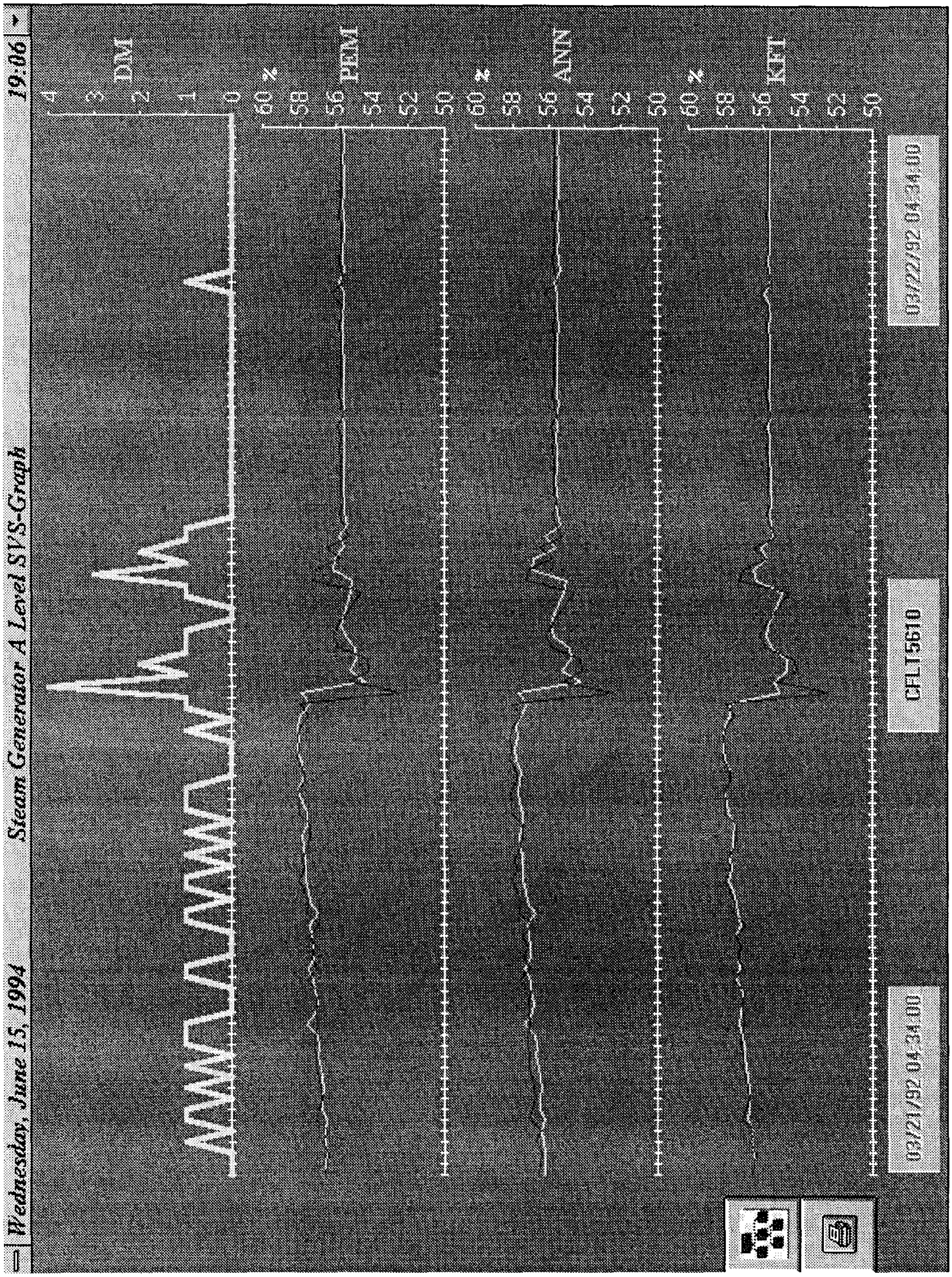

Figure 6.56: Information window displaying the historical trend of steam generator wide range water level and SV results.

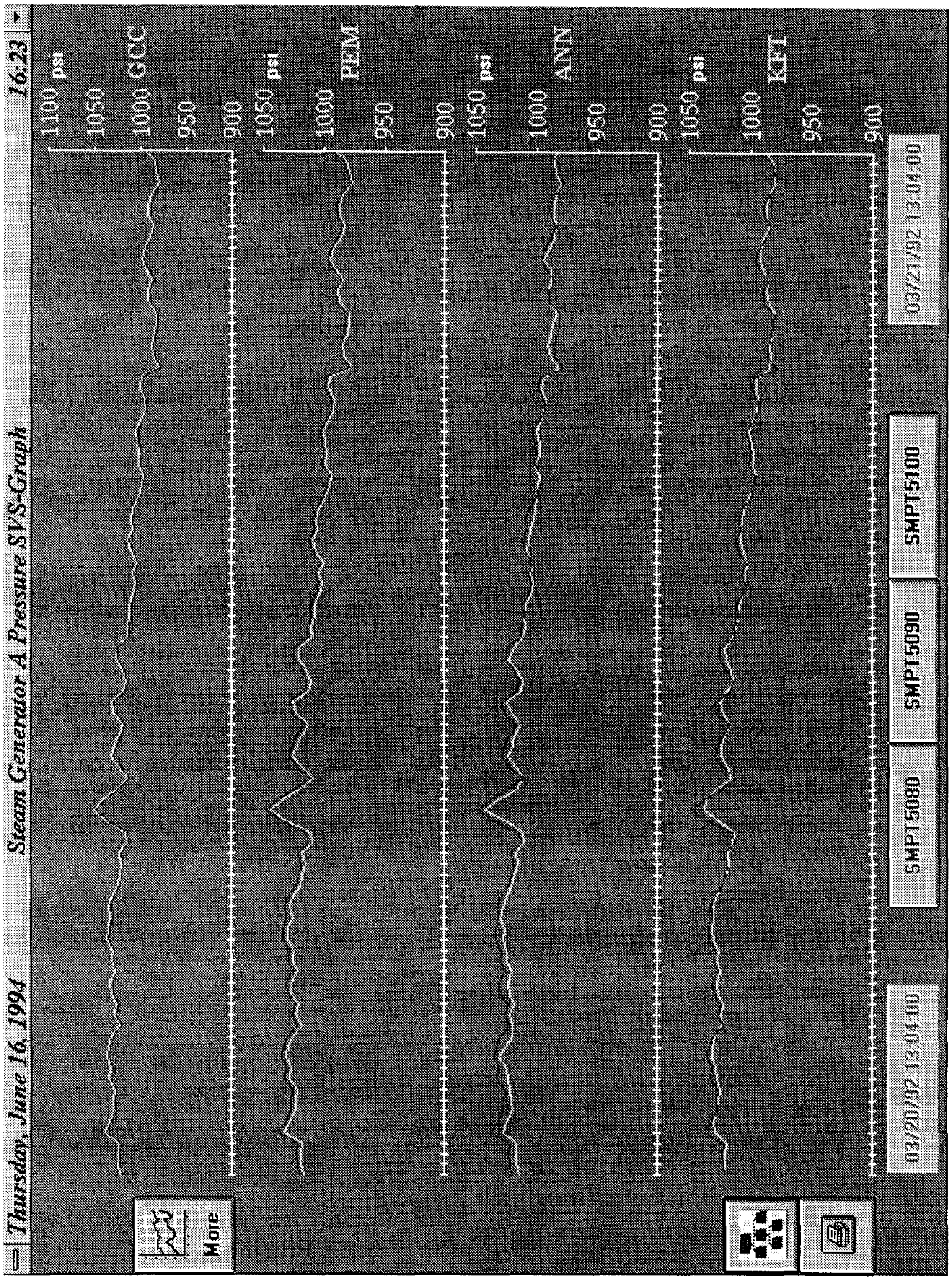

Figure 6.57: Information window displaying the historical trend of steam generator pressure and SV module estimates.

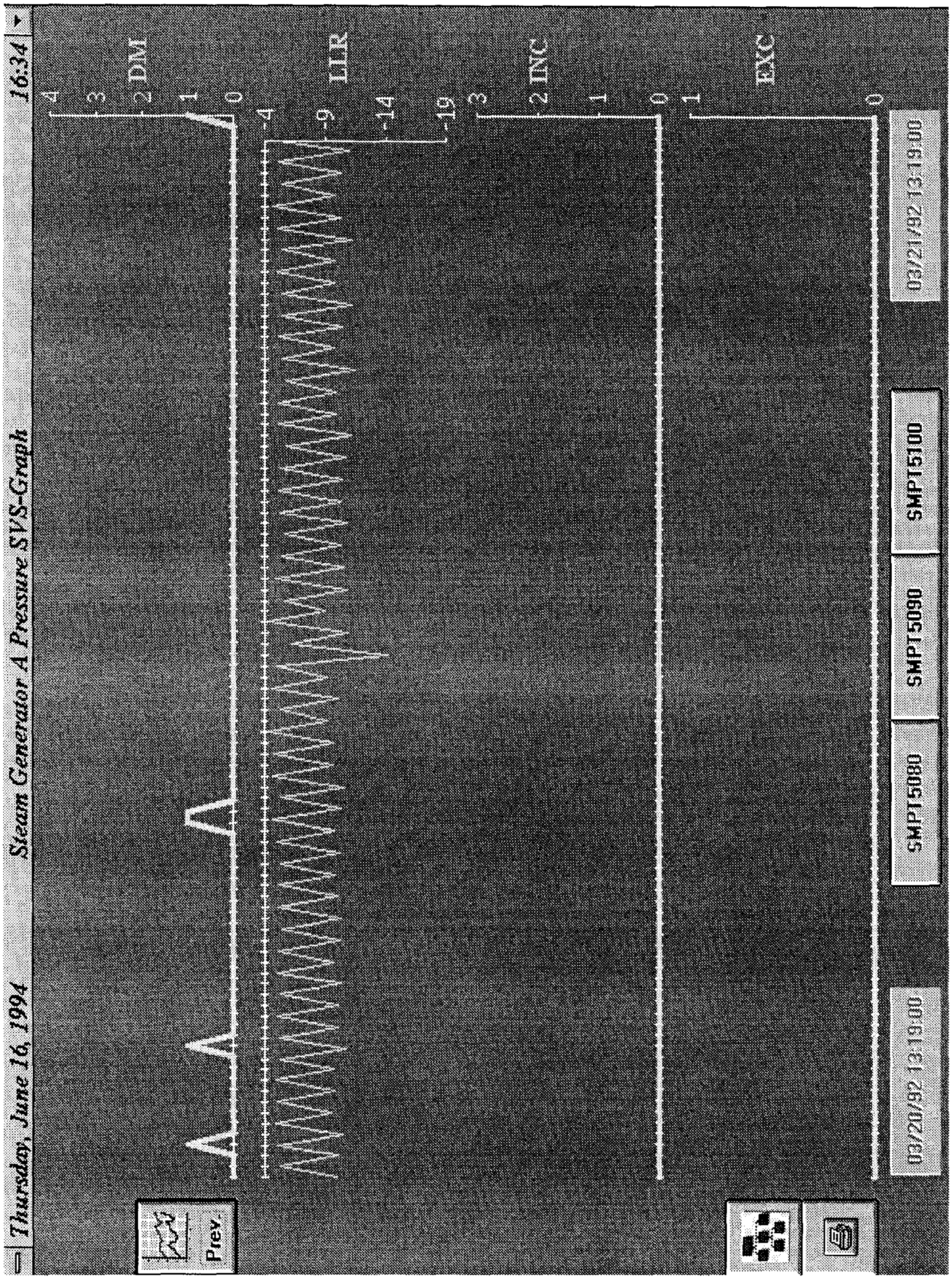

Figure 6.58: Information window displaying the historical trend of SV decision-making results for steam generator pressure.

A graphical representation of these values, which are incorporated into the library of prototype fuzzy sets is shown in Figure 6.59.

As the decision-making plots indicate, faults were detected in the steam generator wide range level for some time instants. The fault might have occurred due to the level fluctuations inside the UTSG during process transients. The decision-making algorithm of the system executive detected very few anomalies (in spikes) in the steam generator pressure sensors.

An example of how a decision is reached is shown in Figure 6.60. The result of the decision-making is shown using the icon representation “smiley” in Figure 6.55.

The system executive also provides some hypertext buttons, which provide links to product information and on-line help. Figure 6.61 shows such a window, displaying product information about the PC-based signal validation system.

The design of the system executive is such that any of the existing SV module can be removed or any new SV module can be added. However, each module must be adapted for specific nuclear power plants.

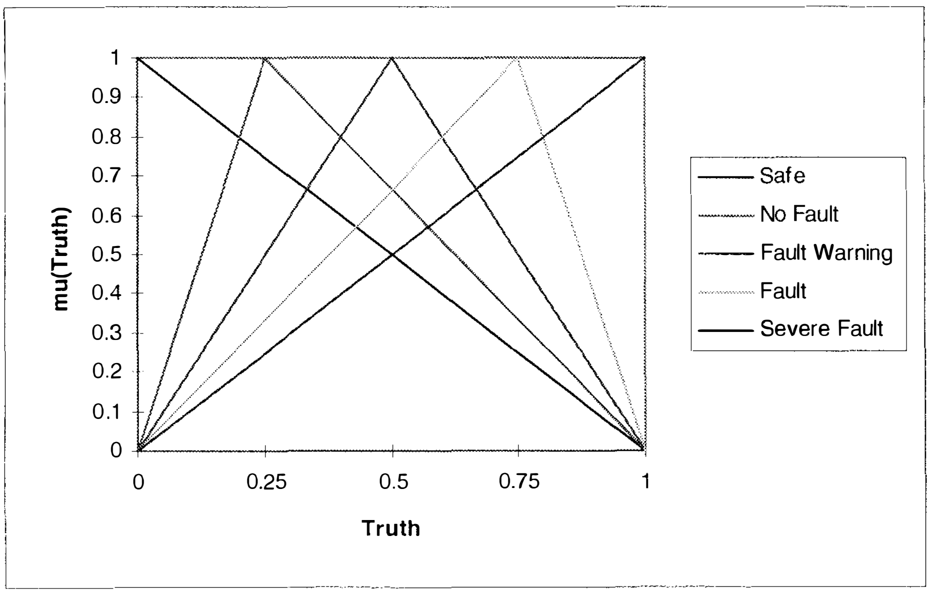

Figure 6.59: Library of prototype fuzzy sets.

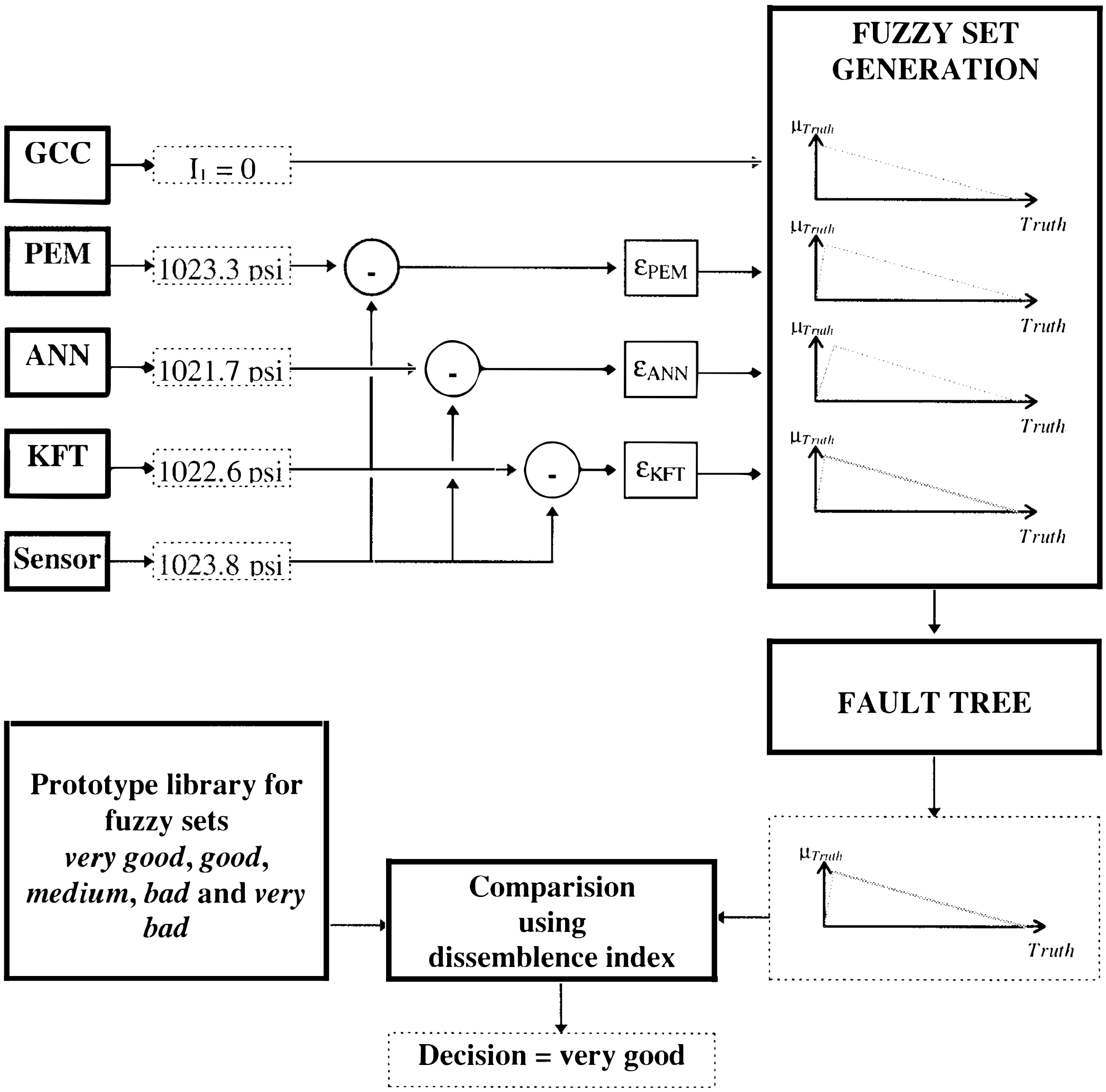

Figure 6.60: An example of making a decision for sensor SMPT5080 status.

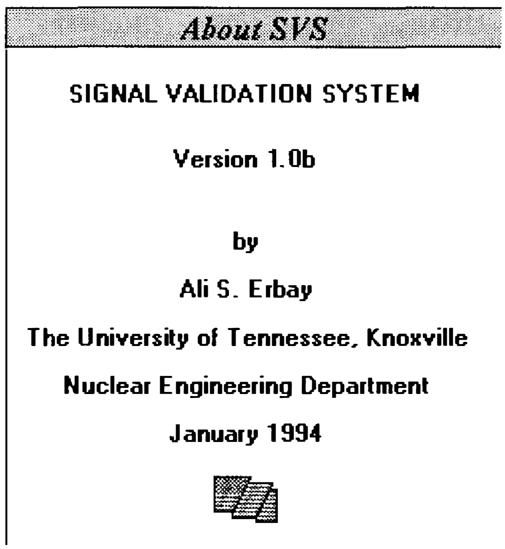

Figure 6.61: Information window displaying the product information about the PC-based signal validation system.

6.7 Summary of Results

In this chapter, results of the performance of the individual signal validation modules and the system executive are presented. A comparision of the four SV modules is provided for steam generator pressure and level in two PWR's. The modules were developed off-line using operational data from PWR-1 and PWR-2.

According to the module results, successful modeling using ANN's and PEM yields very similar results for the same training data set. On the other hand the GCC and KFT modules produce results that agree very well with normal measurements. The use of redundant measurements and taking their average as an estimate in the GCC module is the main reason for its excellent performance. The performance of the KFT module is enhanced by including all available plant measurements and a system model.

The following models and modules were integrated into the PC-based signal validation system.

- GCC module for steam generator pressure measurements.
- Static ANN model for steam generator pressure measurements.
- Dynamic ANN model for steam generator wide range water level measurements.
- Static PEM model for steam generator pressure measurements.
- Dynamic PEM model for steam generator wide range water level measurements.

- KFT with level and pressure measurements.

These modules were successfully integrated through a system executive, which also includes a sensor status evaluation unit. Fuzzy logic and fault-tree methodology were used to make the final validity check. . The methodology in system executive design makes the replacement of modules easy.

The GUI was developed using Visual Basic, so that it is compatible with Microsoft Windows™ graphical objects. The GUI also used virtual realism in displaying data, as well as icon representations and historical plots.

Chapter 7

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

7.1 Summary

A PC-based signal validation system, incorporating previously developed techniques and two new modules, was developed and applied to operational data from two PWR's. The system consists of four signal validation modules (generalized consistency checking, process empirical modeling, artificial neural networks, and the Kalman filtering technique) and a system executive with a fuzzy logic decision-maker.

While the previously developed GCC module was integrated into the PC-based signal validation system, the results obtained from the PEM were incorporated as functions into the program. NeuralWorks provided similar tools for producing functions in C, which were similarly incorporated into the dynamic link libraries (DLL).

A detailed UTSG model was utilized to construct an extended Kalman filter. The KFT module was carefully tuned and programmed as a FORTRAN subroutine. This module was later integrated into the DLL.

The system executive was designed with the aid of Microsoft Visual Basic, which has the ability to create hypertext links and GUI objects in a very easy manner.

A decision-making algorithm, combining fuzzy logic and fault-tree analysis, was developed to establish the degree of a sensor fault. The results were presented in the linguistic domain such as *safe*, *no fault*, *fault warning*, *fault* and *severe fault*. These results were later displayed in icons, to make them easily recognizable on a fast updating information window.

7.2 Conclusions

In general, the results obtained from the studies in this thesis have shown the feasibility of implementing a PC-based signal validation system for nuclear power plants. The UTSG in a nuclear power plant was the focus of study in this research. Steam generator water level and steam generator pressure signals from a UTSG were used to illustrate the performance of the four signal validation modules. The Kalman filtering technique was used for the first time in this system and was found to be very robust. The UTSG model developed using data for TVA's Sequoyah Nuclear Plant (SNP). This model performed very well when used in conjunction with measurements from another similar plant.

The PC-based signal validation system was tested using off-line data obtained from two PWR's. The sampling interval for PWR-1 data was 15 minutes, whereas the sampling interval for PWR-2 data was 30 seconds. It was noticed in this study, that the sampling time of data is important to obtain an accurate model for the PEM and ANN modules. For example, if fluctuating water levels are measured at very short sampling intervals, the model to be fit to the data may confuse the training phase of constructing these models. Therefore, longer sampling times are more suitable for steam generator water level (e.g. in the order of minutes). However, steam generator pressure may be measured with shorter sampling time intervals.

While, the incorporation of the GCC module in the PC-based signal validation system was straightforward, the development of the PEM and ANN modules required several variations in input signal selection. Static models were found to be sufficient to validate the steam generator pressure. The steam generator wide-range water level signal was modeled successfully with dynamic structures such as incorporating past measurements of the input variables. The use of such models is very common in the ANN literature, whereas the same technique was applied to the PEM module for the first time to construct a model of the steam generator wide range water level.

The sensitivity analysis of the dynamic models showed that the most important signal for steam generator water level estimation was the hot leg temperature, while the most important signal for steam generator pressure estimation was the cold leg temperature.

Another important observation that was made during this study was the similarity between ANN's and PEM: for the same number and type of training patterns (100 training patterns over the entire transient and steady-state operating conditions) both models behaved similarly in predicting the signals to be validated.

Since the KFT requires an analytical model of the system to be validated, previously developed models of the UTSG were used in the KFT. The measurement of several signals is crucial in obtaining a good KFT estimate. Since the KFT module and the UTSG model make several assumptions (given in Chapters 3 and 4), the state estimation has some error, compared with the state measurement. Excellent results can be obtained by including state measurement in correcting the KFT estimate. The exclusion of these signals may introduce some error in the estimation. If the steam generator wide range level itself was included in the measurement vector, the KFT could successfully make an estimation for both steady-state and transient operating conditions.

A fault-tree methodology provided a useful tool in developing a procedure for sensor status determination. To reach a final conclusion, a fuzzy logic was used for fault-tree computations. The results were displayed in a user-friendly manner by means of icons, so that the results of the decision-making could be recognized easily, even at short sampling time intervals and at a high screen information update rate.

The development of the signal validation system in the Microsoft Windows 3.1™ environment enabled the use of effective GUI's. Since this is a common operating system, this validation technology can be easily ported to compatible PC's in nuclear power plants.

7.3 Recommendation for Future Research

The SV modules of the PC-based signal validation system were developed using off-line data obtained from two PWR's. The system is currently under development for implementation at the SNP. Plant specific ANN and PEM models will be constructed for this system.

The use of fast back-propagation algorithm may be replaced with the Logicon Projection Network™, which, according to the vendor, is supposed to be the fastest error minimizing neural network. Thus, at least the static ANN models may be updated for field applications.

The GUI, especially the historical trend plots of the measurements and SV results, may be adjusted for the convenience of different users at different power utilities.

LIST OF REFERENCES

LIST OF REFERENCES

[1] P. Polenta et al., "Implementation of a Fault Detection Procedure," Proceeding of the 1986 American Control Conference, Seattle, Washington, pp. 176 - 181, June 1986.

[2] B. R. Upadhyaya et al, "Signal Validation in Nuclear Power Plants," Annual Report prepared for the U.S. Department of Energy by The University of Tennessee, DOE/NE/37959-24, September 1988.

[3] R. Onken and N. Stuckenburg, "Failure Detection in signal Processing and Sensing in Flight Control Systems," Proceedings of the 1978 IEEE Conference on Decision and Control, pp. 449 - 454, 1978.

[4] E. Y. Shapiro and H. E. Decarli, "Analytic Redundancy for Flight control Sensors on the Lockheed L-1011 Aircraft," Proceedings of the 1978 IEEE Conference on Decision and Control, pp.371 - 376, 1978.

[5] R. C. Montgomery and D. Tabak, "Application of Analytical Redundancy Management to shuttle Crafts," Proceedings of the 1978 IEEE Conference on Decision and Control, pp. 442 - 448, 1978.

[6] O. L. Deutsch, R. S. Ornedo and R. W. Lindsay, "Implementation of Real-Time Signal Validation at EBR-II," Transactions of the American Nuclear Society, Vol. 45, pp. 660 - 661, 1983.

[7] B. J. Benedict, R. I. Lutz and L. P. Smith, "Validation of Critical Signals for the Safety Parameter Display System," EPRI Report, NP-5066M, April 1987.

[8] J. Mott, R. Young and R. King, "Pattern Recognition Software for Plant Surveilance," Report prepared for the U.S. Department of Energy, 1987.

[9] R. D. Fournier et al., "Digital Control and Protection Retrofits in Nuclear Power Plants," Nuclear News, American Nuclear Society, La Grange Park, Illinois, pp. 50 - 57, November 1988.

[10] K. E. Holbert, "Comprehensive Signal Validation for Nuclear Power Plants," Ph.D. Dissertation, The University of Tennessee, August 1989.

[11] B. R. Upadhyaya et al, "Development and Testing of an Integrated Signal Validation System for Nuclear Power Plants," Volume 1, 2, 3, Research prepared

for the U.S. Department of Energy, DOE/NE/37959-34, 35, 36, September 1989.

[12] A. L. Qualls, "Signal Validation using Expert System Technology," M.S. Thesis, The University of Tennessee, June 1988.

[13] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Journal of Basic Engineering, Volume 82, 1960.

[14] A. V. Oppenheim, "Applications of Digital Signal Processing," Prentice-Hall, New Jersey, 1978.

[15] S. K. Yung and D. W. Clarke, "Local Sensor Validation", Measurement + Control, Volume 22, June 1989.

[16] B. R. Upadhyaya and M. Skorska, "Sensor Fault Analysis using Decision Theory and Data-Driven Modeling of Pressurized Water Reactor Subsystems", Nuclear Technology, Volume 64, January 1984.

[17] A. Erbay and A. Ikonomopoulos, "A Fuzzy Logic Methodology for Fault-Tree Analysis in Critical Safety Systems," Transactions of the American Nuclear Society, Volume 68, Part A, pp. 134 -135, San Diego California, June 1993.

[18] P. K. Simpson, "Artificial Neural Networks," Pergamon Press, New York, 1990.

[19] R. Hecht - Nielsen, "Neurocomputing," Addison - Wesley, Massachusetts, 1990.

[20] M. Naghedolfeizi, "Dynamic Modeling of A Pressurized Water Reactor Plant for Diagnostics and Control," M.S. Thesis, The University of Tennessee, December 1990.

[21] M. R. A. Ali, "Lumped Parameter, State Variable Dynamic Models for U-Tube Recirculation Type Nuclear Steam Generators," Ph.D. Dissertation, The University of Tennessee, 1976.

[22] E. Eryürek, "A Parallel, Fault-Tolerant control and Diagnostics System for Nuclear Power Plants," Ph. D. Dissertation, The University of Tennessee, 1994.

[23] R. E. Uhrig, "Applications of Neural Networks to the Operation of Nuclear Power Plants," Proceedings of SMORN VI, Gatlinburg, Tennessee, May 1991.

[24] K. Kavaklioglu, "Thermal Performance Monitoring and Diagnostics of Pressurized Water Reactors using Artificial Neural Networks," M.S. Thesis, The University of Tennessee, May 1993.

[25] B. R. Upadhyaya and E. Eryürek, “Sensor Validation for Nuclear Power Plants using Adaptive Backpropagation Neural Networks,” IEEE Tranactions on Nuclear Sciences, Volume 37, pp. 1040 - 1047, April 1990.

[26] E. Eryürek, “Development and Application of Multi-layer Neural Networks for Estimation of Power Plant Variables,” M.S. Thesis, The University of Tennessee, 1991.

[27] NeuralWare, “Neural Computing,” Technical Publications Group, NeuralWare Inc., 1993.

[28] R. G. Brown, “Introduction to Random Signals and Applied Kalman Filtering,” New York 1992.

[29] J. V. Candy, “Signal Processing, The Model-Based Approach,” Mc-Graw-Hill, New York, 1986.

[30] B. R. Upadhyaya, “Theory of Information Processing,” Chapter 11, The University of Tennessee, 1992.

[31] T. Kailath, “Lectures on Wiener and Kalman Filtering,” Springer, New York, 1981.

[32] “Westinghouse Design Pressurized Water Reactor Technology Manual”, pp. 6 -11, 1991.

[33] L. Casey, “The Virtual Reality Primer,” McGraw-Hill, New York, 1993.

[34] M. M. Gupta et al, “Fuzzy Automata and Decision Process,” Elsevier North-Holland, New York, 1977.

[35] L. A. Zadeh, “The Concept of a Linguistic Variable and Its Applications to Approximate Reasoning,” ERL-M411, Berkeley, California, October 1973.

[36] A. Kaufmann and M. M. Gupta, “Introduction to Fuzzy Arithmetic,” Van Nostrand Reinhold, New York, 1991.

[37] A. Jones et al, “Fuzzy Sets Theory and Applications,” NATO ASI Series, Series C, Volume 177, Belgium, July 1985.

[38] Microsoft Corporation, “Microsoft Visual Basic,” 1993.

APPENDICES

APPENDIX A

Code Listing for Generalized Consistency Checking

```
Sub gcc ()
  meas(1) = deger(8)
  meas(2) = deger(9)
  meas(3) = deger(10)
  k = nsignl
  For j = 1 To nsignl
   darray(j, 1) = j
   excl(j) = 0#
  Next j
800 :
  For j = 1 To nsignl
   index(j) = 0
  Next j
  For j = 1 To nsignl - 1
   JS = darray(j, 1)
   For l = j + 1 To nsignl
    ls = darray(l, 1)
    If Abs(meas(JS) - meas(ls)) > (erb(JS) + erb(ls)) Then
     index(JS) = index(JS) + 1
     index(ls) = index(ls) + 1
    End If
   Next l
  Next j
  For j = 1 To nsignl
   JS = darray(j, 1)
   darray(j, 2) = meas(JS)
   darray(j, 3) = index(JS)
  Next j
  itest = 0
  For j = 1 To nsignl
   itest = itest + index(j)
  Next j
  If itest = 0 Then
   Call estmat
   If k = nsignl Then
    GoTo 900
   Else
    For j = k + 1 To nsignl
     excl(darray(j, 1)) = 1#
    Next j
    GoTo 900
   End If
  End If
700 :
  For l = 1 To k - 1
   JMIN = l
   For j = l + 1 To k
    If darray(j, 3) < darray(JMIN, 3) Then JMIN = j
   Next j
   TEMP1 = darray(l, 1)
   TEMP2 = darray(l, 2)
   TEMP3 = darray(l, 3)
   darray(l, 1) = darray(JMIN, 1)
   darray(l, 2) = darray(JMIN, 2)
   darray(l, 3) = darray(JMIN, 3)
   darray(JMIN, 1) = TEMP1
   darray(JMIN, 2) = TEMP2
   darray(JMIN, 3) = TEMP3
  Next l
  For j = 1 To k
   JS = darray(j, 1)
  Next j
  imax = darray(k, 3)
  imin = darray(1, 3)
  nmax = 0
```

```
  For l = k To 1 Step -1
   If darray(l, 3) = darray(k, 3) Then nmax = nmax + 1 Else Exit For
  Next l
  If imax = 0 Or imin = 0 Then
   Call estmat
   If k = nsignl Then
    GoTo 900
   Else
    For j = k + 1 To nsignl
     excl(darray(j, 1)) = 1#
    Next j
    GoTo 900
   End If
  End If
  If imax = (k - 1) Then
   If k = nmax Then
    Call pastest
    If jinclp = 0 Then
     For l = 1 To k
      darray(l, 3) = k - 1#
     Next l
     For j = 1 To nsignl
      excl(darray(j, 1)) = 1#
     Next j
     GoTo 910
    End If
    k = jinclp
    For j = 1 To nsignl
     For l = 1 To jinclp
      If j = ninclp(l) Then GoTo 60
     Next l
     excl(darray(j, 1)) = 1#
60 :
    Next j
    GoTo 900
   Else
    k = k - nmax
    For l = 1 To k
     darray(l, 3) = darray(l, 3) - nmax
    Next l
    GoTo 700
   End If
  End If
  If nmax = 1 Then
   k = k - nmax
   GoTo 800
  End If
  If k = nmax Then
   Call pastest
   If jinclp = 0 Then
    For l = 1 To k
     darray(l, 3) = k - 1#
    Next l
    For j = 1 To nsignl
     excl(darray(j, 1)) = 1#
    Next j
    GoTo 910
   End If
   k = jinclp
   For j = 1 To nsignl
    For 1 = 1 To jinclp
     If j = ninclp(1) Then GoTo 85
    Next l
    excl(darray(j, 1)) = 1#
85 :
   Next j
   GoTo 900
  End If
  k = k - nmax
  GoTo 800
900 :
  For j = 1 To nsignl
   isig(j) = darray(j, 1)
   X(isig(j)) = Abs(darray(j, 2) - xestmt)
   BSETTO0(j) = 0
   sprtb(isig(j)) = sprtb(isig(j)) + BIAS(isig(j)) * (X(isig(j)) - BIAS(isig(j)) / 2#) /
VAR0(isig(j))
   If sprtb(isig(j)) > boundb Then
```

```
    BSETTO0(isig(j)) = 1
   End If
   If sprtb(isig(j)) < bounda Then
    BSETTO0(isig(j)) = 1
   End If
  Next j
910 :
  For l = 1 To nsignl
   nsid = darray(l, 1)
   For j = 1 To nsignl
    If nsid = j Then nplace(j) = l
   Next j
  Next l
  For l = 1 To nsignl
   If sprtb(l) >= boundb Then DBIAS(l) = DBIAS(l) + 1
   If sprtb(l) <= bounda Then NBIAS(l) = NBIAS(l) + 1
   If BSETTO0(l) = 1 Then sprtb(l) = 0#
   NEXCL(l) = NEXCL(l) + excl(l)
   SII(l) = SII(l) + darray(nplace(l), 3)
   SUM(l) = SUM(l) + meas(l)
  Next l
  pestmt = xestmt
  xtot = xtot + xestmt
End Sub

Sub pastest ()
 jinclp = 0
 JEXCLP = 0
 For l = 1 To k
  If Abs(darray(l, 2) - pestmt) < erb(darray(l, 1)) Then
   jinclp = jinclp + 1
   ninclp(jinclp) = l
  End If
 Next l
 If jinclp = 0 Then
  xestmt = pestmt
 Else
  W = 1#
  SUM1 = 0#
  SUM2 = 0#
  For l = 1 To jinclp
   SUM1 = W * darray(ninclp(l), 2) + SUM1
   SUM2 = SUM2 + W
  Next l
  xestmt = SUM1 / SUM2
 End If
End Sub

Sub estmat ()
 SUM1 = 0#
 SUM2 = 0#
 For j = 1 To k
  If darray(j, 3) > (k - 1) / 2 Then
   SUM1 = SUM1 + (1 - (2# / ((k - 1) ^ 2)) * darray(j, 3) ^ 2) * darray(j, 2)
   SUM2 = SUM2 + (1 - (2# / ((k - 1) ^ 2)) * darray(j, 3) ^ 2)
  Else
   SUM1 = SUM1 + ((2# / ((k - 1) ^ 2)) * (k - 1 - darray(j, 3)) ^ 2) * darray(j, 2)
   SUM2 = SUM2 + ((2# / ((k - 1) ^ 2)) * (k - 1 - darray(j, 3)) ^ 2)
  End If
 Next j
 xestmt = SUM1 / SUM2
End Sub
```

APPENDIX B

Code Listing for Process Empirical Modeling

```
Sub pem (lev, pre)
 lev = 1.273137 * .331576 * eskideger(1) + .4570648 * .02491578 * deger(6) + .02568717 *
(.331576 * eskideger(1)) ^ 2 + .2142974 * .005272023 * deger(5) - .0005036674 * .005272023
* deger(5) * (.02491578 * deger(6)) ^ 2 + 17.95273
 pre = 15.52209 * .02975807 * eskideger(4) + 9.233875 * .02491578 * deger(6) + 70.69355 *
.01619268 * deger(2) - 4.102993 * .005272023 * deger(5) - 4.965631 * .03428328 * deger(3)
- 296.0486
End Sub
```

APPENDIX C

Code Listing for Artificial Neural Network

```
/* Wed Nov 03 15:35:14 1993 (lev.c) *//* Recall-Only Run-time for <level> */
/* Control Strategy is: <bkpfast> */

#if __STDC__
#define  ARGS(x) x
#else
#define  ARGS(x) ()
#endif /* __STDC__ */

/* --- External Routines --- */
extern double tanh  ARGS((double));
/* *** MAKE SURE TO LINK IN YOUR COMPILER's MATH LIBRARIES *** */

#if __STDC__
int level( void *NetPtr, float Yin[6], float Yout[1] )
#else
int level( NetPtr, Yin, Yout )
void  *NetPtr; /* Network Pointer (not used) */
float Yin[6], Yout[1];  /* Data */
#endif /* __STDC__ */
{
    float  Xout[19];  /* work arrays */
    long   ICmpT;  /* temp for comparisons */

    /* *** WARNING: Code generated assuming Recall = 0 *** */

    /* Read and scale input into network */
    Xout[2] = Yin[0] * (0.031825828) + (-18.827435);
    Xout[3] = Yin[1] * (0.3618934) + (-201.43719);
    Xout[4] = Yin[2] * (0.022922898) + (-1.0002407);
    Xout[5] = Yin[3] * (0.46774761) + (-39.225541);
    Xout[6] = Yin[4] * (0.5360986) + (-1.0257735);
    Xout[7] = Yin[5] * (0.012248213) + (-12.403489);
LAB110:

    /* Generating code for PE 5 in layer 2 */
    Xout[7] = 0; /* Disabled PE */

    /* Generating code for PE 0 in layer 3 */
    Xout[8]  = (float)(2.3408449) + (float)(1.4408469) * Xout[2] +
        (float)(-1.167382) * Xout[3] + (float)(0.20936276) * Xout[4] +
        (float)(0.127442) * Xout[5] + (float)(0.19233584) * Xout[6] +
        (float)(0.042934984) * Xout[7];
    Xout[8] = tanh( Xout[8] );

    /* Generating code for PE 1 in layer 3 */
    Xout[9]  = (float)(-2.4680657) + (float)(-1.1423932) * Xout[2] +
        (float)(0.52887523) * Xout[3] + (float)(-0.14712702) * Xout[4] +
        (float)(-0.10842845) * Xout[5] + (float)(0.0019552575) * Xout[6] +
        (float)(-0.039009955) * Xout[7];
    Xout[9] = tanh( Xout[9] );

    /* Generating code for PE 2 in layer 3 */
    Xout[10]  = (float)(-2.4405954) + (float)(-0.34002388) * Xout[2] +
        (float)(0.26536015) * Xout[3] + (float)(0.35480371) * Xout[4] +
        (float)(0.066222005) * Xout[5] + (float)(0.27552718) * Xout[6] +
        (float)(0.082573548) * Xout[7];
    Xout[10] = tanh( Xout[10] );

    /* Generating code for PE 3 in layer 3 */
    Xout[11]  = (float)(-2.6141725) + (float)(-0.1461968) * Xout[2] +
        (float)(-0.50969315) * Xout[3] + (float)(0.60152376) * Xout[4] +
        (float)(0.15280902) * Xout[5] + (float)(0.41527015) * Xout[6] +
        (float)(0.055616885) * Xout[7];
    Xout[11] = tanh( Xout[11] );
```

```
    /* Generating code for PE 4 in layer 3 */
    Xout[12]  = (float)(-1.9396212) + (float)(-0.29315224) * Xout[2] +
        (float)(-0.35174656) * Xout[3] + (float)(-0.075055979) * Xout[4] +
        (float)(-0.019682461) * Xout[5] + (float)(0.07240551) * Xout[6] +
        (float)(0.0062356647) * Xout[7];
    Xout[12] = tanh( Xout[12] );

    /* Generating code for PE 5 in layer 3 */
    Xout[13]  = (float)(-2.9059293) + (float)(-1.8452264) * Xout[2] +
        (float)(0.797831) * Xout[3] + (float)(-0.35911021) * Xout[4] +
        (float)(0.069089115) * Xout[5] + (float)(-0.11804507) * Xout[6] +
        (float)(0.099331737) * Xout[7];
    Xout[13] = tanh( Xout[13] );

    /* Generating code for PE 6 in layer 3 */
    Xout[14]  = (float)(2.4999413) + (float)(1.2057008) * Xout[2] +
        (float)(-0.27926686) * Xout[3] + (float)(0.14837676) * Xout[4] +
        (float)(0.087940931) * Xout[5] + (float)(0.26495889) * Xout[6] +
        (float)(0.02565472) * Xout[7];
    Xout[14] = tanh( Xout[14] );

    /* Generating code for PE 7 in layer 3 */
    Xout[15]  = (float)(2.5714412) + (float)(1.2578217) * Xout[2] +
        (float)(-0.73493344) * Xout[3] + (float)(0.1693646) * Xout[4] +
        (float)(0.15939854) * Xout[5] + (float)(0.35849226) * Xout[6] +
        (float)(0.0770225) * Xout[7];
    Xout[15] = tanh( Xout[15] );

    /* Generating code for PE 8 in layer 3 */
    Xout[16]  = (float)(-1.7281979) + (float)(-0.54740685) * Xout[2] +
        (float)(0.13327992) * Xout[3] + (float)(-0.26809275) * Xout[4] +
        (float)(-0.10516206) * Xout[5] + (float)(-0.1878498) * Xout[6] +
        (float)(0.060528226) * Xout[7];
    Xout[16] = tanh( Xout[16] );

    /* Generating code for PE 9 in layer 3 */
    Xout[17]  = (float)(3.9927492) + (float)(2.8859453) * Xout[2] +
        (float)(-1.3116106) * Xout[3] + (float)(0.65250283) * Xout[4] +
        (float)(-0.12934597) * Xout[5] + (float)(0.48706767) * Xout[6] +
        (float)(0.033722606) * Xout[7];
    Xout[17] = tanh( Xout[17] );

    /* Generating code for PE 0 in layer 4 */
    Xout[18]  = (float)(-0.91141182) + (float)(3.6300123) * Xout[8] +
        (float)(3.9339242) * Xout[9] + (float)(4.8950758) * Xout[10] +
        (float)(6.7815599) * Xout[11] + (float)(-7.5799356) * Xout[12] +
        (float)(-3.5039601) * Xout[13] + (float)(-3.7866173) * Xout[14] +
        (float)(-4.0687246) * Xout[15] + (float)(-5.873127) * Xout[16] +
        (float)(4.0936308) * Xout[17];
    Xout[18] = tanh( Xout[18] );

    /* De-scale and write output from network */
    Yout[0] = Xout[18] * (5.7121944) + (59.060925);
    return( 0 );
}

/* Wed Nov 03 15:35:43 1993 (pre.c) *//* Recall-Only Run-time for <press> */
/* Control Strategy is: <bkpfast> */

#if __STDC__
#define  ARGS(x) x
#else
#define  ARGS(x) ()
#endif /* __STDC__ */

/* --- External Routines --- */
extern double tanh  ARGS((double));
/* *** MAKE SURE TO LINK IN YOUR COMPILER's MATH LIBRARIES *** */

#if __STDC__
int press( void *NetPtr, float Yin[5], float Yout[1] )
#else
int press( NetPtr, Yin, Yout )
void  *NetPtr; /* Network Pointer (not used) */
float Yin[5], Yout[1];  /* Data */
#endif /* __STDC__ */
{
```

```
    float  Xout[18];  /* work arrays */
    long   ICmpT;  /* temp for comparisons */

    /* *** WARNING: Code generated assuming Recall = 0 *** */

    /* Read and scale input into network */
    Xout[2] = Yin[0] * (0.031825828) + (-18.827435);
    Xout[3] = Yin[1] * (0.3618934) + (-201.43719);
    Xout[4] = Yin[2] * (0.022922898) + (-1.0002407);
    Xout[5] = Yin[3] * (0.46774761) + (-39.225541);
    Xout[6] = Yin[4] * (0.5360986) + (-1.0257735);
LAB110:

    /* Generating code for PE 0 in layer 3 */
    Xout[7]  = (float)(-0.00085806672) + (float)(-0.097729363) * Xout[2] +
        (float)(0.0012888834) * Xout[3] + (float)(0.083640642) * Xout[4] +
        (float)(0.049953058) * Xout[5] + (float)(-0.032351527) * Xout[6];
    Xout[7] = tanh( Xout[7] );

    /* Generating code for PE 1 in layer 3 */
    Xout[8]  = (float)(-0.15920037) + (float)(0.30350038) * Xout[2] +
        (float)(-0.24113135) * Xout[3] + (float)(0.038593631) * Xout[4] +
        (float)(0.028781993) * Xout[5] + (float)(0.097850241) * Xout[6];
    Xout[8] = tanh( Xout[8] );

    /* Generating code for PE 2 in layer 3 */
    Xout[9]  = (float)(0.058014911) + (float)(-0.11937203) * Xout[2] +
        (float)(0.17020981) * Xout[3] + (float)(-0.020569507) * Xout[4] +
        (float)(-0.014189838) * Xout[5] + (float)(-0.051654916) * Xout[6];
    Xout[9] = tanh( Xout[9] );

    /* Generating code for PE 3 in layer 3 */
    Xout[10]  = (float)(0.010632542) + (float)(-0.28734604) * Xout[2] +
        (float)(0.068643942) * Xout[3] + (float)(-0.024041427) * Xout[4] +
        (float)(0.016147269) * Xout[5] + (float)(-0.039197724) * Xout[6];
    Xout[10] = tanh( Xout[10] );

    /* Generating code for PE 4 in layer 3 */
    Xout[11]  = (float)(0.057009269) + (float)(0.28038415) * Xout[2] +
        (float)(-0.050360292) * Xout[3] + (float)(0.02501322) * Xout[4] +
        (float)(-0.084106296) * Xout[5] + (float)(0.14954394) * Xout[6];
    Xout[11] = tanh( Xout[11] );

    /* Generating code for PE 5 in layer 3 */
    Xout[12]  = (float)(-0.013153192) + (float)(0.19470169) * Xout[2] +
        (float)(-0.055042781) * Xout[3] + (float)(0.15261218) * Xout[4] +
        (float)(0.03364072) * Xout[5] + (float)(0.21617027) * Xout[6];
    Xout[12] = tanh( Xout[12] );

    /* Generating code for PE 6 in layer 3 */
    Xout[13]  = (float)(-0.0086213844) + (float)(-0.10911887) * Xout[2] +
        (float)(0.19086374) * Xout[3] + (float)(-0.073111743) * Xout[4] +
        (float)(0.01840773) * Xout[5] + (float)(-0.10049289) * Xout[6];
    Xout[13] = tanh( Xout[13] );

    /* Generating code for PE 7 in layer 3 */
    Xout[14]  = (float)(0.0012865434) + (float)(-0.0052713477) * Xout[2] +
        (float)(-0.15658161) * Xout[3] + (float)(-0.029851872) * Xout[4] +
        (float)(-0.021224353) * Xout[5] + (float)(0.11160629) * Xout[6];
    Xout[14] = tanh( Xout[14] );

    /* Generating code for PE 8 in layer 3 */
    Xout[15]  = (float)(-0.14632934) + (float)(0.33095348) * Xout[2] +
        (float)(-0.29782265) * Xout[3] + (float)(0.01048635) * Xout[4] +
        (float)(-0.0622917) * Xout[5] + (float)(0.14853939) * Xout[6];
    Xout[15] = tanh( Xout[15] );

    /* Generating code for PE 9 in layer 3 */
    Xout[16]  = (float)(0.049729746) + (float)(-0.21525031) * Xout[2] +
        (float)(0.1495647) * Xout[3] + (float)(-0.013422946) * Xout[4] +
        (float)(-0.12482578) * Xout[5] + (float)(0.03112608) * Xout[6];
    Xout[16] = tanh( Xout[16] );

    /* Generating code for PE 0 in layer 4 */
    Xout[17]  = (float)(-0.22126342) + (float)(0.056802616) * Xout[7] +
        (float)(-0.39728552) * Xout[8] + (float)(0.19744696) * Xout[9] +
        (float)(0.29320255) * Xout[10] + (float)(-0.30782503) * Xout[11] +
        (float)(-0.31775162) * Xout[12] + (float)(0.25342625) * Xout[13] +
```

```
        (float)(-0.11370626) * Xout[14] + (float)(-0.46042106) * Xout[15] +
        (float)(0.27106291) * Xout[16];
    Xout[17] = tanh( Xout[17] );

    /* De-scale and write output from network */
    Yout[0] = Xout[17] * (102.0557) + (1012.6775);
    return( 0 );
}
```

APPENDIX D

Code Listing for Kalman Filtering Technique

```
SUBROUTINE FEX3 (T, Y, YDOT)

        implicit real*8 (d)

        DOUBLE PRECISION T, Y, YDOT,u1,kkk

        DIMENSION Y(24), YDOT(24)
c       common/ali1/u1,kkk
c
        real*8 tsam,a1,a2,a3,p0,puv0,pin0,pr0,pdv0,pdis,puvd0,wmfp00,q0
        real*8 effp,wsg00,i,ltt,kp,tau,area,hin0,hout0,kr1,kr2,kl1,kl2,lsp,lset
        real*8 psuc,tau1,tgo,npump0,wf0,tkick,incl,intpi,intfi,inwpi,inwfi,tmax
        real*8 densm,densw,densr0,densd,densdw,densg0,denss,densb0,n
        real*8 do,di,l,ls10,ar,adw,ad,afs,lr,ldw0,ld,vp,vs,vr,vdr
        real*8 thetai,tpix,tp10,tp20,tp40,tpo0,tm10,tm20,tm40
        real*8 tdw0,td0,tsat0,tfix,tfw,tp30,tm30,tfi0,hf,hfg,vf
        real*8 vfg,xe0,k1,k2,k3.k4,k5,k6,k7,x1,x2,x3,x4,x5,x6,hi,hos
        real*8 hob,kth,cp1,cp2,cm,wfi0,wpix,w10,clx,cd,tou,tou1,tou2,g1,g2,gv
        real*8 wnv,ztv,v0,u0,w0,r0,m0,pi,rho1,wmftp0,kr,i0r,phd0,h0,pdis0
        real*8 f1,f2,f3,kv,fv,hout,w20,w30,w40,ls20,mm,mm1,mm4,mm2
        real*8 mm3,sm,sms1,sms2,sms3,sms4,spm1,spm2,spm3,spm4,pr1
        real*8 pr2,dm,ap,mp,mp1,mp2,mp3,mp4,ms1,upm,ums1,ums2,vpi,mpi,mpo,md
        real*8 hb0,hxe0,lb0,c1,wr,xldw0,xwst0,xp0,txde10,xtfi,tfi,thpi
        real*8 wpi,tpi0,cl,wst,den1,den2,k(27),pr(16),aux(10),w(4),afwv0,fwcont
        real*8 delps,gain1,gain2,gain3,gain4,lsets,puvds,puvs,phds,pdiss
        real*8 hs,nfs1,npump1,dum1,arv1,nf,afwv,wfi,phd
        real*8 puv,pdv,deltap,h,xpt,xpump,wmfpt,wfis,afwvs
        real*8 tpi,tp1,tp2,tp3,tp4,tpo,tm1,tm2,tm3,tm4,densb,densr,ls1,xe,ldw
        real*8 tdw,p,td,wf,puvd,npump,dtpi,dtp1,dtp2,dtp3,dtp4,dtpo,dtm1
        real*8 dtm2,dtm3,dtm4,ddensb,ddensr,dls1,dxe,dldw
        real*8 dtdw,dp,dtd,dwf,dpuvd,dnpump,lll,econtr(2),auto(2)


        real*8 xlset,xldw,xl1,dx12,dx13,ta13,ka13,bxst,bxwf,xst,xwf,dx14,x13
        real*8 ka14,x14,x14a,x14b,x15,x12,x16,afwvb,kfin
        real*8 x120,x130,x140,ta12,ta14
        common /ali33/x120,x130,x140,ta12,ta14
        common /ali31/ xldw,xl1,dx12,dx13,ta13,ka13,bxst,bxwf,xst,xwf,dx14,x13
        common /ali32/ xlset,ka14,x14,x14a,x14b,x15,x12,x16,afwvb,kfin
        common /ali01/
     +         tsam,a1,a2,a3,p0,puv0,pin0,pr0,pdv0,pdis,puvd0,wmfp00,q0
        common /ali02/
     +         effp,wsg00,i,ltt,kp,tau,area,hin0,hout0,kr1,kr2,kl1,kl2,lsp,lset
        common /ali00/
     +         psuc,tau1,tgo,npump0,wf0,tkick,incl,intpi,intfi,inwpi,inwfi,tmax
        common /ali03/ densm,densw,densr0,densd,densdw,densg0,denss,densb0,n
        common /ali04/ do,di,ls10,ar,adw,ad,afs,lr,ldw0,ld,vp,vs,vr,vdr
        common /ali05/ thetai,tpix,tp10,tp20,tp40,tpo0,tm10,tm20,tm40
        common /ali06/ tdw0,td0,tsat0,tfix,tfw,tp30,tm30,tfi0,hf,hfg,vf
        common /ali07/ vfg,xe0,k1,k2,k3,k4,k5,k6,k7,x1,x2,x3,x4,x5,x6,hi,hos
        common /ali08/ hob,kth,cp1,cp2,cm,wfi0,wpix
         common /deli01/ w10,clx,cd,tou,tou1,tou2,g1,g2,gv
        common /ali09/ wnv,ztv,v0,u0,w0,r0,m0,pi,rho1
        common /deli02/ wmftp0,kr,i0r,phd0,h0,pdis0
        common /ali10/ f1,f2,f3,kv,fv,hout,w20,w30,w40,ls20,mm,mm1,mm4,mm2
        common /ali11/ mm3,sm,sms1,sms2,sms3,sms4,spm1,spm2,spm3,spm4,pr1
        common /ali12/ pr2,dm,ap,mp,mp1,mp2,mp3,mp4,ms1,upm
         common /deli03/ums1,ums2,vpi,mpi,mpo,md,thpi
        common /ali13/ hb0,hxe0,lb0,c1,wr,xldw0,xwst0,xp0,txde10,xtfi,tfi
        common /ali14/ wpi,tpi0,cl,wst,den1,den2
         common /deli04/ k,pr,aux,w,afwv0,fwcont
        common /ali15/
     +         delps,gain1,gain2,gain3,gain4,lsets,puvds,puvs,phds,pdiss
        common /ali16/ hs,nfs1,npump1,dum1,arv1,nf,afwv,wfi,phd
        common /ali17/ puv,pdv,deltap,h,xpt,xpump,wmfpt,wfis,afwvs
        common /ali18/
     +         tpi,tp1,tp2,tp3,tp4,tpo,tm1,tm2,tm3,tm4,densb,densr,ls1,xe,ldw
```

```
      common /ali18/
     +        tpi,tp1,tp2,tp3,tp4,tpo,tm1,tm2,tm3,tm4,densb,densr,ls1,xe,ldw
      common /ali19/
     +        tdw,p,td,wf,puvd,npump,dtpi,dtp1,dtp2,dtp3,dtp4,dtpo,dtm1
      common /ali20/ dtm2,dtm3,dtm4,ddensb,ddensr,dls1,dxe,dldw
      common /ali21/ dtdw,dp,dtd,dwf,dpuvd,dnpump,lll,econtr,auto

c       u1=-.05d0*y(1)+.01d0*y(2)+kkk
c        YDOT(1) = -.05D0*Y(1) + .01d0*Y(2)
c        YDOT(2) = .3d0*Y(2)-2.0d0*Y(2)
c-------------------------------------------------------------------

cccccccccccccccccccccccccccccccccccccccc
        tpi=y(1)
        tp1=y(2)
        tp2=y(3)
        Tp3=y(4)
        tp4=y(5)
        tpo=y(6)
        tm1=y(7)
        ls1=y(8)
        tm2=y(9)
        tm3=y(10)
        tm4=y(11)
        xe=y(12)
        p=y(13)
        densb=y(14)
        densr=y(15)
        ldw=y(16)
        tdw=y(17)
        td=y(18)
        puvd=y(19)
        Npump=y(20)
        wf=y(21)

        x12=y(22)
        x13=y(23)
        x14=y(24)

        l=lll


c
c State equation #1
c
        dtpi = 1./thpi *(thetai-tpi)



        DTpiTpi=1.-DELTAT/thpi

cccccccc All other partial derivatives are zero

        ydot(1)=dtpi
c-------------------------------------------------------------------

c
c Sate Equation #2
c

        DTp1 = Wpi*Tpi/(DENSw*Ap*Ls1)-(Wpi/(DENSw*Ap*Ls1)+
     +  (Upm*Spm1)/(Mp1*Cp1) )*Tp1+( (Upm*Spm1)/(Mp1*Cp1) )*Tm1
        ydot(2)=dtp1
c+++++++++++++++++++++++++++++++++++

        DTp1Tpi= DELTAT*Wpi/(DENSw*Ap*ls1)
        DTp1Tp1= 1.+DELTAT*(Wpi/(DENSw*Ap*Ls1)+(Upm*Spm1)/(Mp1*Cp1))
        DTp1Tm1= DELTAT*(Upm*Spm1)/(Mp1*Cp1)
        DTp1Ls1= DELTAT*(Wpi/(DENSw*Ap)-Wpi*Tpi/(DENSw*Ap))*(ls1**(-2.0))

cccccccc All other partial derivatives are zero

c++-++++++++++++++++++++++++++++++++
c       wf=wf0

        W(1)    = C1*(( DENSd*(Ldw+Ld-Ls1)-(L-Ls1)*DENSb-Lr*DENSr)**.5)/12.

        Dw1Ls1= C1*(( DENSd*(Ldw+Ld-Ls1)-(L-Ls1)*DENSb-Lr*DENSr)**(-.5))/24.*
```

```
     +           (DENSb-DENSd)
      Dw1DENSb= C1*(( DENSd*(Ldw+Ld-Ls1)-(L-Ls1)*DENSb-Lr*DENSr)**(-.5))/24.*
     +           (Ls1-L)

      Dw1DENSr= C1*(( DENSd*(Ldw+Ld-Ls1)-(L-Ls1)*DENSb-Lr*DENSr)**(-.5))/24.*
     +           (-Lr)


       AUX(7)   = Md*(Tdw - Td )/W(1)

cc
      DAux7Ls1= -Dw1Ls1*Aux(7)/W(1)
      DAux7Densb= -Dw1Densb*Aux(7)/W(1)
      DAux7Densr= -Dw1Densr*Aux(7)/W(1)
      DAux7Tdw= Md/W(1)
      DAux7Td= -Md/W(1)
cc

       DEN1   =  (Afs*DENSs*Cp2*(Td +X1+K5*P)/2.)
cc
      DDen1Td= Afs*DENSs*Cp2/2.
      DDen1P= Afs*DENSs*Cp2*K5/2.
cc


       DEN2   =   (DENSb*Afs*(L-Ls1)*(X5+K4*P)/2.)

cc
      DDen2Densb= Afs*(L-Ls1)*(X5+K4*P)/2.
      DDen2Ls1= DENSb*Afs*(X5+K4*P)/2.
      DDen2P= Afs*(L-Ls1)*K4/2.
cc
       K(01)    =-(K1 + K2*Xe/2.)/(((X2+K1*P) + Xe*(X3 + K2*P)/2.)**2.)
cc

      DK01P= 2*(K1 + K2*Xe/2.)/(((X2+K1*P) + Xe*(X3 + K2*P)/2.)**3.) *
     +         (K1+Xe*K2/2)
      DK01Xe= 2*(K1 + K2*Xe/2.)/(((X2+K1*P) + Xe*(X3 + K2*P)/2.)**3.) *
     +         (X3 + K2*P)/2. - (K2/2.)/
     +         (((X2+K1*P) + Xe*(X3 + K2*P)/2.)**2.)

cc

       K(02)    =-(X3+ P*K2)/2*(((X2+K1*P) + Xe*(X3 + K2*P)/2. )**2.)

cc
       DK02P= -K2/2.*(((X2+K1*P) + Xe*(X3 + K2*P)/2. )**2.) -
     +         (X3+ P*K2)*((X2+K1*P) + Xe*(X3 + K2*P)/2. )*(K1+Xe*K2/2.)
      DK02Xe= -(X3+ P*K2)*((X2+K1*P) + Xe*(X3 + K2*P)/2. )*
     +          (X3 + K2*P)/2.
cc


       K(03)    =-(K1 + K2*Xe)/(((X2+K1*P) + Xe*(X3 + K2*P) )**2.)
cc
      DK03P= 2*(K1 + K2*Xe)/(((X2+K1*P) + Xe*(X3 + K2*P) )**3.)*
     +         (K1+Xe*K2)
      DK03Xe= -K2/(((X2+K1*P) + Xe*(X3 + K2*P) )**2.) +
     +          2*(K1 + K2*Xe)/(((X2+K1*P) + Xe*(X3 + K2*P) )**3.)*
     +          (X3 + K2*P)
cc


       K(04)    =-(X3+P*K2)/( ( (X2+K1*P) + Xe*(X3 + K2*P) )**2.)
cc
      DK04P= -K2/( ( (X2+K1*P) + Xe*(X3 + K2*P) )**2.) +
     +         2*(X3+P*K2)/( ( (X2+K1*P) + Xe*(X3 + K2*P) )**3.)*
     +         (K1+Xe*K2)
      DK04Xe= 2*(X3+P*K2)/( ( (X2+K1*P) + Xe*(X3 + K2*P) )**3.)*
     +          (X3 + K2*P)
cc

       K(05)    =-DENSs*Afs
       K(06)    =( Ums1*Pr2*Ls1*(Tm1+Tm4-Td-X1-K5*P)+W(1)*Cp2*Td-
     +             Afs*DENSs*Ls1*Cp2*AUX(7)/2.)/DEN1

cc
      DK06Ls1= ( Ums1*Pr2*(Tm1+Tm4-Td-X1-K5*P) + DW1Ls1*Cp2*Td-
```

```
     +           (Afs*DENSs*Cp2*AUX(7)+Afs*DENSs*Ls1*Cp2*DAUX7Ls1)/2.)
     +           /DEN1
      DK06DENSb= (DW1DENSb*Cp2*Td-Afs*DENSs*Ls1*Cp2*DAUX7DENSb/2.)
     +          /DEN1
      DK06DENSr= (DW1DENSr*Cp2*Td-Afs*DENSs*Ls1*Cp2*DAUX7DENSr/2.)
     +          /DEN1
      DK06Tm1= Ums1*Pr2*Ls1/DEN1
      DK06Tm4= Ums1*Pr2*Ls1/DEN1
      DK06P= -Ums1*Pr2*Ls1*K5/DEN1- K(06)*DDen1P/DEN1
      DK06Tdw= (DW1Tdw*Cp2*Td-Afs*DENSs*Ls1*Cp2*DAUX7Tdw/2.)
     +          /DEN1
      DK06Td= (-Ums1*Pr2 + W(1)*Cp2 + Td*Cp2*DW1Td -
     +         Afs*DENSs*Ls1*Cp2*DAUX7Td/2.)/DEN1- K(06)*DDenTd/DEN1
cc

       K(07)   =-Cp2*(X1+K5*P)/DEN1
cc
      DK07P= -Cp2*K5/DEN1 -K(07)*DDEN1P/DEN1
      DK07Td= -K(07)*DDEN1Td/DEN1
cc


       K(08)   =-Afs*DENSs*Ls1*Cp2*K5/DEN1
cc
cc
      DK08Ls1= -Afs*DENSs*Cp2*K5/DEN1
      DK08Td= -K(08)*DDEN1Td/DEN1
      DK08P= -K(08)*DDEN1P/DEN1
cc

       K(09)   =- Afs*(L-Ls1)
cc
      DK09Ls1= Afs
cc


       K(10)  =Afs*DENSb
cc
      DDK10DENSb= Afs
cc
       K(11)  =( Ums2*Pr2*(L-Ls1)*( Tm2+Tm3-2*(X1+K5*P) ) )/DEN2
cc
      DK11Ls1= -Ums2*Pr2*( Tm2+Tm3-2*(X1+K5*P) )/DEN2 -
     +           K(11)*DDEN2Ls1/DEN2
      DK11Tm2= Ums2*Pr2*(L-Ls1)/DEN2
      DK11TM3= Ums2*Pr2*(L-Ls1)/DEN2
      DK11P=  -Ums2*Pr2*(L-Ls1)*2*K5/DEN2 - K(11)*DDEN2P/DEN2
      DK11DENSb= -K(11)*DDEN2DENSb/DEN2
cc

       K(12)  =(X4+K3*P)/DEN2
cc
      DK12P= K3/DEN2- K(12)*DDEN2P/DEN2
      DK12Ls1= -K(12)*DDEN2Ls1/DEN2
      DK12DENSb= K(12)*DDENSb/DEN2
cc

       K(13)  =-( X4+K3*P + Xe*(X5 + K4*P) ) /DEN2

cc
      DK13P= K(13)*DDEN2P/DEN2 -(K3 + Xe*K4)/DEN2
      DK13Xe= -(X5 + K4*P)/DEN2
      DK13Ls1= -K(13)*DDEN2Ls1/DEN2
      DK13DENSb= -K(13)*DDEN2DENSb/DEN2
cc

       K(14)  =-Afs*(L-Ls1)*( X4+K3*P + Xe*(X5 + K4*P)/2. )/DEN2
cc
      DK14Ls1= Afs*( X4+K3*P + Xe*(X5 + K4*P)/2. )/DEN2 -
     +           K(14)*DDEN2Ls1/DEN2
      DK14P= -Afs*(L-Ls1)*(K3 + Xe*K4/2.)/DEN2 -
     +           K(14)*DDEN2P/DEN2
      DK14DENSb= -K(14)*DDEN2DENSb/DEN2
      DK14Xe= -Afs*(L-Ls1)*((X5 + K4*P)/2. )/DEN2
cc

       K(15)  =Afs*DENSb*( X4+K3*P + Xe*(X5 + K4*P)/2. )/DEN2
cc
```

```
      DK15DENSb= K(15)/DENSb - K(15)*DDEN2DENSb/DEN2
      DK15P= Afs*DENSb*( K3*P + Xe*K4*P/2. )/DEN2 -
     +         K(15)*DDEN2P/DEN2
      DK15Xe= Afs*DENSb*(X5 + K4*P)/2./DEN2
      DK15Ls1= -K(15)*DDEN2Ls1/DEN2
cc
       K(16)   = -Afs*(L-Ls1)*(K3 + Xe*K4/2.)/DEN2
cc
      DK16Ls1= Afs*(K3 + Xe*K4/2.)/DEN2 - K(16)*DDEN2Ls1/DEN2
      DK16DENSb= -K(16)*DDEN2DENSb/DEN2
      DK16P= -K(16)*DDEN2P/DEN2
      DK16Xe= -Afs*(L-Ls1)*K4/2./DEN2
cc
       K(17)   =- Vr

       K(18)   =wf/( Adw*DENSdw)
cc
      DK18wf= K(18)/wf
cc


       K(19)   =(1-Xe)/( Adw*DENSdw)
cc
      DK19Xe= -1/( Adw*DENSdw)
cc


       K(20)   =-W(1)/( Adw*DENSdw)
cc
      DK20Ls1= -Dw1Ls1/( Adw*DENSdw)
      DK20DENSb= -Dw1DENSb/( Adw*DENSdw)
      DK20DENSr= -Dw1DENSr/( Adw*DENSdw)
cc
       K(21)   =wf*Tfi/( Adw*DENSdw*Ldw)
cc
      DK21wf= K(21)/wf
      DK21Ldw= -K(21)/Ldw
cc

       K(22)   =(1-Xe)*(X1+K5*P)/( Adw*DENSdw*Ldw)
cc
      DK22Xe= - (X1+K5*P)/( Adw*DENSdw*Ldw)
      DK22P= (1-Xe)*K5/( Adw*DENSdw*Ldw)
      DK22Ldw= -K(22)/Ldw
cc

       K(23)   =-W(1)*Td/( Adw*DENSdw*Ldw)
cc
      DK23Ls1= -Dw1Ls1*Td/( Adw*DENSdw*Ldw)
      DK23DENSb= -Dw1DENSb*Td/( Adw*DENSdw*Ldw)
      DK23DENSr= -Dw1DENSr*Td/( Adw*DENSdw*Ldw)
      DK23Ldw= -K(23)/Ldw
cc

       K(24)   =-Tdw*DENSdw*Adw/( Adw*DENSdw*Ldw)
cc
      DK24Tdw= K(24)/Tdw
      DK24Ldw= -K(24)/Ldw
cc

       K(25)   =Xe/(K7*(Vdr-Adw*Ldw) )
cc
      DK25Xe= K(25)/Xe
      DK25Ldw= K(25)*Adw/(Vdr-Adw*Ldw)
cc

       K(26)   =-C1*P/(K7*(Vdr-Adw*Ldw) )
cc
      DK26P= K(26)/P
      DK26Ldw= K(26)*Adw/(Vdr-Adw*Ldw)
cc

       K(27)   =(X6 + K7*P)*Adw/(K7*(Vdr-Adw*Ldw) )
cc
      DK27P= K7*Adw/(K7*(Vdr-Adw*Ldw) )
      DK27Ldw= K(27)*Adw/(Vdr-Adw*Ldw)
```

```
cc
       pr(1)     =K(18)+K(19)*W(1)+K(20)
cc
      Dpr1wf= DK18wf
      Dpr1Xe= DK19Xe*W(1)
      Dpr1Ls1= Dw1Ls1*K(19) + DK20Ls1
      Dpr1DENSb= Dw1DENSb*K(19) + DK20DENSb
      Dpr1DENSr= Dw1DENSr*K(19) + DK20DENSr
cc

       pr(2)     =K(19)*(K(5)+K(10))
cc
      Dpr2Xe= K(5)+K(10)
      Dpr2DENSb= K(19)*DK10DENSb
cc

       pr(3)     =(K(17)*K(3)+K(9)*K(1))*K(19)
cc
      Dpr3Ls1= DK09Ls1*K(1)*K(19)
      Dpr3P= (K(17)*DK03P+K(9)*DK01P)*K(19)
      Dpr3Xe= DK19Xe*(K(17)*DK03P+K(9)*DK01P) +
     +          K(19)*(DK03Xe*K(17)+DK01Xe*K(9))
cc

       pr(4)     =(K(17)*K(4)+K(9)*K(2))*K(19)
      Dpr4Ls1= DK09Ls1*K(2)*K(19)
      Dpr4P= (DK04P*K(17)+DK02P*K(9))*K(19)
      Dpr4Xe= DK19Xe*(K(17)*K(4)+K(9)*K(2)) +
     +          K(19)*(K(17)*DK04Xe+K(9)*DK02Xe)
cc

       pr(5)     =K(21) + K(22)*W(1)+K(23)+K(24)*pr(1)
cc
      Dpr5wf= DK21wf +Dpr1wf*K(24)
      Dpr5Ls1= Dw1Ls1*K(22)+DK23Ls1+K(24)*Dpr1Ls1
      Dpr5DENSb= Dw1DENSb*K(22)+DK23DENSb+K(24)*Dpr1DENSb
      Dpr5DENSr= Dw1DENSr*K(22)+DK23DENSr+K(24)*Dpr1DENSr
      Dpr5Ldw= DK22Ldw*w(1)+DK23Ldw+DK24Ldw*pr(1)
      Dpr5Xe= DK22Xe*w(1)+K(24)*Dpr1Xe
      Dpr5P= DK22P*w(1)
      Dpr5Tdw= DK24Tdw*pr(1)
cc


       pr(6)     =K(22)*(K(5)+K(10))+K(24)*pr(2)
cc
      Dpr6Xe= DK22Xe*(K(5)+K(10))+K(24)*Dpr2Xe
      Dpr6P= DK22P*(K(5)+K(10))
      Dpr6Ldw= DK22Ldw*(K(5)+K(10))+DK24Ldw*pr(2)
      Dpr6DENSb= K(22)*DK10DENSb+K(24)*Dpr2DENSb
      Dpr6Tdw= DK24Tdw*pr(2)
cc

       pr(7)     =K(22)*(K(17)*K(3)+K(9)*K(1))+K(24)*pr(3)
cc
      Dpr7Xe= DK22Xe*(K(17)*K(3)+K(9)*K(1))+K(22)*(DK03Xe*K(17)+DK01Xe*K(9))
     +          +Dpr3Xe*K(24)
      Dpr7P= DK22P*(K(17)*K(3)+K(9)*K(1))+K(22)*(DK03P*K(17)+DK01P*K(9))
     +          +Dpr3P*K(24)
      Dpr7Ldw= Dk22Ldw*(K(17)*K(3)+K(9)*K(1))+DK24Ldw*pr(3)
      Dpr7Ls1= DK09Ls1*K(1)*K(22)+K(24)*Dpr3Ls1
      Dpr7Tdw= DK24Tdw*pr(3)
cc


       pr(8)     =(K(17)*K(4)+K(9)*K(2))*K(22)+K(24)*pr(4)
cc
      Dpr8P= (K(17)*DK04P+K(9)*DK02P)*K(22)+
     +          (K(17)*K(4)+K(9)*K(2))*DK22P+K(24)*Dpr4P
      Dpr8Xe= (K(17)*DK04Xe+K(9)*DK02Xe)*K(22)+
     +          (K(17)*K(4)+K(9)*K(2))*DK22Xe+K(24)*Dpr4Xe
      Dpr8Ls1= K(22)*K(2)*DK09Ls1+K(24)*Dpr4Ls1
      Dpr8Ldw= DK22Ldw*(K(17)*K(4)+K(9)*K(2))+DK24Ldw*pr(4)
      Dpr8Tdw= DK22Tdw*(K(17)*K(4)+K(9)*K(2))+DK24Tdw*pr(4)
cc

       pr(9)     =(K(25)*W(1)+K(26)+K(27)*pr(1))/
     +           (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))
```

```
cc
      Dpr9Xe=((DK25Xe*W(1)+K(27)*Dpr1Xe)*
     +        (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3)) -
     +        (K(25)*W(1)+K(26)+K(27)*pr(1))*
     +        (DK25Xe*(K(17)*K(3)+K(9)*K(1))+K(25)*(
     +         DK03Xe*K(17)+DK01Xe*K(9))-K(27)*Dpr3Xe))*
     +        (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))**(-2.)
      Dpr9Ldw=((DK25Ldw*W(1)+DK26Ldw+DK27Ldw*pr(1))*
     +        (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3)) -
     +        (K(25)*W(1)+K(26)+K(27)*pr(1))*
     +        (DK25Ldw*(K(17)*K(3)+K(9)*K(1))-DK27Ldw*pr(3)))*
     +        (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))**(-2.)
      Dpr9P= ((DK26P+DK27P*pr(1))*
     +        (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))+
     +        (K(25)*W(1)+K(26)+K(27)*pr(1))*
     +        (DK27P*pr(3)+K(27)*Dp3P))*
     +        (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))**(-2.)
      Dpr9DENSb=(K(25)*Dw1DENSb+K(27)*Dpr1DENSb)/
     +          (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))
      Dpr9DENSr=(K(25)*Dw1DENSr+K(27)*Dpr1DENSr)/
     +          (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))
      Dpr9Ls1=(K(25)*Dw1Ls1*
     +        (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))-
     +        (K(25)*W(1)+K(26)+K(27)*pr(1))*
     +        (K(1)*K(25)*DK9Ls1-K(27)*Dpr3Ls1))*
     +        (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))**(-2.)
      Dpr9wf=K(27)*Dpr1wf/
     +        (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))
cc

       pr(10)    =(K(25)*(K(5)+K(10))+K(27)*pr(2))/
     +          (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))
cc
      Dpr10Xe=((DK25Xe*(K(5)+K(10))+K(27)*Dpr2Xe)*
     +        (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))-
     +        (K(25)*(K(5)+K(10))+K(27)*pr(2))*
     +        (DK25Xe*(K(17)*K(3)+K(9)*K(1))+
     +        K(25)*(K(17)*DK03Xe+K(9)*DK01Xe)-K(27)*Dpr3Xe))*
     +        (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))**(-2.)
      Dpr10Ldw=((DK25Xe*(K(5)+K(10)))*
     +        (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))-
     +        (K(25)*(K(5)+K(10))+K(27)*pr(2))*
     +        (DK25Ldw*(K(17)*K(3)+K(9)*K(1))-DK27Ldw*pr(3)))*
     +        (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))**(-2.)
      Dpr10P=(DK27P*pr(2)*(1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))-
     +        (K(25)*(K(5)+K(10))+K(27)*pr(2))*
     +        (K(25)*(K(17)*DK03P+K(9)*DK01P)-DK27P*pr(3)))*
     +        (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))**(-2.)
      Dpr10DENSb=(DK10DENSb*K(25)+K(27)*Dpr2DENSb)/
     +          (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))
      Dpr10Ls1=(K(25)*(K(5)+K(10))+K(27)*pr(2))*
     +         ((K(25)*DK09Ls1-K(27))*Dpr3Ls1)*
     +        (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))**(-2.)
cc

       pr(11)    =((K(17)*K(4)+K(9)*K(2))*K(25)+K(27)*pr(4))/
     +          (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))
cc
      Dpr11Xe=(((K(17)*DK04Xe+K(9)*DK02Xe)*K(25)+
     +           DK25Xe*(K(17)*K(4)+K(9)*K(2))+K(27)*Dpr4Xe)*
     +          (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))-
     +          ((K(17)*K(4)+K(9)*K(2))*K(25)+K(27)*pr(4))*
     +          ((K(17)*DK03Xe+K(9)*DK01Xe)*K(25)+
     +          DK25Xe*(K(17)*K(3)+K(9)*K(1))-K(27)*Dpr3Xe))*
     +          (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))**(-2.)

      Dpr11P=(((K(17)*DK04P+K(9)*DK02P)*K(25)+DK27P*pr(4)+Dpr4P*K(27))*
     +          (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))-
     +          ((K(17)*K(4)+K(9)*K(2))*K(25)+K(27)*pr(4))*
     +          ((K(17)*DK03P+K(9)*DK01P)*K(25)
     +          -K(27)*Dpr3Xe-DK27P*pr(3)))*
     +          (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))**(-2.)

      Dpr11Ldw=(((DK25Ldw*(K(17)*K(4)+K(9)*K(2))+DK27Ldw*pr(4)))*
     +          (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))-
     +          ((K(17)*K(4)+K(9)*K(2))*K(25)+K(27)*pr(4))*
     +          (DK25Ldw*(K(17)*K(3)+K(9)*K(1))-DK27Ldw*pr(3)))*
     +          (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))**(-2.)
```

```
      Dpr11Ls1=(((DK09Ls1*K(2)*K(25)+K927)*Dpr4Ls1)*
     +           (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))-
     +          ((K(17)*K(4)+K(9)*K(2))*K(25)+K(27)*pr(4))*
     +          (DK09Ls1*K(1)*K(25)-K(27)*Dpr3Ls1))*
     +           (1-(K(17)*K(3)+K(9)*K(1))*K(25)-K(27)*pr(3))**(-2.)
cc
       pr(12)    =(K(6) + K(7)*W(1) )/(1-K(5)*K(7))
cc
      Dpr12Ls1= (DK06Ls1+K(7)*Dw1Ls1)/(1-K(5)*K(7))
      Dpr12DENSb= (DK06DENSb+K(7)*Dw1DENSb)/(1-K(5)*K(7))
      Dpr12DENSr= (DK06DENSr+K(7)*Dw1DENSr)/(1-K(5)*K(7))
      Dpr12Tdw=DK06Tdw/(1-K(5)*K(7))
      Dpr12Tm1=DK06Tm1/(1-K(5)*K(7))
      Dpr12Tm4=DK06Tm4/(1-K(5)*K(7))
      Dpr12P=(DK06P+DK07P*w(1))/(1-K(5)*K(7))-
     +        (K(6) + K(7)*W(1) )*(K(5)*DK07P)*(1-K(5)*K(7))**(-2.)
      Dpr12Td=(DK06Td+DK07Td*w(1))/(1-K(5)*K(7))-
     +        (K(6) + K(7)*W(1) )*(K(5)*DK07Td)*(1-K(5)*K(7))**(-2.)
cc


       pr(13)    =K(8)/(1-K(5)*K(7))
cc
      Dpr13Ls1=DK08Ls1/(1-K(5)*K(7))
      Dpr13Td=DK08Td/(1-K(5)*K(7))-K(8)*DK07Td*(1-K(5)*K(7))**(-2.)
      Dpr13P=DK08P/(1-K(5)*K(7))-K(8)*DK07P*(1-K(5)*K(7))**(-2.)
cc

       pr(14)    =(K(11)+(K(13)+K(12))*W(1))/(1-(K(2)*K(9)*K(13)+K(14)
     +            *K(2)))
cc
      Dpr14Tm2=DK11Tm2/(1-(K(2)*K(9)*K(13)+K(14)*K(2)))
      Dpr14Tm3=DK11Tm3/(1-(K(2)*K(9)*K(13)+K(14)*K(2)))
      Dpr14P=((DK11P+W(1)*(DK13P+DK12P))*(1-(K(2)*K(9)*K(13)+K(14)*K(2)))
     +        +(K(9)*(DK02P*K(13)+DK13P*K(2))+K(2)*DK14P+DK02P*K(13))*
     +        (K(11)+(K(13)+K(12))*W(1)))*
     +        (1-(K(2)*K(9)*K(13)+K(14)*K(2)))**(-2.)
      Dpr14DENSb=((DK11DENSb+DW1DENSb*(K(13)+K(12))+W(1)*
     +               (DK13DENSb+DK12DENSb))*
     +               (1-(K(2)*K(9)*K(13)+K(14)*K(2)))+
     +               (K(11)+(K(13)+K(12))*W(1))*
     +               (K(2)*K(9)*DK13DENSb+K(2)*DK14DENSb))*
     +               (1-(K(2)*K(9)*K(13)+K(14)*K(2)))**(-2.)
      Dpr14Ls1=((DK11Ls1+DW1Ls1*(K(13)+K(12))+W(1)*(DK13Ls1+DK12Ls1))*
     +               (1-(K(2)*K(9)*K(13)+K(14)*K(2)))+
     +               (K(11)+(K(13)+K(12))*W(1))*
     +               (K(2)*(DK09Ls1*K(13)+K(9)*DK13Ls1)+K(2)*DK14Ls1))*
     +               (1-(K(2)*K(9)*K(13)+K(14)*K(2)))**(-2.)
      Dpr14Xe=(W(1)*DK13Xe+(K(9)*(DK02Xe*K(13)+K(2)*DK13Xe)+K(2)*DK14Xe+
     +        DK02Xe*K(14)))*(1-(K(2)*K(9)*K(13)+K(14)*K(2)))**(-2.)
      Dpr14DENSr=DW1DENSr*(K(13)+K(12))/(1-(K(2)*K(9)*K(13)+K(14)
     +            *K(2)))


cc
       pr(15)    =(K(12)*K(5)-(K(5)+K(10))*K(13)+K(15))/
     +           (1-(K(2)*K(9)*K(13)+K(14)*K(2)))
cc
      Dpr15P=((DK12P*K(5)+DK13P*(K(5)+K(10))+DK15P)*
     +        (1-(K(2)*K(9)*K(13)+K(14)*K(2)))+
     +        (K(12)*K(5)+(K(5)+K(10))*K(13)+K(15))*
     +        (DK13P*K(2)*K(9)+K(14)*DK02P+DK14P*K(2)))*
     +          (1-(K(2)*K(9)*K(13)+K(14)*K(2)))**(-2.)
      Dpr15Ls1=((DK12Ls1*K(5)+DK13Ls1*(K(5)+K(10))+DK15Ls1)*
     +          (1-(K(2)*K(9)*K(13)+K(14)*K(2)))+
     +          (K(12)*K(5)+(K(5)+K(10))*K(13)+K(15))*
     +          (K(2)*(DK13P*K(9)+K(13)*DK09Ls1)+DK14Ls1*K(2)))*
     +          (1-(K(2)*K(9)*K(13)+K(14)*K(2)))**(-2.)
      Dpr15DENSb=((DK12DENSb*K(5)+DK13DENSb*(K(5)+K(10))+
     +              DK10DENSb*K(13)+DK15DENSb)*
     +          (1-(K(2)*K(9)*K(13)+K(14)*K(2)))+
     +         (K(12)*K(5)+(K(5)+K(10))*K(13)+K(15))*
     +          (K(2)*K(9)*DK13DENSb+DK14DENSb*K(2)))*
     +          (1-(K(2)*K(9)*K(13)+K(14)*K(2)))**(-2.)
      Dpr15Xe=((DK13Xe*(K(5)+K(10))+DK15Xe)*
     +        (1-(K(2)*K(9)*K(13)+K(14)*K(2)))+
     +        (K(12)*K(5)+(K(5)+K(10))*K(13)+K(15))*
     +        (K(2)*K(9)*DK13Xe+DK14Xe*K(2)))*
```

```
+           (1-(K(2)*K(9)*K(13)+K(14)*K(2)))**(-2.)

      pr(16)    =(K(1)*K(9)*K(13)+K(14)*K(1)+K(16))/
+           (1-(K(2)*K(9)*K(13)+K(14)*K(2)))
      Dpr16P=((DK01P*K(9)*K(13)+K(1)*K(9)*DK13P+
+          DK14P*K(1)+DK01P*K(14)+DK16P)*
+         (1-(K(2)*K(9)*K(13)+K(14)*K(2)))+
+         (K(1)*K(9)*K(13)+K(14)*K(1)+K(16))*
+         (DK02P*K(9)*K(13)+K(2)*DK13P*K(9)+DK14P*K(2)+
+         DK02P*K(14)))*
+         (1-(K(2)*K(9)*K(13)+K(14)*K(2)))**(-2.)

      Dpr16Xe=((DK01Xe*K(9)*K(13)+K(1)*K(9)*DK13Xe+
+          DK14Xe*K(1)+DK01Xe*K(14)+DK16Xe)*
+         (1-(K(2)*K(9)*K(13)+K(14)*K(2)))+
+         (K(1)*K(9)*K(13)+K(14)*K(1)+K(16))*
+         (DK02Xe*K(9)*K(13)+K(2)*DK13Xe*K(9)+DK14Xe*K(2)+
+         DK02Xe*K(14)))*
+         (1-(K(2)*K(9)*K(13)+K(14)*K(2)))**(-2.)

      Dpr16Ls1=((K(1)*(DK09Ls1*K(13)+DK13Ls1*K(9))+K(1)*DK14Ls1+
+          DK16Ls1)*(1-(K(2)*K(9)*K(13)+K(14)*K(2)))+
+          (K(1)*K(9)*K(13)+K(14)*K(1)+K(16))*
+          (K(2)*(DK09LS1*K(13)+K(9)*DK13LS1)+K(2)*DK14LS1))*
+         (1-(K(2)*K(9)*K(13)+K(14)*K(2)))**(-2.)

      Dpr16DENSb=((K(1)*K(9)*DK13DENSb+DK14DENSb*K(1)+DK16DENSb)*
+             (1-(K(2)*K(9)*K(13)+K(14)*K(2)))+
+             (K(1)*K(9)*K(13)+K(14)*K(1)+K(16))*
+             (K(2)*K(9)*DK13DENSb+DK14DENSb*K(2)))*
+             (1-(K(2)*K(9)*K(13)+K(14)*K(2)))**(-2.)

cc

      AUX(1)   =(pr(12)+pr(13)*pr(9)+(pr(13)*pr(11)*(pr(14)+pr(16)
+           *pr(9))/(1-pr(16)*pr(11))))/((1-pr(10)*pr(13))-
+           (pr(13)*pr(11)*(pr(15)+pr(16)*pr(10)))/
+           (1-pr(16)*pr(11)))

cc
      DAUX1Tm1=Dpr12Tm1/((1-pr(10)*pr(13))-
+           (pr(13)*pr(11)*(pr(15)+pr(16)*pr(10)))/
+           (1-pr(16)*pr(11)))
      DAUX1Tm2=pr(13)*pr(11)*Dpr14Tm2/((1-pr(10)*pr(13))-
+           (pr(13)*pr(11)*(pr(15)+pr(16)*pr(10)))/
+           (1-pr(16)*pr(11)))
      DAUX1Tm3=pr(13)*pr(11)*Dpr14Tm3/((1-pr(10)*pr(13))-
+           (pr(13)*pr(11)*(pr(15)+pr(16)*pr(10)))/
+           (1-pr(16)*pr(11)))
      DAUX1Tm4=Dpr12Tm4/((1-pr(10)*pr(13))-
+           (pr(13)*pr(11)*(pr(15)+pr(16)*pr(10)))/
+           (1-pr(16)*pr(11)))
c     DAUX1Ls1=(Dpr12Ls1+Dpr13Ls1*pr(9)+pr(13)*Dpr9Ls1+
c    +           ((Dpr13Ls1*pr(11)*(pr(14)+pr(16)*pr(9))+
c    +            pr(13)*(Dpr11Ls1*(pr(14)+pr(16)*pr(9))+
c    +            pr(11)*(Dpr14Ls1+Dpr16Ls1*pr(9)+Dpr9Ls1*
c    +            pr(16))))*(1-pr(16)*pr(11))+
c    +            pr(13)*pr(11)*(pr(14)+pr(16)*pr(9))*
c    +            (Dpr16Ls1*pr(11)+Dpr11Ls1*pr(16)))*
c    +            (1-pr(16)*pr(11))**(-2.))*((1-pr(10)*pr(13))-
c    +            (pr(13)*pr(11)*(pr(15)+pr(16)*pr(10)))/
c    +            (1-pr(16)*pr(11)))-
c    +            (pr(12)+pr(13)*pr(9)+(pr(13)*pr(11)*(pr(14)+pr(16)
c    +            *pr(9))/(1-pr(16)*pr(11))))*
c    +          (Dpr10Ls1*pr(13




c++++++++++++++++++++++++++++++++++++
c----------------------------------------------------------------
      DTp2 = Wpi*(Tp1-Tp2)/(DENSw*Ap*(L-Ls1))
+         -Upm*Spm2*(Tp2-Tm2)/(Mp2*Cp1) -(Tp1-Tp2)*AUX(1)/(L-Ls1)

      DTp2Tp2=(-Wpi/(DENSw*Ap*(L-Ls1))-Upm*Spm2/(Mp2*Cp1)+
+         AUX(1)/(L-Ls1))*deltat+1.

cc
```

```
      DTp3 = Wpi*(Tp2-Tp3)/(DENSw*Ap*(L-Ls1))
     +        -Upm*Spm2*(Tp3-Tm3)/(Mp2*Cp1)


      DTp3Tp3=(-Wpi/(DENSw*Ap*(L-Ls1))-Upm*Spm2/(Mp2*Cp1))*
     +        deltat+1.


      DTp4 = Wpi*(Tp3-Tp4)/(DENSw*Ap*Ls1)
     +        -Upm*Spm1*(Tp4-Tm4)/(Mp1*Cp1) -(Tp4-Tp3)*AUX(1)/Ls1

      DTp4Tp4=(-Wpi/(DENSw*Ap*Ls1)-Upm*Spm1/(Mp1*Cp1)-AUX(1)/Ls1)
     +        *deltat+1.

       DTpo = (1./Thpi)*(Tp4 -Tpo)

      DTpoTpo= 1.-1./Thpi*deltat

       DTm1 = ( (Upm*Spm1)/(Mm1*Cm) )*Tp1+Td*Ums1*Sms1/(2*Mm1*Cm)
     +        -( (Upm*Spm1 + Ums1*Sms1)/(Mm1*Cm) )*Tm1+
     +        ( Ums1*Sms1/(2*Mm1*Cm) )*(X1 + K5*P)-
     +        (Tm1-Tm2)*AUX(1)/(2*Ls1)


      DTm1Tm1=1.-((Upm*Spm1 + Ums1*Sms1)/(Mm1*Cm)+AUX(1)/(2*Ls1))
     +        *deltat-(Tm1-Tm2)*DAUX1Tm1/(2*Ls1)*deltat


      Dls1=aux(1)

c     Dls1ls1=DAUX1Ls1

      Dls1Ls1=1.

       DTm2 = ( (Upm*Spm2)/(Mm2*Cm) )*Tp2+
     +        -( (Upm*Spm2 + Ums2*Sms2)/(Mm2*Cm) )*Tm2 +
     +        ( Ums2*Sms2/(Mm2*Cm) )*(X1 + K5*P)-
     +        (Tm1-Tm2)*AUX(1)/(2*(L-Ls1))

      DTm2Tm2= 1.-(((Upm*Spm2 + Ums2*Sms2)/(Mm2*Cm))-
     +         AUX(1)/(2*(L-Ls1)))*deltat+
     +         (Tm1-Tm2)*DAUX1Tm2/(2*(L-Ls1))*deltat


      DTm3 = ( (Upm*Spm3)/(Mm3*Cm) )*Tp3
     +        -( (Upm*Spm3 + Ums2*Sms3)/(Mm3*Cm) )*Tm3
     +        +( Ums2*Sms3/(Mm3*Cm) )*(X1 + K5*P)-
     +         (Tm4-Tm3)*AUX(1)/(2*(L-Ls1))

      DTm3Tm3=1.-((Upm*Spm3 + Ums2*Sms3)/(Mm3*Cm)+
     +        AUX(1)/(2*(L-Ls1)))*deltat-
     +        (Tm4-Tm3)*DAUX1Tm3/(2*(L-Ls1))*deltat

       DTm4 = ( (Upm*Spm4)/(Mm4*Cm) )*Tp4+Td*Ums1*Sms4/(2*Mm4*Cm)
     +        -( (Upm*Spm4 + Ums1*Sms4)/(Mm4*Cm) )*Tm4
     +        +( Ums1*Sms4/(2*Mm4*Cm) )*(X1 + K5*P)-
     +       (Tm4-Tm3)*AUX(1)/(2*Ls1)

      DTm4Tm4= 1.-((Upm*Spm4 + Ums1*Sms4)/(Mm4*Cm)-AUX(1)/(2*Ls1))
     +        *deltat-(Tm4-Tm3)*DAUX1Tm4/(2*Ls1)*deltat

      return
      end

      subroutine firla
      real*8 tsam,a1,a2,a3,p0,puv0,pin0,pr0,pdv0,pdis,puvd0,wmfp00,q0
      real*8 effp,wsg00,i,ltt,kp,tau,area,hin0,hout0,kr1,kr2,k11,k12,lsp,lset
      real*8 psuc,tau1,tgo,npump0,wf0,tkick,incl,intpi,intfi,inwpi,inwfi,tmax
      real*8 densm,densw,densr0,densd,densdw,densg0,denss,densb0,n
      real*8 do,di,l,ls10,ar,adw,ad,afs,lr,ldw0,ld,vp,vs,vr,vdr
      real*8 thetai,tpix,tp10,tp20,tp40,tpo0,tm10,tm20,tm40
      real*8 tdw0,td0,tsat0,tfix,tfw,tp30,tm30,tfi0,hf,hfg,vf
      real*8 vfg,xe0,k1,k2,k3,k4,k5,k6,k7,x1,x2,x3,x4,x5,x6,hi,hos
      real*8 hob,kth,cp1,cp2,cm,wfi0,wpix,w10,c1x,cd,tou,tou1,tou2,g1,g2,gv
      real*8 wnv,ztv,v0,u0,w0,r0,m0,pi,rho1,wmftp0,kr,i0r,phd0,h0,pdis0
      real*8 f1,f2,f3,kv,fv,hout,w20,w30,w40,ls20,mm,mm1,mm4,mm2
      real*8 mm3,sm,sms1,sms2,sms3,sms4,spm1,spm2,spm3,spm4,pr1
      real*8 pr2,dm,ap,mp,mp1,mp2,mp3,mp4,ms1,upm,ums1,ums2,vpi,mpi,mpo,md
```

```
      real*8 hb0,hxe0,lb0,c1,wr,xldw0,xwst0,xp0,txde10,xtfi,tfi,thpi
      real*8 wpi,tpi0,cl,wst,den1,den2,k(27),pr(16),aux(10),w(4),afwv0,fwcont
      real*8 delps,gain1,gain2,gain3,gain4,lsets,puvds,puvs,phds,pdiss
      real*8 hs,nfs1,npump1,dum1,arv1,nf,afwv,wfi,phd
      real*8 puv,pdv,deltap,h,xpt,xpump,wmfpt,wfis,afwvs
      real*8 tpi,tp1,tp2,tp3,tp4,tpo,tm1,tm2,tm3,tm4,densb,densr,ls1,xe,ldw
      real*8 tdw,p,td,wf,puvd,npump,dtpi,dtp1,dtp2,dtp3,dtp4,dtpo,dtm1
      real*8 dtm2,dtm3,dtm4,ddensb,ddensr,dls1,dxe,dldw
      real*8 dtdw,dp,dtd,dwf,dpuvd,dnpump,lll,econtr(2),auto(2)


      real*8 xlset,xldw,x11,dx12,dx13,ta13,ka13,bxst,bxwf,xst,xwf,dx14,x13
      real*8 ka14,x14,x14a,x14b,x15,x12,x16,afwvb,kfin
      real*8 x120,x130,x140,ta12,ta14
      common /ali33/x120,x130,x140,ta12,ta14
      common /ali31/ xldw,x11,dx12,dx13,ta13,ka13,bxst,bxwf,xst,xwf,dx14,x13
      common /ali32/ xlset,ka14,x14,x14a,x14b,x15,x12,x16,afwvb,kfin



      common /ali01/
     +        tsam,a1,a2,a3,p0,puv0,pin0,pr0,pdv0,pdis,puvd0,wmfp00,q0
      common /ali02/
     +        effp,wsg00,i,ltt,kp,tau,area,hin0,hout0,kr1,kr2,kl1,kl2,lsp,lset
      common /ali00/
     +        psuc,tau1,tgo,npump0,wf0,tkick,incl,intpi,intfi,inwpi,inwfi,tmax
      common /ali03/ densm,densw,densr0,densd,densdw,densg0,denss,densb0,n
      common /ali04/ do,di,ls10,ar,adw,ad,afs,lr,ldw0,ld,vp,vs,vr,vdr
      common /ali05/ thetai,tpix,tp10,tp20,tp40,tpo0,tm10,tm20,tm40
      common /ali06/ tdw0,td0,tsat0,tfix,tfw,tp30,tm30,tfi0,hf,hfg,vf
      common /ali07/ vfg,xe0,k1,k2,k3,k4,k5,k6,k7,x1,x2,x3,x4,x5,x6,hi,hos
      common /ali08/ hob,kth,cp1,cp2,cm,wfi0,wpix
       common /deli01/ w10,clx,cd,tou,tou1,tou2,g1,g2,gv
      common /ali09/ wnv,ztv,v0,u0,w0,r0,m0,pi,rho1
      common /deli02/ wmftp0,kr,i0r,phd0,h0,pdis0
      common /ali10/ f1,f2,f3,kv,fv,hout,w20,w30,w40,ls20,mm,mm1,mm4,mm2
      common /ali11/ mm3,sm,sms1,sms2,sms3,sms4,spm1,spm2,spm3,spm4,pr1
      common /ali12/ pr2,dm,ap,mp,mp1,mp2,mp3,mp4,ms1,upm
       common /deli03/ums1,ums2,vpi,mpi,mpo,md,thpi
      common /ali13/ hb0,hxe0,lb0,c1,wr,xldw0,xwst0,xp0,txde10,xtfi,tfi
      common /ali14/ wpi,tpi0,cl,wst,den1,den2
       common /deli04/ k,pr,aux,w,afwv0,fwcont
      common /ali15/
     +        delps,gain1,gain2,gain3,gain4,lsets,puvds,puvs,phds,pdiss
      common /ali16/ hs,nfs1,npump1,dum1,arv1,nf,afwv,wfi,phd
      common /ali17/ puv,pdv,deltap,h,xpt,xpump,wmfpt,wfis,afwvs
      common /ali18/
     +        tpi,tp1,tp2,tp3,tp4,tpo,tm1,tm2,tm3,tm4,densb,densr,ls1,xe,ldw
      common /ali19/
     +        tdw,p,td,wf,puvd,npump,dtpi,dtp1,dtp2,dtp3,dtp4,dtpo,dtm1
      common /ali20/ dtm2,dtm3,dtm4,ddensb,ddensr,dls1,dxe,dldw
      common /ali21/ dtdw,dp,dtd,dwf,dpuvd,dnpump,lll,econtr,auto
      tsam=5
      a1=2440
      a2=241.19
      a3=-869.17
      p0=848.934
      puv0=917
      pin0=160
      pr0=160
      pdv0=877.6
      pdis=989.17
      puvd0=917.584
      wmfp00=132274
      q0=18600.98
      effp=1
      wsg00=14921703
      i=160
      ltt=1000
      kp=5
      tau=100
      area=2.7
      hin0=1271.4
      hout=976.2
      kr1=0.03
      kr2=0.0003
      kl1=3.3
      kl2=2
```

```
lsp=10
lset=42.17
psuc=360
tau1=5
tgo=0
npump0=5343.31
wf0=1035.26
tkick=500
incl=0
intpi=0
intfi=0
inwpi=0
inwfi=0
tmax=100
densm=530
densw=45.710
densr0=7.87695
densd=50.32
densdw=47.66
densg0=1.8325
denss=52.32
densb0=13.6269
n=3388
do=.875
di=.775
l=35.54
ls10=3.44372
ar=48.7
adw=110.74
ad=32
afs=60.67
lr=9.63
ldw0=9.63
ld=35.54
vp=1077
vs=3332.28
vr=468.981
vdr=4398.706
thetai=592.5
tpix=592.5
tp10=587.37
tp20=557.343
tp40=539.256
tpo0=539.256
tm10=553.468
tm20=536.05
tm40=527.302
tdw0=504.315
td0=504.315
tsat0=521.9
tfix=434.3
tfw=434.3
tp30=541.065
tm30=529.521
tfi0=434
hf=515.2
hfg=678.3
vf=0.02098
vfg=0.5247
xe0=0.19975
k1=3.5e-6
k2=-7.135e-4
k3=0.17
k4=-0.2
k5=0.14
k6=0.14
k7=2.37e-3
x1=402.94
x2=0.018
x3=1.13096
x4=370.751
x5=850.04
x6=-0.181289
hi=1.25
hos=0.87603
hob=1.87
kth=0.0088275
cp1=1.39
```

```
cp2=1.165
cm=0.11
wfi0=1035.26
wpix=10941.6
w10=5181.95
clx=1.2195
cd=4.10148623e-7
tou=5.2
tou1=250
tou2=120
g1=65.2
g2=1.0
gv=32.2
wnv=0.63
ztv=3.18
v0=0
u0=0
w0=0
r0=0
m0=0
pi=acos(-1.)
Wmfpt0=wmfp00/3600/2
Kr=Wmfpt0/166./0.5
I0r=0.5/Kr2
rhol=50
Phd0=45. + 0.66E-05*Wf0
H0=a1+a2+a3
Pdis0=Psuc + H0*Rho1/144
f1=(Pdis0 - Phd0)/Wf0**2
f2=(Phd0 - Puv0)/Wf0**2
f3=(Pdv0 - P0)/Wf0**2
Kv=Wf0/sqrt(Puv0 - Pdv0)
fv=1/Kv/Kv
hout=hin0 - H0*Wf0/(778*Effp*Wmfpt0)

 W20  = W10
 W30  = W10
 W40  = W10
Ls20 = L - Ls10
Mm   = DENSm*N*L*PI*(Do**2-Di**2)/(4*144)
Mml  = Mm*Ls10/L
Mm4  = Mm1
Mm2  = Mm*(Ls20/L)
Mm3  = Mm2
Sm   = L*PI*Do*N/12
Sms1 = Sm*Ls10/L
Sms2 = Sm*(Ls20/L)
 Sms3 = Sms2
 Sms4 = Sms1
 Spml = Sms1*Di/Do
Spm2 = Sms2*Di/Do
Spm3 = Spm2
Spm4 = Spm1
Pr1  = Spm1/Ls10
Pr2  = Sms2/Ls20
Dm   = (Di +Do)/2

 Ap   = PI*Di**2*N/(4.*144.)
 Mp   = DENSw*Ap*L
 Mp1  = Mp*Ls10/L
 Mp2  = Mp*Ls20/L
Mp3  = Mp2
Mp4  = Mp1
Ms1  = Afs*DENSs*Ls10

Upm  = 1./(1./hi +(Di*alog(real(Dm/Di)))/(24*Kth) )
Ums1 = 1./(1./hos + (Do*alog(real(Do/Dm)))/(24*Kth))
Ums2 = 1./(1./hob + (Do*alog(real(Do/Dm)))/(24*Kth))

Vpi  = (Vp-Ap*2*L)/2
Mpi  = DENSw*Vpi
Mpo  = Mpi
Md   = DENSd*Ad*Ld

wpi=wpix

 Thpi  = Mpi/Wpi
```

```
        Hb0   = Hf + Xe0*Hfg/2
        Hxe0  = Hf + Xe0*Hfg
        DENSb0= 1/(Vf+Xe0*Vfg/2.)
        Lb0   = Ls20

        C1   = 1./SQRT(Cd)
        Wr   = (1-Xe0)*W40

        X1dw0=1dw0
        Xwst0=p0*c1x
        Xp0=p0


        txdel0=(tp40-tp10)*(tfi0-tdw0)

        xtfi0=tfix

        l11=1
        tpi0=tpix
        delps=195.0
        gain1=10.0
        gain2=10.0
        gain3=5.0
        gain4=5.0
        lsets=9.63
        tfi=tfix
        cl=clx

        ta12=5.0
        ta13=1800.0
        ka13=4.0
        ta14=300.0
        ka14=0.001
        kfin=1.0
        x120=0.0
        x130=0.0
        x140=0.0
        return
        end

Sub kf (kflev, kfpre)

 est(1) = 1.273137 * .331576 * kflev + .4570648 * .02491578 * deger(6) + .02568717 *
(.331576 * kflev) ^ 2 + .2142974 * .005272023 * deger(5) - .0005036674 * .005272023 *
deger(5) * (.02491578 * deger(6)) ^ 2 + 17.95273
 est(2) = 15.52209 * .02975807 * kfpre + 9.233875 * .02491578 * deger(6) + 70.69355 *
.01619268 * deger(2) - 4.102993 * .005272023 * deger(5) - 4.965631 * .03428328 * deger(3)
- 296.0486


 dldl = 1.273137 * .331576 + 2# * .02568717 * (.331576 * kflev) * .331576
 dldp = 0#
 dpdl = 0#
 dpdp = 15.52209 * .02975807

 f(1, 1) = dldl
 f(1, 2) = dldp
 f(2, 1) = dpdl
 f(2, 2) = dpdp

 po(1, 1) = f(1, 1) * (p(1, 1) * f(1, 1) + p(1, 2) * f(1, 2)) + f(1, 2) * (p(2, 1) * f(1,
1) + p(2, 2) * f(1, 2))
 po(1, 2) = f(1, 1) * (p(1, 1) * f(2, 1) + p(1, 2) * f(2, 2)) + f(1, 2) * (p(2, 1) * f(2,
1) + p(2, 2) * f(2, 2))
 po(2, 1) = f(2, 1) * (p(1, 1) * f(1, 1) + p(1, 2) * f(1, 2)) + f(2, 2) * (p(2, 1) * f(1,
1) + p(2, 2) * f(1, 2))
 po(2, 2) = f(2, 1) * (p(1, 1) * f(2, 1) + p(1, 2) * f(2, 2)) + f(2, 2) * (p(2, 1) * f(2,
1) + p(2, 2) * f(2, 2))

 tmp(1, 1) = po(1, 1) + .3
 tmp(1, 2) = po(1, 2)
 tmp(2, 1) = po(2, 1)
 tmp(2, 2) = po(2, 2) + 1#

 dettmp = tmp(1, 1) * tmp(2, 2) - tmp(1, 2) * tmp(2, 1)

 tmpi(1, 1) = tmp(2, 2) / dettmp
```

```
 tmpi(2, 2) = tmp(1, 1) / dettmp
 tmpi(1, 2) = -tmp(1, 2) / dettmp
 tmpi(2, 1) = -tmp(2, 1) / dettmp

 g(1, 1) = po(1, 1) * tmpi(1, 1) + po(1, 2) * tmpi(2, 1)
 g(1, 2) = po(1, 1) * tmpi(1, 2) + po(1, 2) * tmpi(2, 2)
 g(2, 1) = po(2, 1) * tmpi(1, 1) + po(2, 2) * tmpi(2, 1)
 g(2, 2) = po(2, 1) * tmpi(1, 2) + po(2, 2) * tmpi(2, 2)

 kerr(1) = deger(7) - est(1)
 kerr(2) = deger(10) - est(2)

 corr(1) = g(1, 1) * kerr(1) + g(1, 2) * kerr(2)
 corr(2) = g(2, 1) * kerr(1) + g(2, 2) * kerr(2)

 est(1) = est(1) + corr(1)
 est(2) = est(2) + corr(2)

 p(1, 1) = (1# - g(1, 1)) * po(1, 1) + g(1, 2) * po(2, 1)
 p(1, 2) = (1# - g(1, 1)) * po(1, 2) + g(1, 2) * po(2, 2)
 p(2, 1) = g(2, 1) * po(1, 1) + (1# - g(2, 2)) * po(2, 1)
 p(2, 2) = g(2, 1) * po(1, 2) + (1# - g(2, 2)) * po(2, 2)

 kflev = est(1)
 kfpre = est(2)

End Sub
```

APPENDIX E

Code Listing for System Executive

```
VERSION 2.00
Begin Form display
   BackColor       =   &H00C0C0C0&
   Caption         =   "Instant SV for Steam Generator A - Unit 1"
   ClientHeight    =   8715
   ClientLeft      =   705
   ClientTop       =   300
   ClientWidth     =   11295
   Height          =   9120
   Icon            =   DISPLAY.FRX:0000
   Left            =   645
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   ScaleHeight     =   8715
   ScaleWidth      =   11295
   Top             =   -45
   Width           =   11415
   Begin CommonDialog CMDialog1
      DialogTitle     =   "SVS Help"
      Left            =   10200
      Top             =   7440
   End
   Begin SSPanel Panel3D1
      BackColor       =   &H00C0C0C0&
      BevelInner      =   1  'Inset
      Caption         =   "Last SV at DD-MMM-YYYY HH:MM:SS.ss"
      Font3D          =   1  'Raised w/light shading
      Height          =   855
      Left            =   8880
      TabIndex        =   49
      Top             =   6480
      Width           =   2295
   End
   Begin Timer Timer1
      Interval        =   60000
      Left            =   10560
      Top             =   7920
   End
   Begin SSCommand Command3D9
      Caption         =   "Print All SV"
      Font3D          =   3  'Inset w/light shading
      Height          =   735
      Left            =   8880
      Picture         =   DISPLAY.FRX:0302
      TabIndex        =   48
      Top             =   7680
      Width           =   1215
   End
   Begin SSFrame Frame3D13
      Alignment       =   2  'Center
      Caption         =   "System"
      Font3D          =   0  'None
      ForeColor       =   &H00000000&
      Height          =   2295
      Left            =   6960
      TabIndex        =   44
      Top             =   6360
      Width           =   1695
      Begin SSCommand Command3D6
         Caption         =   "About"
         Font3D          =   3  'Inset w/light shading
         Height          =   495
         Left            =   240
         TabIndex        =   47
         Top             =   360
```

```
      Width           =   1215
   End
   Begin SSCommand Command3D7
      Caption         =   "Help"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   240
      TabIndex        =   46
      Top             =   960
      Width           =   1215
   End
   Begin SSCommand Command3D8
      Caption         =   "Quit"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   240
      TabIndex        =   45
      Top             =   1560
      Width           =   1215
   End
End
Begin SSFrame Frame3D12
   Alignment       =   2  'Center
   Caption         =   "History"
   Font3D          =   0  'None
   Height          =   2295
   Left            =   3240
   TabIndex        =   39
   Top             =   6360
   Width           =   3495
   Begin SSCommand Command3D1
      Caption         =   "SV Graph"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   2040
      TabIndex        =   43
      Top             =   960
      Width           =   1215
   End
   Begin SSCommand Command3D2
      Caption         =   "Signals"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   240
      TabIndex        =   42
      Top             =   360
      Width           =   1215
   End
   Begin SSCommand Command3D3
      Caption         =   "Nominal"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   240
      TabIndex        =   41
      Top             =   960
      Width           =   1215
   End
   Begin SSCommand Command3D4
      Caption         =   "Y-Axis"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   240
      TabIndex        =   40
      Top             =   1560
      Width           =   1215
   End
   Begin Line Line1
      BorderColor     =   &H00808080&
      BorderWidth     =   3
      X1              =   1440
      X2              =   1680
      Y1              =   600
      Y2              =   600
   End
   Begin Line Line2
      BorderColor     =   &H00808080&
      BorderWidth     =   3
      X1              =   1440
```

```
         X2              =   1680
         Y1              =   1200
         Y2              =   1200
      End
      Begin Line Line3
         BorderColor     =   &H00808080&
         BorderWidth     =   3
         X1              =   1440
         X2              =   1680
         Y1              =   1800
         Y2              =   1800
      End
      Begin Line Line4
         BorderColor     =   &H00808080&
         BorderWidth     =   3
         X1              =   1680
         X2              =   1680
         Y1              =   600
         Y2              =   1800
      End
      Begin Line Line5
         BorderColor     =   &H00808080&
         BorderWidth     =   3
         X1              =   1680
         X2              =   2040
         Y1              =   1200
         Y2              =   1200
      End
      Begin Line Line6
         BorderColor     =   &H00808080&
         BorderWidth     =   3
         X1              =   2040
         X2              =   1920
         Y1              =   1200
         Y2              =   1320
      End
      Begin Line Line7
         BorderColor     =   &H00808080&
         BorderWidth     =   3
         X1              =   1920
         X2              =   2040
         Y1              =   1080
         Y2              =   1200
      End
   End
   Begin SSFrame Pressure
      Alignment       =   2  'Center
      Caption         =   "Pressure"
      Font3D          =   1  'Raised w/light shading
      ForeColor       =   &H00000000&
      Height          =   6015
      Left            =   3240
      TabIndex        =   19
      Top             =   120
      Width           =   7935
      Begin Gauge Gauge2
         Autosize        =   -1  'True
         BackColor       =   &H00808080&
         ForeColor       =   &H00808080&
         Height          =   2775
         Index           =   2
         InnerBottom     =   5
         InnerLeft       =   5
         InnerRight      =   5
         InnerTop        =   5
         Left            =   5880
         Max             =   1100
         Min             =   700
         NeedleWidth     =   1
         Style           =   1  'Vertical Bar
         TabIndex        =   55
         Top             =   480
         Value           =   700
         Width           =   1095
      End
      Begin Gauge Gauge2
         Autosize        =   -1  'True
         BackColor       =   &H00808080&
```

```
      ForeColor       =   &H00808080&
      Height          =   2775
      Index           =   1
      InnerBottom     =   5
      InnerLeft       =   5
      InnerRight      =   5
      InnerTop        =   5
      Left            =   3360
      Max             =   1100
      Min             =   700
      NeedleWidth     =   1
      Style           =   1  'Vertical Bar
      TabIndex        =   54
      Top             =   480
      Value           =   700
      Width           =   1095
   End
   Begin SSFrame Measurement
      Alignment       =   2  'Center
      Caption         =   "Measurement"
      Font3D          =   1  'Raised w/light shading
      ForeColor       =   &H00000000&
      Height          =   1095
      Index           =   3
      Left            =   5280
      TabIndex        =   36
      Top             =   3360
      Width           =   2415
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   19
         Left            =   1800
         Picture         =   DISPLAY.FRX:0484
         ScaleHeight     =   495
         ScaleWidth      =   495
         TabIndex        =   74
         Top             =   480
         Visible         =   0   'False
         Width           =   495
      End
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   18
         Left            =   1800
         Picture         =   DISPLAY.FRX:0786
         ScaleHeight     =   495
         ScaleWidth      =   495
         TabIndex        =   73
         Top             =   480
         Visible         =   0   'False
         Width           =   495
      End
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   17
         Left            =   1800
         Picture         =   DISPLAY.FRX:0A88
         ScaleHeight     =   495
         ScaleWidth      =   495
         TabIndex        =   72
         Top             =   480
         Visible         =   0   'False
         Width           =   495
      End
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   16
         Left            =   1800
         Picture         =   DISPLAY.FRX:0D8A
         ScaleHeight     =   495
```

```
         ScaleWidth      =   495
         TabIndex        =   71
         Top             =   480
         Visible         =   0   'False
         Width           =   495
      End
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   15
         Left            =   1800
         Picture         =   DISPLAY.FRX:108C
         ScaleHeight     =   495
         ScaleWidth      =   495
         TabIndex        =   70
         Top             =   480
         Visible         =   0   'False
         Width           =   495
      End
      Begin SSFrame Frame3D2
         Alignment       =   2  'Center
         Caption         =   "1P0402A"
         Font3D          =   1  'Raised w/light shading
         ForeColor       =   &H00000000&
         Height          =   735
         Index           =   3
         Left            =   240
         TabIndex        =   37
         Top             =   240
         Width           =   1455
         Begin SSPanel mes
            BackColor       =   &H00C0C0C0&
            BevelInner      =   2  'Raised
            BevelOuter      =   1  'Inset
            Font3D          =   3  'Inset w/light shading
            ForeColor       =   &H00404080&
            Height          =   375
            Index           =   3
            Left            =   120
            TabIndex        =   38
            Top             =   240
            Width           =   1215
         End
      End
   End
   Begin SSFrame Measurement
      Alignment       =   2  'Center
      Caption         =   "Measurement"
      Font3D          =   1  'Raised w/light shading
      ForeColor       =   &H00000000&
      Height          =   1095
      Index           =   2
      Left            =   2760
      TabIndex        =   33
      Top             =   3360
      Width           =   2415
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   14
         Left            =   1800
         Picture         =   DISPLAY.FRX:138E
         ScaleHeight     =   495
         ScaleWidth      =   495
         TabIndex        =   69
         Top             =   480
         Visible         =   0   'False
         Width           =   495
      End
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   13
         Left            =   1800
         Picture         =   DISPLAY.FRX:1690
```

```
      ScaleHeight     =   495
      ScaleWidth      =   495
      TabIndex        =   68
      Top             =   480
      Visible         =   0   'False
      Width           =   495
   End
   Begin PictureBox Picture1
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      Height          =   495
      Index           =   12
      Left            =   1800
      Picture         =   DISPLAY.FRX:1992
      ScaleHeight     =   495
      ScaleWidth      =   495
      TabIndex        =   67
      Top             =   480
      Visible         =   0   'False
      Width           =   495
   End
   Begin PictureBox Picture1
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      Height          =   495
      Index           =   11
      Left            =   1800
      Picture         =   DISPLAY.FRX:1C94
      ScaleHeight     =   495
      ScaleWidth      =   495
      TabIndex        =   66
      Top             =   480
      Visible         =   0   'False
      Width           =   495
   End
   Begin PictureBox Picture1
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      Height          =   495
      Index           =   10
      Left            =   1800
      Picture         =   DISPLAY.FRX:1F96
      ScaleHeight     =   495
      ScaleWidth      =   495
      TabIndex        =   65
      Top             =   480
      Visible         =   0   'False
      Width           =   495
   End
   Begin SSFrame Frame3D2
      Alignment       =   2  'Center
      Caption         =   "1P0401A"
      Font3D          =   1  'Raised w/light shading
      ForeColor       =   &H00000000&
      Height          =   735
      Index           =   2
      Left            =   240
      TabIndex        =   34
      Top             =   240
      Width           =   1455
      Begin SSPanel mes
         BackColor       =   &H00C0C0C0&
         BevelInner      =   2  'Raised
         BevelOuter      =   1  'Inset
         Font3D          =   3  'Inset w/light shading
         ForeColor       =   &H00404080&
         Height          =   375
         Index           =   2
         Left            =   120
         TabIndex        =   35
         Top             =   240
         Width           =   1215
      End
   End
End
Begin Gauge Gauge2
   Autosize        =   -1  'True
   BackColor       =   &H00808080&
```

```
      ForeColor       =   &H00808080&
      Height          =   2775
      Index           =   0
      InnerBottom     =   5
      InnerLeft       =   5
      InnerRight      =   5
      InnerTop        =   5
      Left            =   840
      Max             =   1100
      Min             =   700
      NeedleWidth     =   1
      Style           =   1  'Vertical Bar
      TabIndex        =   32
      Top             =   480
      Value           =   700
      Width           =   1095
   End
   Begin SSFrame Measurement
      Alignment       =   2  'Center
      Caption         =   "Measurement"
      Font3D          =   1  'Raised w/light shading
      ForeColor       =   &H00000000&
      Height          =   1095
      Index           =   1
      Left            =   240
      TabIndex        =   29
      Top             =   3360
      Width           =   2415
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   9
         Left            =   1800
         Picture         =   DISPLAY.FRX:2298
         ScaleHeight     =   495
         ScaleWidth      =   495
         TabIndex        =   64
         Top             =   480
         Visible         =   0   'False
         Width           =   495
      End
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   8
         Left            =   1800
         Picture         =   DISPLAY.FRX:259A
         ScaleHeight     =   495
         ScaleWidth      =   495
         TabIndex        =   63
         Top             =   480
         Visible         =   0   'False
         Width           =   495
      End
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   7
         Left            =   1800
         Picture         =   DISPLAY.FRX:289C
         ScaleHeight     =   495
         ScaleWidth      =   495
         TabIndex        =   62
         Top             =   480
         Visible         =   0   'False
         Width           =   495
      End
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   6
         Left            =   1800
         Picture         =   DISPLAY.FRX:2B9E
         ScaleHeight     =   495
```

```
         ScaleWidth      =   495
         TabIndex        =   61
         Top             =   480
         Visible         =   0   'False
         Width           =   495
      End
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   5
         Left            =   1800
         Picture         =   DISPLAY.FRX:2EA0
         ScaleHeight     =   495
         ScaleWidth      =   495
         TabIndex        =   60
         Top             =   480
         Visible         =   0   'False
         Width           =   495
      End
      Begin SSFrame Frame3D2
         Alignment       =   2  'Center
         Caption         =   "1P0400A"
         Font3D          =   1  'Raised w/light shading
         ForeColor       =   &H00000000&
         Height          =   735
         Index           =   1
         Left            =   240
         TabIndex        =   30
         Top             =   240
         Width           =   1455
         Begin SSPanel mes
            BackColor       =   &H00C0C0C0&
            BevelInner      =   2  'Raised
            BevelOuter      =   1  'Inset
            Font3D          =   3  'Inset w/light shading
            ForeColor       =   &H00404080&
            Height          =   375
            Index           =   1
            Left            =   120
            TabIndex        =   31
            Top             =   240
            Width           =   1215
         End
      End
   End
   Begin SSFrame Frame3D7
      Alignment       =   2  'Center
      Caption         =   "Estimates"
      Font3D          =   1  'Raised w/light shading
      ForeColor       =   &H00000000&
      Height          =   1215
      Left            =   240
      TabIndex        =   20
      Top             =   4560
      Width           =   7455
      Begin SSFrame Frame3D11
         Alignment       =   2  'Center
         Caption         =   "KFT"
         Font3D          =   1  'Raised w/light shading
         ForeColor       =   &H00000000&
         Height          =   735
         Left            =   5760
         TabIndex        =   27
         Top             =   240
         Width           =   1455
         Begin SSPanel est
            BackColor       =   &H00C0C0C0&
            BevelOuter      =   1  'Inset
            Font3D          =   1  'Raised w/light shading
            ForeColor       =   &H00C00000&
            Height          =   375
            Index           =   7
            Left            =   120
            TabIndex        =   28
            Top             =   240
            Width           =   1215
         End
```

```
    End
    Begin SSFrame Frame3D10
       Alignment       =   2  'Center
       Caption         =   "ANN"
       Font3D          =   1  'Raised w/light shading
       ForeColor       =   &H00000000&
       Height          =   735
       Left            =   3960
       TabIndex        =   25
       Top             =   240
       Width           =   1455
       Begin SSPanel est
          BackColor       =   &H00C0C0C0&
          BevelOuter      =   1  'Inset
          Font3D          =   1  'Raised w/light shading
          ForeColor       =   &H00C00000&
          Height          =   375
          Index           =   6
          Left            =   120
          TabIndex        =   26
          Top             =   240
          Width           =   1215
       End
    End
    Begin SSFrame Frame3D9
       Alignment       =   2  'Center
       Caption         =   "PEM"
       Font3D          =   1  'Raised w/light shading
       ForeColor       =   &H00000000&
       Height          =   735
       Left            =   2040
       TabIndex        =   23
       Top             =   240
       Width           =   1455
       Begin SSPanel est
          BackColor       =   &H00C0C0C0&
          BevelOuter      =   1  'Inset
          Font3D          =   1  'Raised w/light shading
          ForeColor       =   &H00C00000&
          Height          =   375
          Index           =   5
          Left            =   120
          TabIndex        =   24
          Top             =   240
          Width           =   1215
       End
    End
    Begin SSFrame Frame3D8
       Alignment       =   2  'Center
       Caption         =   "GCC"
       Font3D          =   1  'Raised w/light shading
       ForeColor       =   &H00000000&
       Height          =   735
       Left            =   240
       TabIndex        =   21
       Top             =   240
       Width           =   1455
       Begin SSPanel est
          BackColor       =   &H00C0C0C0&
          BevelOuter      =   1  'Inset
          Font3D          =   1  'Raised w/light shading
          ForeColor       =   &H00C00000&
          Height          =   375
          Index           =   4
          Left            =   120
          TabIndex        =   22
          Top             =   240
          Width           =   1215
       End
    End
 End
 Begin Label Label8
    BackColor       =   &H00C0C0C0&
    Caption         =   "700 Psig"
    Height          =   255
    Left            =   6960
    TabIndex        =   56
    Top             =   3000
```

```
         Width            =   855
      End
      Begin Label Label7
         BackColor        =   &H00C0C0C0&
         Caption          =   "1100 Psig"
         Height           =   255
         Left             =   6960
         TabIndex         =   59
         Top              =   480
         Width            =   855
      End
      Begin Label Label6
         BackColor        =   &H00C0C0C0&
         Caption          =   "1100 Psig"
         Height           =   255
         Left             =   4440
         TabIndex         =   58
         Top              =   480
         Width            =   855
      End
      Begin Label Label5
         BackColor        =   &H00C0C0C0&
         Caption          =   "700 Psig"
         Height           =   255
         Left             =   4440
         TabIndex         =   57
         Top              =   3000
         Width            =   855
      End
      Begin Label Label4
         BackColor        =   &H00C0C0C0&
         Caption          =   "1100 Psig"
         Height           =   255
         Left             =   1920
         TabIndex         =   53
         Top              =   480
         Width            =   855
      End
      Begin Label Label3
         BackColor        =   &H00C0C0C0&
         Caption          =   "700 Psig"
         Height           =   255
         Left             =   1920
         TabIndex         =   52
         Top              =   3000
         Width            =   855
      End
   End
   Begin SSFrame Level
      Alignment       =   2  'Center
      Caption         =   "Wide Range Water Level"
      Font3D          =   1  'Raised w/light shading
      ForeColor       =   &H00000000&
      Height          =   8535
      Left            =   120
      TabIndex        =   0
      Top             =   120
      Width           =   2895
      Begin SSFrame Estimates
         Alignment       =   2  'Center
         Caption         =   "Estimates"
         Font3D          =   1  'Raised w/light shading
         ForeColor       =   &H00000000&
         Height          =   3735
         Left            =   480
         TabIndex        =   5
         Top             =   4560
         Width           =   1935
         Begin SSFrame Frame3D1
            Alignment       =   2  'Center
            Caption         =   "GCC"
            Font3D          =   1  'Raised w/light shading
            ForeColor       =   &H00000000&
            Height          =   735
            Left            =   240
            TabIndex        =   12
            Top             =   240
            Width           =   1455
```

```
      Begin SSPanel est
         BackColor       =   &H00C0C0C0&
         BevelOuter      =   1  'Inset
         Caption         =   "N/A"
         Font3D          =   1  'Raised w/light shading
         ForeColor       =   &H00C00000&
         Height          =   375
         Index           =   0
         Left            =   120
         TabIndex        =   13
         Top             =   240
         Width           =   1215
      End
   End
   Begin SSFrame Frame3D3
      Alignment       =   2  'Center
      Caption         =   "PEM"
      Font3D          =   1  'Raised w/light shading
      ForeColor       =   &H00000000&
      Height          =   735
      Left            =   240
      TabIndex        =   10
      Top             =   1080
      Width           =   1455
      Begin SSPanel est
         BackColor       =   &H00C0C0C0&
         BevelOuter      =   1  'Inset
         Font3D          =   1  'Raised w/light shading
         ForeColor       =   &H00C00000&
         Height          =   375
         Index           =   1
         Left            =   120
         TabIndex        =   11
         Top             =   240
         Width           =   1215
      End
   End
   Begin SSFrame Frame3D4
      Alignment       =   2  'Center
      Caption         =   "ANN"
      Font3D          =   1  'Raised w/light shading
      ForeColor       =   &H00000000&
      Height          =   735
      Left            =   240
      TabIndex        =   8
      Top             =   1920
      Width           =   1455
      Begin SSPanel est
         BackColor       =   &H00C0C0C0&
         BevelOuter      =   1  'Inset
         Font3D          =   1  'Raised w/light shading
         ForeColor       =   &H00C00000&
         Height          =   375
         Index           =   2
         Left            =   120
         TabIndex        =   9
         Top             =   240
         Width           =   1215
      End
   End
   Begin SSFrame Frame3D5
      Alignment       =   2  'Center
      Caption         =   "KFT"
      Font3D          =   1  'Raised w/light shading
      ForeColor       =   &H00000000&
      Height          =   735
      Left            =   240
      TabIndex        =   6
      Top             =   2760
      Width           =   1455
      Begin SSPanel est
         BackColor       =   &H00C0C0C0&
         BevelOuter      =   1  'Inset
         Font3D          =   1  'Raised w/light shading
         ForeColor       =   &H00C00000&
         Height          =   375
         Index           =   3
         Left            =   120
```

```
            TabIndex        =   7
            Top             =   240
            Width           =   1215
         End
      End
   End
   Begin SSFrame Measurement
      Alignment       =   2  'Center
      Caption         =   "Measurement"
      Font3D          =   1  'Raised w/light shading
      ForeColor       =   &H00000000&
      Height          =   1095
      Index           =   0
      Left            =   240
      TabIndex        =   2
      Top             =   3360
      Width           =   2415
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   2
         Left            =   1800
         Picture         =   DISPLAY.FRX:31A2
         ScaleHeight     =   495
         ScaleWidth      =   495
         TabIndex        =   18
         Top             =   480
         Visible         =   0   'False
         Width           =   495
      End
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   0
         Left            =   1800
         Picture         =   DISPLAY.FRX:34A4
         ScaleHeight     =   495
         ScaleWidth      =   495
         TabIndex        =   17
         Top             =   480
         Visible         =   0   'False
         Width           =   495
      End
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   1
         Left            =   1800
         Picture         =   DISPLAY.FRX:37A6
         ScaleHeight     =   495
         ScaleWidth      =   495
         TabIndex        =   16
         Top             =   480
         Visible         =   0   'False
         Width           =   495
      End
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   3
         Left            =   1800
         Picture         =   DISPLAY.FRX:3AA8
         ScaleHeight     =   495
         ScaleWidth      =   495
         TabIndex        =   15
         Top             =   480
         Visible         =   0   'False
         Width           =   495
      End
      Begin PictureBox Picture1
         BackColor       =   &H00C0C0C0&
         BorderStyle     =   0  'None
         Height          =   495
         Index           =   4
```

```
         Left             =   1800
         Picture          =   DISPLAY.FRX:3DAA
         ScaleHeight      =   495
         ScaleWidth       =   495
         TabIndex         =   14
         Top              =   480
         Visible          =   0   'False
         Width            =   495
      End
      Begin SSFrame Frame3D2
         Alignment        =   2  'Center
         Caption          =   "1L0403A"
         Font3D           =   1  'Raised w/light shading
         ForeColor        =   &H00000000&
         Height           =   735
         Index            =   0
         Left             =   240
         TabIndex         =   3
         Top              =   240
         Width            =   1455
         Begin SSPanel mes
            BackColor        =   &H00C0C0C0&
            BevelInner       =   2  'Raised
            BevelOuter       =   1  'Inset
            Font3D           =   3  'Inset w/light shading
            ForeColor        =   &H00404080&
            Height           =   375
            Index            =   0
            Left             =   120
            TabIndex         =   4
            Top              =   240
            Width            =   1215
         End
      End
   End
   Begin Gauge Gauge1
      Autosize         =   -1  'True
      BackColor        =   &H00808080&
      ForeColor        =   &H00808080&
      Height           =   2775
      InnerBottom      =   5
      InnerLeft        =   5
      InnerRight       =   5
      InnerTop         =   5
      Left             =   840
      Max              =   100
      Min              =   50
      NeedleWidth      =   1
      Style            =   1  'Vertical Bar
      TabIndex         =   1
      Top              =   480
      Value            =   50
      Width            =   1095
   End
   Begin Label Label2
      BackColor        =   &H00C0C0C0&
      Caption          =   "100 %"
      Height           =   255
      Left             =   1920
      TabIndex         =   51
      Top              =   480
      Width            =   615
   End
   Begin Label Label1
      BackColor        =   &H00C0C0C0&
      Caption          =   "50 %"
      Height           =   255
      Left             =   1920
      TabIndex         =   50
      Top              =   3000
      Width            =   495
   End
  End
End
Dim fuzpre(3)

Sub Command3D1_Click ()
 display.WindowState = 1
```

```
 plot.Show
 plot.WindowState = 0
End Sub

Sub Command3D2_Click ()
 signals.Show
End Sub

Sub Command3D3_Click ()
 norm.Show
End Sub

Sub Command3D4_Click ()
 yaxis.Show
End Sub

Sub Command3D6_Click ()
 about.Show
End Sub

Sub Command3D7_Click ()

  cmdialog1.HelpFile = "svs.hlp"
  cmdialog1.HelpCommand = &H101
  cmdialog1.HelpKey = "SVS"
  cmdialog1.Action = 6
End Sub

Sub Command3D8_Click ()
 cmdialog1.HelpCommand = &H2
 cmdialog1.Action = 6
 End
End Sub

Sub Command3D9_Click ()
  printer.Print
  printer.Print "Signal Validation Results for Steam Generator A - Unit 1"
  printer.Print
  printer.Print "Time                           : "; zaman$
  printer.Print
  printer.Print "1L0403A                        : "; deger(7); "%"
  printer.Print "Decision                       : "; plot_data(96, 27)
  printer.Print "1P0400A                        : "; deger(8); "Psig"
  printer.Print "Decision                       : "; plot_data(96, 28)
  printer.Print "1P0401A                        : "; deger(9); "Psig"
  printer.Print "Decision                       : "; plot_data(96, 29)
  printer.Print "1P0402A                        : "; deger(10); "Psig"
  printer.Print "Decision                       : "; plot_data(96, 30)
  printer.Print
  printer.Print "GCC Pressure Estimation        : "; plot_data(96, 5); "Psig"
  printer.Print "PEM Wide Range Level Estimation : "; plot_data(96, 6); "%"
  printer.Print "PEM Pressure Estimation        : "; plot_data(96, 7); "Psig"
  printer.Print "ANN Wide Range Level Estimation : "; plot_data(96, 8); "%"
  printer.Print "ANN Pressure Estimation        : "; plot_data(96, 9); "Psig"
  printer.Print "KFT Wide Range Level Estimation : "; plot_data(96, 10); "%"
  printer.Print "KFT Pressure Estimation        : "; plot_data(96, 11); "Psig"
  printer.Print
  printer.Print "1T0406A                        : "; deger(1); "DegF"
  printer.Print "1T0419A                        : "; deger(2); "DegF"
  printer.Print "1T0418A                        : "; deger(3); "DegF"
  printer.Print "1F0403A                        : "; deger(4); "Kbh"
  printer.Print "1F0405A                        : "; deger(5); "Kbh"
  printer.Print "1P0403A                        : "; deger(6); "Psig"
  printer.EndDoc
  OldWidth = plot.Graph1.Width
  oldheight = plot.Graph1.Height
  plot.Graph1.Width = printer.Width
  plot.Graph1.Height = printer.Height
  plot.Graph1.DrawMode = 5
  printer.EndDoc
  plot.Graph1.Width = OldWidth
  plot.Graph1.Height = oldheight
  OldWidth = gcc_plot.Graph1.Width
  oldheight = gcc_plot.Graph1.Height
  gcc_plot.Graph1.Width = printer.Width
  gcc_plot.Graph1.Height = printer.Height
  gcc_plot.Graph1.DrawMode = 5
  printer.EndDoc
```

```
  gcc_plot.Graph1.Width = OldWidth
  gcc_plot.Graph1.Height = oldheight
  OldWidth = gcc_plot.Graph2.Width
  oldheight = gcc_plot.Graph2.Height
  gcc_plot.Graph2.Width = printer.Width
  gcc_plot.Graph2.Height = printer.Height
  gcc_plot.Graph2.DrawMode = 5
  printer.EndDoc
  gcc_plot.Graph2.Width = OldWidth
  gcc_plot.Graph2.Height = oldheight
  OldWidth = gcc_plot.Graph3.Width
  oldheight = gcc_plot.Graph3.Height
  gcc_plot.Graph3.Width = printer.Width
  gcc_plot.Graph3.Height = printer.Height
  gcc_plot.Graph3.DrawMode = 5
  printer.EndDoc
  gcc_plot.Graph3.Width = OldWidth
  gcc_plot.Graph3.Height = oldheight
  OldWidth = gcc_plot.Graph4.Width
  oldheight = gcc_plot.Graph4.Height
  gcc_plot.Graph4.Width = printer.Width
  gcc_plot.Graph4.Height = printer.Height
  gcc_plot.Graph4.DrawMode = 5
  printer.EndDoc
  gcc_plot.Graph4.Width = OldWidth
  gcc_plot.Graph4.Height = oldheight
End Sub

Sub Form_Load ()
 Rem file i/o
 Rem zaman$ = "24-MAR-1994 20:12:34"
 Rem Open "f:\ase\tva2\tvapem.dat" For Input Access Read As #1
 Rem Input #1, dummy$
 Rem GoTo 10

 Open "svs.dat" For Input As #1
 Input #1, filename$
 Input #1, numskip
 Input #1, alarm(1, 1), alarm(1, 2), alarm(1, 3), alarm(1, 4)
 Input #1, alarm(2, 1), alarm(2, 2), alarm(2, 3), alarm(2, 4)
 Close #1
 Open "log.dat" For Output As #2

 Rem 10:
 Rem begin initialization of modules...

 eskideger(1) = 75#
 eskideger(4) = 830#
 npoint = 2290
 flag = 0
 p(1, 1) = .3
 p(2, 2) = 1#
 p(1, 2) = 0#
 p(2, 1) = 0#

 npoint = 2290
 nsignl = 3
 erb(1) = 4#
 erb(2) = 4#
 erb(3) = 4#
 BIAS(1) = 6#
 BIAS(2) = 6#
 BIAS(3) = 6#
 VAR0(1) = 2#
 VAR0(2) = 2#
 VAR0(3) = 2#
 alpha = .0001
 beta = .0001
 bounda = Log(beta / (1# - alpha))
 boundb = Log((1# - beta) / alpha)
 For k = 1 To nsignl
  sprtb(k) = 0#
  excl(k) = 0#
  NEXCL(k) = 0#
  NBIAS(k) = 0#
  DBIAS(k) = 0#
  SII(k) = 0#
  sd(k) = 0#
```

```
  mstdd(k) = 0#
  mstdn(k) = 0#
  snum(k) = 0#
 Next k

 Rem plot init...

 signals.Check3D1(0).Value = -1
 signals.Check3D1(5).Value = -1
 signals.Check3D1(7).Value = -1
 signals.Check3D1(9).Value = -1
 For i = 0 To 10
  going_to_plot(i + 1) = signals.Check3D1(i).Value
 Next i
 Open "norm.dat" For Input As #5
 For i = 0 To 7
  Input #5, norm_data(i + 10)
  norm.Text1(i).Text = norm_data(i + 10)
 Next i
 norm_data(1) = norm_data(10)
 norm_data(2) = norm_data(11)
 norm_data(3) = norm_data(11)
 norm_data(4) = norm_data(11)
 norm_data(5) = norm_data(11)
 norm_data(6) = norm_data(10)
 norm_data(7) = norm_data(11)
 norm_data(8) = norm_data(10)
 norm_data(9) = norm_data(11)
 Close #5
 plot.Graph1.ThisSet = 1
 plot.Graph1.ThisPoint = 1
   If   signals.Check3D1(0).Value   Then   plot.Graph1.LegendText   =   "1L0403A"   Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 2
 plot.Graph1.ThisPoint = 2
   If   signals.Check3D1(1).Value   Then   plot.Graph1.LegendText   =   "1P0400A"   Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 3
 plot.Graph1.ThisPoint = 3
   If   signals.Check3D1(2).Value   Then   plot.Graph1.LegendText   =   "1P0401A"   Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 4
 plot.Graph1.ThisPoint = 4
   If   signals.Check3D1(3).Value   Then   plot.Graph1.LegendText   =   "1P0402A"   Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 5
 plot.Graph1.ThisPoint = 5
 If signals.Check3D1(4).Value Then plot.Graph1.LegendText = "GCC Pressure Estimate" Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 6
 plot.Graph1.ThisPoint = 6
 If  signals.Check3D1(5).Value  Then  plot.Graph1.LegendText  =  "PEM  Level  Estimate"  Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 7
 plot.Graph1.ThisPoint = 7
 If signals.Check3D1(6).Value Then plot.Graph1.LegendText = "PEM Pressure Estimate" Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 8
 plot.Graph1.ThisPoint = 8
 If  signals.Check3D1(7).Value  Then  plot.Graph1.LegendText  =  "ANN  Level  Estimate"  Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 9
 plot.Graph1.ThisPoint = 9
 If signals.Check3D1(8).Value Then plot.Graph1.LegendText = "ANN Pressure Estimate" Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 10
 plot.Graph1.ThisPoint = 10
 If  signals.Check3D1(9).Value  Then  plot.Graph1.LegendText  =  "KFT  Level  Estimate"  Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 11
 plot.Graph1.ThisPoint = 11
 If signals.Check3D1(10).Value Then plot.Graph1.LegendText = "KFT Pressure Estimate" Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 12
 plot.Graph1.ThisPoint = 12
 If going_to_plot(12) Then plot.Graph1.LegendText = "1T0406A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.ThisSet = 13
```

```
 plot.Graph1.ThisPoint = 13
 If going_to_plot(13) Then plot.Graph1.LegendText = "1T0419A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.ThisSet = 14
 plot.Graph1.ThisPoint = 14
 If going_to_plot(14) Then plot.Graph1.LegendText = "1T0418A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.ThisSet = 15
 plot.Graph1.ThisPoint = 15
 If going_to_plot(15) Then plot.Graph1.LegendText = "1F0403A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.ThisSet = 16
 plot.Graph1.ThisPoint = 16
 If going_to_plot(16) Then plot.Graph1.LegendText = "1F0405A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.ThisSet = 17
 plot.Graph1.ThisPoint = 17
 If going_to_plot(17) Then plot.Graph1.LegendText = "1P0403A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.DrawMode = 3

 Load plot
 Load gcc_plot

 old_inp_checked = 0

 App.HelpFile = "svs.hlp"

End Sub

Sub Timer1_Timer ()

 Rem read data...

  Rem Input #1, deger(1), deger(2), deger(3), deger(4), deger(5), deger(6), deger(7),
deger(8), deger(9), deger(10), deger(11), deger(12)

 Call kutuk_oku

 If eskizaman$ <> zaman$ Then
  eskizaman$ = zaman$
  Panel3D1.Caption = "Last SV at " + zaman$
  plot.Panel3D1.Caption = Panel3D1.Caption
  gcc_plot.Panel3D1.Caption = Panel3D1.Caption

  plot.Graph1.ThisPoint = 1
  plot.Graph1.LabelText = DateAdd("d", -1, zaman$)
  plot.Graph1.ThisPoint = 49
  plot.Graph1.LabelText = DateAdd("h", -12, zaman$)
  gcc_plot.Graph1.ThisPoint = 1
  gcc_plot.Graph1.LabelText = DateAdd("d", -1, zaman$)
  gcc_plot.Graph1.ThisPoint = 49
  gcc_plot.Graph1.LabelText = DateAdd("h", -12, zaman$)
  gcc_plot.Graph2.ThisPoint = 1
  gcc_plot.Graph2.LabelText = DateAdd("d", -1, zaman$)
  gcc_plot.Graph2.ThisPoint = 49
  gcc_plot.Graph2.LabelText = DateAdd("h", -12, zaman$)
  gcc_plot.Graph3.ThisPoint = 1
  gcc_plot.Graph3.LabelText = DateAdd("d", -1, zaman$)
  gcc_plot.Graph3.ThisPoint = 49
  gcc_plot.Graph3.LabelText = DateAdd("h", -12, zaman$)
  gcc_plot.Graph4.ThisPoint = 1
  gcc_plot.Graph4.LabelText = DateAdd("d", -1, zaman$)
  gcc_plot.Graph4.ThisPoint = 49
  gcc_plot.Graph4.LabelText = DateAdd("h", -12, zaman$)



 Rem begin SV...

 If flag = 0 Then
  lev_kft = deger(7)
  pre_kft = deger(10)
  flag = 1
 End If
 lev_ann = ann_lev()
 pre_ann = ann_pre()
```

```
 Call kf(lev_kft, pre_kft)
 Call pem(lev_pem, pre_pem)
 Call gcc
 eskideger(1) = deger(7)
 eskideger(4) = deger(10)

 Rem GUI

 For i = 1 To 95
  For j = 1 To 30
   plot_data(i, j) = plot_data(i + 1, j)
  Next j
 Next i

 plot_data(96, 1) = deger(7)
 plot_data(96, 2) = deger(8)
 plot_data(96, 3) = deger(9)
 plot_data(96, 4) = deger(10)
 plot_data(96, 5) = xestmt
 plot_data(96, 6) = lev_pem
 plot_data(96, 7) = pre_pem
 plot_data(96, 8) = lev_ann
 plot_data(96, 9) = pre_ann
 plot_data(96, 10) = lev_kft
 plot_data(96, 11) = pre_kft
 plot_data(96, 12) = deger(1)
 plot_data(96, 13) = deger(2)
 plot_data(96, 14) = deger(3)
 plot_data(96, 15) = deger(4)
 plot_data(96, 16) = deger(5)
 plot_data(96, 17) = deger(6)
 plot_data(96, 18) = sprtb(1)
 plot_data(96, 19) = sprtb(2)
 plot_data(96, 20) = sprtb(3)
 plot_data(96, 21) = darray(nplace(1), 3)
 plot_data(96, 22) = darray(nplace(2), 3)
 plot_data(96, 23) = darray(nplace(3), 3)
 plot_data(96, 24) = excl(1)
 plot_data(96, 25) = excl(2)
 plot_data(96, 26) = excl(3)


 gauge2(0).Value = deger(8)
 gauge2(1).Value = deger(9)
 gauge2(2).Value = deger(10)
 gauge1.Value = deger(7)

 mes(1).Caption = Format$(deger(8), "####.#" + Chr$(34) + " Psig" + Chr$(34))
 mes(2).Caption = Format$(deger(9), "####.#" + Chr$(34) + " Psig" + Chr$(34))
 mes(3).Caption = Format$(deger(10), "####.#" + Chr$(34) + " Psig" + Chr$(34))
 mes(0).Caption = Format$(deger(7) / 100#, "##.## %")

 est(1).Caption = Format$(lev_pem / 100#, "##.## %")
 est(2).Caption = Format$(lev_ann / 100#, "##.## %")
 est(3).Caption = Format$(lev_kft / 100#, "##.## %")
 est(4).Caption = Format$(xestmt, "####.#" + Chr$(34) + " Psig" + Chr$(34))
 est(5).Caption = Format$(pre_pem, "####.#" + Chr$(34) + " Psig" + Chr$(34))
 est(6).Caption = Format$(pre_ann, "####.#" + Chr$(34) + " Psig" + Chr$(34))
 est(7).Caption = Format$(pre_kft, "####.#" + Chr$(34) + " Psig" + Chr$(34))

Rem fuz

 Rem level signal

 buyuk = Abs(lev_kft - deger(7))

 Select Case buyuk
 Case Is > 3#
  gor = 4
 Case 1.501 To 3#
  gor = 3
 Case 1.001 To 1.5
  gor = 2
 Case .401 To 1#
  gor = 1
 Case Else
  gor = 0
 End Select
```

```
 If ilk = 1 Then gor = 2
 Picture1(fuzlev).Visible = 0
 fuzlev = gor
 Picture1(fuzlev).Visible = -1
 plot_data(96, 27) = fuzlev

 Rem pressure signals
 For i = 0 To 2
  buyuk = Abs(xestmt - deger(8 + i))

  Select Case buyuk
  Case Is > 15#
   gor = 4
  Case 10.001 To 15#
   gor = 3
  Case 7.001 To 10#
   gor = 2
  Case 4.001 To 7#
   gor = 1
  Case Else
   gor = 0
  End Select
  Picture1(fuzpre(i + 1) + 5 * (i + 1)).Visible = 0
  fuzpre(i + 1) = gor
  Picture1(fuzpre(i + 1) + 5 * (i + 1)).Visible = -1
  plot_data(96, 28 + i) = gor
 Next i

 For i = 1 To 96
  For j = 1 To 17
   plot.Graph1.ThisSet = j
   plot.Graph1.ThisPoint = i
    If going_to_plot(j) Then plot.Graph1.GraphData = plot_data(i, j) / norm_data(j) Else
plot.Graph1.GraphData = plot.Graph1.YAxisMin
  Next j
  For j = 18 To 20
   gcc_plot.Graph1.ThisSet = j - 17
   gcc_plot.Graph1.ThisPoint = i
   gcc_plot.Graph1.GraphData = plot_data(i, j)
  Next j
  For j = 21 To 23
   gcc_plot.Graph2.ThisSet = j - 20
   gcc_plot.Graph2.ThisPoint = i
   gcc_plot.Graph2.GraphData = plot_data(i, j)
  Next j
  For j = 24 To 26
   gcc_plot.Graph3.ThisSet = j - 23
   gcc_plot.Graph3.ThisPoint = i
   gcc_plot.Graph3.GraphData = plot_data(i, j)
  Next j
  For j = 27 To 30
   gcc_plot.Graph4.ThisSet = j - 26
   gcc_plot.Graph4.ThisPoint = i
   gcc_plot.Graph4.GraphData = plot_data(i, j)
  Next j
 Next i

 If (deger(7) > alarm(1, 4)) Or (deger(7) < alarm(1, 1)) Then
  gauge1.ForeColor = RGB(0, 255, 0)
 Else
  If (alarm(1, 1) <= deger(7) And deger(7) < alarm(1, 2)) Or (alarm(1, 3) < deger(7) And
deger(7) <= alarm(1, 4)) Then
   gauge1.ForeColor = RGB(255, 130, 0)
  Else
   gauge1.ForeColor = RGB(255, 0, 0)
  End If
 End If

 For i = 0 To 2
  If (deger(8 + i) > alarm(2, 4)) Or (deger(8 + i) < alarm(2, 1)) Then
   gauge2(i).ForeColor = RGB(0, 255, 0)
  Else
    If (alarm(2, 1) <= deger(8 + i) And deger(8 + i) < alarm(2, 2)) Or (alarm(2, 3) <
deger(8 + i) And deger(8 + i) <= alarm(2, 4)) Then
    gauge2(i).ForeColor = RGB(255, 130, 0)
   Else
     gauge2(i).ForeColor = RGB(255, 0, 0)
   End If
```

```
  End If
 Next i

 plot.Graph1.DrawMode = 3
 gcc_plot.Graph1.DrawMode = 3
 gcc_plot.Graph2.DrawMode = 3
 gcc_plot.Graph3.DrawMode = 3
 gcc_plot.Graph4.DrawMode = 3

 Print #2, zaman$,
 For i = 1 To 30
  Print #2, plot_data(96, i),
 Next i
 Print #2,
 End If
End Sub

VERSION 2.00
Begin Form About
   BorderStyle     =   1  'Fixed Single
   Caption         =   "About SVS"
   ClientHeight    =   3690
   ClientLeft      =   3015
   ClientTop       =   1965
   ClientWidth     =   3720
   ControlBox      =   0   'False
   Height          =   4095
   Left            =   2955
   LinkMode        =   1  'Source
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   3690
   ScaleWidth      =   3720
   Top             =   1620
   Width           =   3840
   Begin PictureBox Picture1
      BorderStyle     =   0  'None
      Height          =   495
      Left            =   1560
      Picture         =   ABOUT.FRX:0000
      ScaleHeight     =   495
      ScaleWidth      =   495
      TabIndex        =   7
      Top             =   3120
      Width           =   495
   End
   Begin Timer Timer1
      Interval        =   15000
      Left            =   120
      Top             =   3120
   End
   Begin Label Label1
      Alignment       =   2  'Center
      Caption         =   "August 1994"
      Height          =   255
      Index           =   8
      Left            =   0
      TabIndex        =   5
      Top             =   2760
      Width           =   3615
   End
   Begin Label Label1
      Alignment       =   2  'Center
      Caption         =   "Nuclear Engineering Department"
      Height          =   255
      Index           =   7
      Left            =   0
      TabIndex        =   4
      Top             =   2400
      Width           =   3615
   End
   Begin Label Label1
      Alignment       =   2  'Center
      Caption         =   "The University of Tennessee, Knoxville"
      Height          =   255
      Index           =   6
      Left            =   0
```

```
      TabIndex        =   3
      Top             =   2040
      Width           =   3615
   End
   Begin Label Label1
      Alignment       =   2  'Center
      Caption         =   "Ali S. Erbay"
      Height          =   255
      Index           =   5
      Left            =   0
      TabIndex        =   2
      Top             =   1680
      Width           =   3615
   End
   Begin Label Label1
      Alignment       =   2  'Center
      Caption         =   "by"
      Height          =   255
      Index           =   4
      Left            =   0
      TabIndex        =   1
      Top             =   1320
      Width           =   3615
   End
   Begin Label Label1
      Alignment       =   2  'Center
      Caption         =   "Version 2.0"
      Height          =   255
      Index           =   9
      Left            =   0
      TabIndex        =   6
      Top             =   720
      Width           =   3615
   End
   Begin Label Label1
      Alignment       =   2  'Center
      Caption         =   "SIGNAL VALIDATION SYSTEM"
      Height          =   375
      Index           =   0
      Left            =   0
      TabIndex        =   0
      Top             =   240
      Width           =   3615
   End
End
Sub Timer1_Timer ()
 Unload about
End Sub

VERSION 2.00
Begin Form Norm
   BackColor       =   &H00C0C0C0&
   BorderStyle     =   3  'Fixed Double
   Caption         =   "Normalize Plot with"
   ClientHeight    =   3735
   ClientLeft      =   495
   ClientTop       =   1275
   ClientWidth     =   5115
   ControlBox      =   0   'False
   Height          =   4140
   Left            =   435
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   3735
   ScaleWidth      =   5115
   Top             =   930
   Width           =   5235
   Begin SSCommand Command3D2
      Caption         =   "Cancel"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   2640
      TabIndex        =   25
      Top             =   3120
      Width           =   1215
   End
   Begin SSCommand Command3D1
```

```
      Caption         =   "OK"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   1320
      TabIndex        =   24
      Top             =   3120
      Width           =   1215
   End
   Begin TextBox Text1
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      ForeColor       =   &H00FF0000&
      Height          =   285
      Index           =   7
      Left            =   3480
      TabIndex        =   15
      Top             =   2640
      Width           =   855
   End
   Begin TextBox Text1
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      ForeColor       =   &H00FF0000&
      Height          =   285
      Index           =   6
      Left            =   3480
      TabIndex        =   14
      Top             =   2280
      Width           =   855
   End
   Begin TextBox Text1
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      ForeColor       =   &H00FF0000&
      Height          =   285
      Index           =   5
      Left            =   3480
      TabIndex        =   13
      Top             =   1920
      Width           =   855
   End
   Begin TextBox Text1
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      ForeColor       =   &H00FF0000&
      Height          =   285
      Index           =   4
      Left            =   3480
      TabIndex        =   12
      Top             =   1560
      Width           =   855
   End
   Begin TextBox Text1
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      ForeColor       =   &H00FF0000&
      Height          =   285
      Index           =   3
      Left            =   3480
      TabIndex        =   11
      Top             =   1200
      Width           =   855
   End
   Begin TextBox Text1
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      ForeColor       =   &H00FF0000&
      Height          =   285
      Index           =   2
      Left            =   3480
      TabIndex        =   10
      Top             =   840
      Width           =   855
   End
   Begin TextBox Text1
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      ForeColor       =   &H00FF0000&
```

```
      Height          =   285
      Index           =   1
      Left            =   3480
      TabIndex        =   9
      Top             =   480
      Width           =   855
   End
   Begin TextBox Text1
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      ForeColor       =   &H00FF0000&
      Height          =   285
      Index           =   0
      Left            =   3480
      TabIndex        =   8
      Top             =   120
      Width           =   855
   End
   Begin Label Label16
      BackColor       =   &H00C0C0C0&
      Caption         =   "Psig"
      Height          =   255
      Left            =   4440
      TabIndex        =   23
      Top             =   2640
      Width           =   615
   End
   Begin Label Label15
      BackColor       =   &H00C0C0C0&
      Caption         =   "Kbh"
      Height          =   255
      Left            =   4440
      TabIndex        =   22
      Top             =   2280
      Width           =   615
   End
   Begin Label Label14
      BackColor       =   &H00C0C0C0&
      Caption         =   "Kbh"
      Height          =   255
      Left            =   4440
      TabIndex        =   21
      Top             =   1920
      Width           =   615
   End
   Begin Label Label13
      BackColor       =   &H00C0C0C0&
      Caption         =   "DegF"
      Height          =   255
      Left            =   4440
      TabIndex        =   20
      Top             =   1560
      Width           =   615
   End
   Begin Label Label12
      BackColor       =   &H00C0C0C0&
      Caption         =   "DegF"
      Height          =   255
      Left            =   4440
      TabIndex        =   19
      Top             =   1200
      Width           =   615
   End
   Begin Label Label11
      BackColor       =   &H00C0C0C0&
      Caption         =   "DegF"
      Height          =   255
      Left            =   4440
      TabIndex        =   18
      Top             =   840
      Width           =   615
   End
   Begin Label Label10
      BackColor       =   &H00C0C0C0&
      Caption         =   "Psig"
      Height          =   255
      Left            =   4440
      TabIndex        =   17
```

```
      Top             =   480
      Width           =   615
   End
   Begin Label Label9
      BackColor       =   &H00C0C0C0&
      Caption         =   "%"
      Height          =   255
      Left            =   4440
      TabIndex        =   16
      Top             =   120
      Width           =   615
   End
   Begin Label Label8
      BackColor       =   &H00C0C0C0&
      Caption         =   "1P0403A                                         :"
      Height          =   375
      Left            =   120
      TabIndex        =   7
      Top             =   2640
      Width           =   3255
   End
   Begin Label Label7
      BackColor       =   &H00C0C0C0&
      Caption         =   "1F0405A                                         :"
      Height          =   375
      Left            =   120
      TabIndex        =   6
      Top             =   2280
      Width           =   3255
   End
   Begin Label Label6
      BackColor       =   &H00C0C0C0&
      Caption         =   "1F0403A                                         :"
      Height          =   375
      Left            =   120
      TabIndex        =   5
      Top             =   1920
      Width           =   3255
   End
   Begin Label Label5
      BackColor       =   &H00C0C0C0&
      Caption         =   "1T0418A                                         :"
      Height          =   375
      Left            =   120
      TabIndex        =   4
      Top             =   1560
      Width           =   3255
   End
   Begin Label Label4
      BackColor       =   &H00C0C0C0&
      Caption         =   "1T0419A                                         :"
      Height          =   375
      Left            =   120
      TabIndex        =   3
      Top             =   1200
      Width           =   3255
   End
   Begin Label Label3
      BackColor       =   &H00C0C0C0&
      Caption         =   "1T0403A                                         :"
      Height          =   375
      Left            =   120
      TabIndex        =   2
      Top             =   840
      Width           =   3255
   End
   Begin Label Label2
      BackColor       =   &H00C0C0C0&
      Caption         =   "Steam Generator Pressure                :"
      Height          =   255
      Left            =   120
      TabIndex        =   1
      Top             =   480
      Width           =   3255
   End
   Begin Label Label1
      BackColor       =   &H00C0C0C0&
      Caption         =   "Steam Generator Wide Range Level:"
```

```
      Height          =   255
      Left            =   120
      TabIndex        =   0
      Top             =   120
      Width           =   3255
   End
End

Sub Command3D1_Click ()
 Open "norm.dat" For Output As #5
 norm_data(1) = Val(Text1(0).Text)
 norm_data(2) = Val(Text1(1).Text)
 norm_data(3) = Val(Text1(1).Text)
 norm_data(4) = Val(Text1(1).Text)
 norm_data(5) = Val(Text1(1).Text)
 norm_data(6) = Val(Text1(0).Text)
 norm_data(7) = Val(Text1(1).Text)
 norm_data(8) = Val(Text1(0).Text)
 norm_data(9) = Val(Text1(1).Text)
 For i = 0 To 7
  norm_data(i + 10) = Val(Text1(i).Text)
  Print #5, norm_data(i + 10)
 Next i
 Close #5
 For i = 1 To 96
  For j = 1 To 17
   plot.graph1.ThisSet = j
   plot.graph1.ThisPoint = i
    If going_to_plot(j) Then plot.graph1.GraphData = plot_data(i, j) / norm_data(j) Else
plot.graph1.GraphData = plot.graph1.YAxisMin
  Next j
 Next i
 plot.graph1.drawmode = 3
 norm.Hide
End Sub

Sub Command3D2_Click ()
 For i = 0 To 7
  norm.Text1(i).Text = norm_data(i + 10)
 Next i
 norm.Hide
End Sub

VERSION 2.00
Begin Form plot
   BackColor       =   &H00C0C0C0&
   Caption         =   "History of SV for Steam Generator A - Unit 1"
   ClientHeight    =   8715
   ClientLeft      =   -15
   ClientTop       =   300
   ClientWidth     =   12000
   Height          =   9120
   Icon            =   PLOT.FRX:0000
   Left            =   -75
   LinkTopic       =   "Form2"
   MaxButton       =   0   'False
   ScaleHeight     =   8715
   ScaleWidth      =   12000
   Top             =   -45
   Width           =   12120
   Begin SSCommand Command3D6
      Caption         =   "More GCC"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   2640
      TabIndex        =   6
      Top             =   8160
      Width           =   1215
   End
   Begin SSCommand Command3D1
      Caption         =   "Main"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   1320
      TabIndex        =   5
      Top             =   8160
      Width           =   1215
   End
```

```
   Begin SSCommand Command3D4
      Caption         =   "Y-Axis"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   6600
      TabIndex        =   4
      Top             =   8160
      Width           =   1215
   End
   Begin SSCommand Command3D3
      Caption         =   "Nominal"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   5280
      TabIndex        =   3
      Top             =   8160
      Width           =   1215
   End
   Begin SSCommand Command3D2
      Caption         =   "Signals"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   3960
      TabIndex        =   2
      Top             =   8160
      Width           =   1215
   End
   Begin GRAPH Graph1
      AsciiFSize      =   "70~70~70~70"
                           AsciiLabel                          =                "DD-MMM-YYYY
HH:MM:SS.ss~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~DD-MMM-YYYY
HH:MM:SS.ss~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~1"
      AsciiLegend     =   "1L0403A~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~1L0403A"
      AsciiPattern    =   "0~0~0~0~0~0~2~2~2~2~2~2~4~4~4~4~4"
      AutoInc         =   0  'Off
      DrawMode        =   3  'Blit
      GraphType       =   6  'Line
      Height          =   7935
      LabelEvery      =   4
      Left            =   120
      NumPoints       =   96
      NumSets         =   17
      PrintStyle      =   1  'Color
      RandomData      =   0  'Off
      TabIndex        =   1
      TickEvery       =   4
      Top             =   120
      Width           =   11775
      YAxisMax        =   1.2
      YAxisMin        =   0.8
      YAxisPos        =   2  'Right
      YAxisStyle      =   2  'User-defined
      YAxisTicks      =   4
   End
   Begin SSPanel Panel3D1
      BackColor       =   &H00C0C0C0&
      BevelInner      =   1  'Inset
      Caption         =   "Last SV at DD-MMM-YYYY HH:MM:SS.ss"
      Font3D          =   1  'Raised w/light shading
      Height          =   495
      Left            =   7920
      TabIndex        =   0
      Top             =   8160
      Width           =   3975
   End
End
Sub Command3D1_Click ()
 plot.WindowState = 1
 display.Show
End Sub

Sub Command3D2_Click ()
 signals.Show
End Sub

Sub Command3D3_Click ()
 norm.Show
End Sub
```

```
Sub Command3D4_Click ()
 yaxis.Show
End Sub

Sub Command3D5_Click ()
 graph1.DrawMode = 5
 printer.EndDoc
End Sub

Sub Command3D6_Click ()
 plot.WindowState = 1
 gcc_plot.Show
End Sub

VERSION 2.00
Begin Form signals
   BackColor       =   &H00C0C0C0&
   BorderStyle     =   3  'Fixed Double
   Caption         =   "Signals to Plot"
   ClientHeight    =   6375
   ClientLeft      =   3705
   ClientTop       =   1845
   ClientWidth     =   2790
   ControlBox      =   0   'False
   Height          =   6780
   Left            =   3645
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   6375
   ScaleWidth      =   2790
   Top             =   1500
   Width           =   2910
   Begin SSCheck Check3D2
      Caption         =   "Inputs:"
      Font3D          =   0  'None
      Height          =   375
      Left            =   120
      TabIndex        =   14
      Top             =   4080
      Width           =   2535
   End
   Begin SSCheck Check3D1
      Caption         =   "KFT Pressure Estimate"
      Font3D          =   0  'None
      Height          =   375
      Index           =   10
      Left            =   120
      TabIndex        =   13
      Top             =   3720
      Width           =   2535
   End
   Begin SSCheck Check3D1
      Caption         =   "KFT Level Estimate"
      Font3D          =   0  'None
      Height          =   375
      Index           =   9
      Left            =   120
      TabIndex        =   12
      Top             =   3360
      Width           =   2535
   End
   Begin SSCheck Check3D1
      Caption         =   "ANN Pressure Estimate"
      Font3D          =   0  'None
      Height          =   375
      Index           =   8
      Left            =   120
      TabIndex        =   11
      Top             =   3000
      Width           =   2535
   End
   Begin SSCheck Check3D1
      Caption         =   "ANN Level Estimate"
      Font3D          =   0  'None
      Height          =   375
      Index           =   7
```

```
      Left            =   120
      TabIndex        =   10
      Top             =   2640
      Width           =   2535
   End
   Begin SSCheck Check3D1
      Caption         =   "PEM Pressure Estimate"
      Font3D          =   0  'None
      Height          =   375
      Index           =   6
      Left            =   120
      TabIndex        =   9
      Top             =   2280
      Width           =   2535
   End
   Begin SSCheck Check3D1
      Caption         =   "PEM Level Estimate"
      Font3D          =   0  'None
      Height          =   375
      Index           =   5
      Left            =   120
      TabIndex        =   8
      Top             =   1920
      Width           =   2535
   End
   Begin SSCheck Check3D1
      Caption         =   "GCC Pressure Estimate"
      Font3D          =   0  'None
      Height          =   375
      Index           =   4
      Left            =   120
      TabIndex        =   7
      Top             =   1560
      Width           =   2535
   End
   Begin SSCheck Check3D1
      Caption         =   "1P0402A"
      Font3D          =   0  'None
      Height          =   375
      Index           =   3
      Left            =   120
      TabIndex        =   6
      Top             =   1200
      Width           =   2535
   End
   Begin SSCheck Check3D1
      Caption         =   "1P0401A"
      Font3D          =   0  'None
      Height          =   375
      Index           =   2
      Left            =   120
      TabIndex        =   0
      Top             =   840
      Width           =   2535
   End
   Begin SSCheck Check3D1
      Caption         =   "1P0400A"
      Font3D          =   0  'None
      Height          =   375
      Index           =   1
      Left            =   120
      TabIndex        =   5
      Top             =   480
      Width           =   2535
   End
   Begin SSCheck Check3D1
      Caption         =   "1L0403A"
      Font3D          =   0  'None
      Height          =   375
      Index           =   0
      Left            =   120
      TabIndex        =   4
      Top             =   120
      Width           =   2535
   End
   Begin SSCommand Command3D2
      Caption         =   "Cancel"
      Font3D          =   3  'Inset w/light shading
```

```
      Height          =   495
      Left            =   1440
      TabIndex        =   3
      Top             =   5760
      Width           =   1095
   End
   Begin SSCommand Command3D1
      Caption         =   "OK"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   240
      TabIndex        =   2
      Top             =   5760
      Width           =   1095
   End
   Begin Label Label1
      BackColor       =   &H00C0C0C0&
      Caption         =   "1T0406A 1T0419A 1T0418A 1F0403A 1F0405A 1P0403A"
      Height          =   1215
      Left            =   600
      TabIndex        =   1
      Top             =   4440
      Width           =   855
   End
End
Sub Check3D2_Click (Value As Integer)
 If Value Then
  If check3d1(4).Value Then
   check3d1(1).Value = -1
   check3d1(2).Value = -1
   check3d1(3).Value = -1
  End If
 End If
 If check3d1(5).Value Then check3d1(0).Value = -1
 If check3d1(6).Value Then check3d1(3).Value = -1
End Sub

Sub Command3D1_Click ()
 For i = 0 To 10
  going_to_plot(i + 1) = check3d1(i).Value
 Next i
 If check3d1(11).Value Then
  old_inp_checked = -1
  If check3d1(5).Value Then
   going_to_plot(16) = -1
   going_to_plot(17) = -1
  End If
  If check3d1(6).Value Then
   going_to_plot(13) = -1
   going_to_plot(14) = -1
   going_to_plot(16) = -1
  End If
  For j = 7 To 10
   If check3d1(j) Then
    For i = 12 To 17
     going_to_plot(i) = -1
    Next i
   End If
  Next j
 Else
  For i = 12 To 17
   going_to_plot(i) = 0
  Next i
 End If
 plot.Graph1.ThisSet = 1
 plot.Graph1.ThisPoint = 1
 If check3d1(0).Value Then plot.Graph1.LegendText = "1L0403A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.ThisSet = 2
 plot.Graph1.ThisPoint = 2
 If check3d1(1).Value Then plot.Graph1.LegendText = "1P0400A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.ThisSet = 3
 plot.Graph1.ThisPoint = 3
 If check3d1(2).Value Then plot.Graph1.LegendText = "1P0401A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.ThisSet = 4
 plot.Graph1.ThisPoint = 4
```

```
 If check3d1(3).Value Then plot.Graph1.LegendText = "1P0402A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.ThisSet = 5
 plot.Graph1.ThisPoint = 5
  If check3d1(4).Value Then plot.Graph1.LegendText = "GCC Pressure Estimate" Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 6
 plot.Graph1.ThisPoint = 6
  If check3d1(5).Value Then plot.Graph1.LegendText = "PEM Level Estimate" Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 7
 plot.Graph1.ThisPoint = 7
  If check3d1(6).Value Then plot.Graph1.LegendText = "PEM Pressure Estimate" Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 8
 plot.Graph1.ThisPoint = 8
  If check3d1(7).Value Then plot.Graph1.LegendText = "ANN Level Estimate" Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 9
 plot.Graph1.ThisPoint = 9
  If check3d1(8).Value Then plot.Graph1.LegendText = "ANN Pressure Estimate" Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 10
 plot.Graph1.ThisPoint = 10
  If check3d1(9).Value Then plot.Graph1.LegendText = "KFT Level Estimate" Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 11
 plot.Graph1.ThisPoint = 11
  If check3d1(10).Value Then plot.Graph1.LegendText = "KFT Pressure Estimate" Else
plot.Graph1.LegendText = ""
 plot.Graph1.ThisSet = 12
 plot.Graph1.ThisPoint = 12
 If going_to_plot(12) Then plot.Graph1.LegendText = "1T0406A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.ThisSet = 13
 plot.Graph1.ThisPoint = 13
 If going_to_plot(13) Then plot.Graph1.LegendText = "1T0419A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.ThisSet = 14
 plot.Graph1.ThisPoint = 14
 If going_to_plot(14) Then plot.Graph1.LegendText = "1T0418A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.ThisSet = 15
 plot.Graph1.ThisPoint = 15
 If going_to_plot(15) Then plot.Graph1.LegendText = "1F0403A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.ThisSet = 16
 plot.Graph1.ThisPoint = 16
 If going_to_plot(16) Then plot.Graph1.LegendText = "1F0405A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.ThisSet = 17
 plot.Graph1.ThisPoint = 17
 If going_to_plot(17) Then plot.Graph1.LegendText = "1P0403A" Else plot.Graph1.LegendText
= ""
 plot.Graph1.DrawMode = 3
 signals.Hide
End Sub

Sub Command3D2_Click ()
 For i = 0 To 10
  check3d1(i).Value = going_to_plot(i + 1)
 Next i
 If old_inp_checked Then check3d1(11) = -1
 signals.Hide
End Sub

VERSION 2.00
Begin Form yaxis
   BackColor       =   &H00C0C0C0&
   BorderStyle     =   3  'Fixed Double
   Caption         =   "Display Plot in the Range of"
   ClientHeight    =   1440
   ClientLeft      =   7710
   ClientTop       =   3675
   ClientWidth     =   3030
   ControlBox      =   0   'False
   Height          =   1845
   Left            =   7650
```

```
   LinkTopic       =   "Form3"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   1440
   ScaleWidth      =   3030
   Top             =   3330
   Width           =   3150
   Begin SSCommand Command3D2
      Caption         =   "Cancel"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   1560
      TabIndex        =   5
      Top             =   840
      Width           =   1215
   End
   Begin SSCommand Command3D1
      Caption         =   "OK"
      Font3D          =   3  'Inset w/light shading
      Height          =   495
      Left            =   240
      TabIndex        =   4
      Top             =   840
      Width           =   1215
   End
   Begin TextBox Text2
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      ForeColor       =   &H00FF0000&
      Height          =   285
      Left            =   2400
      TabIndex        =   3
      Top             =   480
      Width           =   375
   End
   Begin TextBox Text1
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      ForeColor       =   &H00FF0000&
      Height          =   285
      Left            =   2400
      TabIndex        =   2
      Top             =   120
      Width           =   375
   End
   Begin Label Label2
      BackColor       =   &H00C0C0C0&
      Caption         =   "Y-Axis Maximum Value:"
      Height          =   255
      Left            =   360
      TabIndex        =   1
      Top             =   480
      Width           =   2055
   End
   Begin Label Label1
      BackColor       =   &H00C0C0C0&
      Caption         =   "Y-Axis Minimum Value :"
      Height          =   255
      Left            =   360
      TabIndex        =   0
      Top             =   120
      Width           =   2055
   End
End
Sub Command3D1_Click ()
 plot.Graph1.yaxismin = Val(text1.text)
 plot.Graph1.yaxismax = Val(text2.text)
 plot.Graph1.DrawMode = 3
 yaxis.Hide
End Sub

Sub Command3D2_Click ()
 text1.text = plot.Graph1.yaxismin
 text2.text = plot.Graph1.yaxismax
 yaxis.Hide
End Sub

Sub Form_Load ()
```

```
 text1.text = plot.Graph1.yaxismin
 text2.text = plot.Graph1.yaxismax
End Sub

Global mchan(5), alpha, beta, bounda, boundb
Global erb(5), var0(5), bias(5), nexcl(5)
Global mstdd(5), mstdn(5), sii(5), sum(5), sd(5), snum(5)
Global meas(5), darray(5, 3), sprtb(5), excl(5), tm(5, 1000)
Global BSETTO0(5), nplace(5), ninclp(5), dbias(5), nbias(5)
Global index(5), x(5)
Global nsignl, k, npoint, itest, xestmt
Global jinclp, jexclp
Global isig(5) As Integer
Global deger(12), eskideger(4)
Global g(2, 2), po(2, 2), p(2, 2), tmp(2, 2), tmpi(2, 2), f(2, 2)
Global going_to_plot(17), plot_data(96, 30), norm_data(17)
Global decision(96, 4)
Global filename$, numskip, zaman$, eskizaman$
Global alarm(2, 4), old_inp_check
```

APPENDIX F

Code Listing for Input / Output Interface

```
C
C       OUTVALUE - UT PROGRAM
C
        IMPLICIT INTEGER*4 (A-Z)

        INCLUDE      '($IODEF)'
        INCLUDE      '($SSDEF)'

        INCLUDE 'DATA0:[SAIPMS.INCLUDE] SYSPAR.I    '
        INCLUDE 'DATA0:[SAIPMS.INCLUDE] MESTAB.I    '
        INCLUDE 'DATA0:[SAIPMS.INCLUDE] POINTS.I    '
        INCLUDE 'DATA0:[SAIPMS.INCLUDE] CVTTAB.I    '

        INTEGER*4       IRETURN
        INTEGER*4       MESPTR          ! INDEX FOR POINT
        INTEGER*4       CYCLE

C       MAXIMUM NUMBER OF POINTS IS IMAX

        PARAMETER (IMAX=200)

        character*23 datetime
        CHARACTER*8     ZZPID(1:IMAX),ZPID
        REAL            OUTVAL(1:IMAX),deltat
        integer*2       outqua(1:IMAX)

        OPEN (1,FILE='OUTVALUE.INP',TYPE='UNKNOWN')
        OPEN (2,TYPE='UNKNOWN',FILE='OUTVALUE.OUT',
     +          CARRIAGECONTROL='LIST',SHARED)
        REWIND(1)
        REWIND(2)
        NUMPTS=0
        read(1,*) deltat
        DO 1 I=1,IMAX
                READ  (1,'(A8)',END=310) ZZPID(I)
                NUMPTS=NUMPTS+1
1       CONTINUE
310     CONTINUE

        CLOSE (UNIT=1)

1000    CONTINUE

        status=lib$date_time(datetime)


        DO 2 I=1,NUMPTS

         ZPID=ZZPID(I)

         CALL GNVERMES (IRETURN, ZPID, MESPTR)

         IF (MESPTR .NE. -1 )THEN

C        WRITE (5,*)    'TYPE: ',IME_TYPE(MESPTR)

                   IF (IME_TYPE(MESPTR) .EQ. IDA_EXTR) THEN

               outqua(I)=ICV_QUAL (MESPTR)

               OUTVAL(I)=XCV_EU (MESPTR)

            ELSE IF (IME_TYPE(MESPTR) .EQ. 8) THEN

               outqua(I)=ICV_QUAL (MESPTR)
```

```
            OUTVAL(I)=XCV_EU (MESPTR)

        ELSE IF (IME_TYPE(MESPTR) .EQ. IDA_EXTL) THEN

            outqua(I)=ICV_QUAL (MESPTR)

            OUTVAL(I)=ICV_EU (MESPTR)

        ELSE
            OUTVAL(I)=0.0
            outqua(I)=0
        END IF
    END IF

2   CONTINUE

    REWIND(2)

    write(2,*) datetime

    DO 3 I=1,NUMPTS
     WRITE(2,*) OUTVAL(I),outqua(I)
3   CONTINUE

    CALL LIB$WAIT(deltat)

    GOTO 1000

    END
```

VITA

Ali Seyfettin Erbay was born in Bruchsal, Germany on July 19, 1967. He finished Schönbornschule and Stirumschule primary schools in Germany and came to Türkiye in 1978. He finished Beylerbeyi Primary School and attended Beylerbeyi Middle School for one year. Thereafter, he moved to Izmir with his family, where he finished Esrefpasa Middle School and Atatürk High School. In 1985, he entered the Nuclear Engineering Department of Hacettepe University in Ankara, Türkiye. He received his Bachelor of Science degree in Nuclear Engineering in June 1990. In September 1990, he entered the Master of Science program of the Computer Science and Engineering Department at Hacettepe University, where he also worked as a research assistant until December 1991. In January 1992, he entered the Nuclear Engineering Department of The University of Tennessee, and in December 1994, he received his Master of Science degree in Nuclear Engineering. He is presently a research assistant and a doctoral student in the same department.