INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

ANALYSIS AND MONITORING OF A CANDU NUCLEAR POWER PLANT USING MULTIVARIATE STATISTICAL PROCESS CONTROL METHODS

By

ROBERT P. LEGER, B.Sc. Eng., Ms.Eng.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

© Copyright by Robert P. Leger, August 1999

ANALYSIS AND MONITORING OF A CANDU NUCLEAR POWER PLANT USING MULTIVARIATE STATISTICAL PROCESS CONTROL METHODS

Doctor of Philosophy (1999)
Department of Engineering Physics

McMaster University
Hamilton, Ontario

TITLE:	Analysis and Monitoring of a CANDU Nuclear Power Plant Using Multivariate Statistical Process Control Methods	
AUTHOR:	Robert P. Leger,	M.Eng. (McMaster University) B.Sc.Eng. (University of New Brunswick)
SUPERVISOR:	Dr. Wm. J. Garland	
NUMBER OF PAGES:	x, 241	

ABSTRACT

In the nuclear power industry, the ability to efficiently analyse historical data, and to detect and diagnose process faults in a timely manner are critical tasks in operating and maintaining a nuclear power plant. The objectives of this research were to prove that established Statistical Process Control (SPC) techniques could be used to analyse nuclear power plant data and to develop a hierarchical process monitoring methodology which could deliver relevant information to different functional groups within a plant.

The use of established Multivariate SPC techniques to analyse nuclear power plant data was successfully proven in several areas and is considered a significant contribution to the development of data analysis tools for the nuclear energy industry. By analysing actual data from an operational nuclear power plant, the techniques were able to provide key insights into the process. Process tests and different plant configurations were easily identified. The multivariate techniques could relate the different plant configurations to sensor calibrations and process changes. Also, the techniques were able to identify two anomalies in the data which were not previously detected using the existing analysis tools.

In order to produce a hierarchical process monitoring methodology, a multi-block, multi-level Principal Component Analysis algorithm, and associated prediction code, was

developed and tested. This algorithm is an extension of existing multi-block, two-level algorithms and represents a contribution to the current state of Multivariate SPC techniques. It was found that the algorithm was very useful for analysis but marginal for delivering relevant information in a process monitoring capacity. This finding resulted in the third major contribution of this research work. The Multivariate SPC techniques are very useful for analysing nuclear power plant data but not as feasible for monitoring the process in an on-line manner. This was attributed to the goals defined for the monitoring methodology, the scaling method used for the data, and the numerous normal plant operating states.

ACKNOWLEDGMENTS

I would like to thank my supervisor, Dr. Bill Garland for his guidance, insightful suggestions, ideas and support over the last six years. I could not have picked a better mentor.

I would like to thank the other members of my supervisory committee, Dr. Skip Poehlman and Dr. John MacGregor. Their comments and suggestions throughout this work have been greatly appreciated.

I would like to thank Dr. Frank Steward at CNER. His encouragement and support is very much appreciated. I would also like to thank Heather Payne for her help with the figures.

I am especially grateful to my parents for their encouragement to finish this project.

Finally, I would like to express my special thanks to my family. To Samantha and Danny, who both arrived during the course of this work and who unknowingly brought perspective when it was needed. To my wife, Wendy, words cannot express my appreciation for your support over the past six years. This has truly been a team effort.

Robert Leger

August 1999

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AECL	Atomic Energy of Canada Limited
ANN	Artificial Neural Network
CANDU	CANada Deuterium Uranium
CPCA	Consensus Principal Component Analysis
CUSUM	Cumulative Summation
D_20	Heavy Water
FDD	Fault Detection and Diagnosis
FDI	Fault Detection and Isolation
HPCA	Hierarchical Principal Component Analysis
HTS	Heat Transport System
LDPE	Low-density Polyethylene
MLP	Multi-layer Perceptron
MPLS	Multi-block Partial Least Squares
NPP	Nuclear Power Plant
OHN	Ontario Hydro Nuclear
OM&A	Operations, Maintenance and Administration
PCA	Principal Component Analysis
RBF	Radial Basis Function
RIH	Reactor Inlet Header

ROH	Reactor Outlet Header
S	Covariance Matrix
SDS1	Shutdown System 1
SDS2	Shutdown System 2
SPC	Statistical Process Control
SSexp	Sum of Squares Explained
STD	Standard Deviation
TAMS	Transmitter Accuracy Monitoring System

CHAPTER 1

INTRODUCTION

1.1 Background

Electricity is used in virtually every aspect of modern day life. Electrical energy is used at home, at work, for transportation and entertainment. The modern world has become so dependent on electricity that a U.S. public attitude study in 1982 revealed that 51% of the surveyed people viewed electricity as a basic human right as opposed to a commodity that has to be manufactured [1]. Indeed, one has to look no further than the impact of the severe ice storm in 1998 which left 100,000's of people in eastern Ontario and Quebec without electrical power to confirm this dependency. It is a sobering realization that lives are threatened when they are cut off from electricity.

Although the general public may view electricity as a right, it still needs to be produced in some fashion. There are several methods used to generate electricity including conventional hydro-electric dams, burning fossil fuels (coal, oil, natural gas), nuclear power, solar power and wind power. The debate as to which generation method is best in terms of public safety, environmental impact, renewability, efficiency and cost is on

going. Wyatt presents several comparisons between the different methods of generating electricity [1]. The focus of this research will be on nuclear power, specifically the generation of electricity through the operation of a CANDU 6 nuclear power plant. CANDU 6 nuclear power plants have advantages in both fuel consumption and waste generation [2]. They consistently rank in the top 10 in the world for lifetime performance and in 1996 had the highest lifetime capacity factor for all types of nuclear power plants [2]. The lifetime capacity factor is defined as the total gross generation divided by the capacity divided by the total number of hours from the time of first synchronization [2]. Over the past recent years, some CANDU 6 plants have run into some operational issues. The Point Lepreau Nuclear Power Plant (NPP) experienced an extended unplanned outage due to wood being dispersed in the heat transport system [3]. The wood was left in the system due to inadequate maintenance procedures. In the summer of 1997, Ontario announced that it would shut down 7 of its 19 nuclear reactors as a result of a highly critical internal report. This report contended that Ontario Hydro Nuclear (OHN) failed to shift from an engineering and construction organization to an operation and maintenance organization when the need for electricity decreased [4]. The author of the report, Mr. Carl Andognini, stated that it takes different talents to build than it does to operate and this lack of a shift resulted in declining performance and increased costs.

The issues outlined above lead into the motivation for this research work. Some key aspects in operating and maintaining a NPP, or complex process in general, are the abilities to efficiently analyse historical data collected from the process and convert it

into useful information, monitor the process for faults, and diagnosis faults in a timely manner once they are detected. The analysis of historical data is useful for the detection of slowly developing faults and for post-incident investigations and is obviously done in an off-line manner. The post-incident analysis is very desirable as it can help with diagnosis and possibly identify some future early warning signs for the specific fault. Process monitoring is directed more towards on-line monitoring. In terms of personnel roles in a NPP, historical data analysis is more the role of the system responsible engineer while process monitoring is more of the concern of the control room operator. Performing these functions can lead to an increase in plant safety, availability, and performance while lowering overall operations, maintenance and administration (OM&A) costs [5]. The tools currently used in CANDU 6 plants for surveillance or monitoring and diagnostic tasks have room for improvement. This could be due to the fact that the Canadian nuclear industry is just currently in the process of developing and implementing historical data systems [5]. Once the historical data systems are implemented, the focus will turn to the development of analysis tools. The concepts of analysing historical process data and monitoring processes are not new. Much research has been done in these areas in recent years. The next section will provide an overview of these concepts.

1.2 Overview of Data Analysis and Process Monitoring

1.2.1 Data Analysis

There are several standard tools and techniques which are useful for analysing process data, many related to statistical analysis. They include pareto diagrams, cause and effect diagrams, histograms, sequence plots or run charts and scatter plots. The uses for each tool are summarized in Table 1.1 [6].

Tool	Uses
Pareto Diagrams	identifies high potential opportunities separates 'vital few' from 'trivial many'
Cause-and-Effect Diagram	Describes hierarchy and relationships Is a systematic way to identify potential causes
Histogram	Shows distribution of data Infers information about population - center, dispersion, shape
Run Chart	Displays behavior over time Shows trends and shifts
Scatter Plots	Reveals relationships between pairs of variables Gives insights into patterns in the sample data

Table 1.1: Standard Data Analysis Tools

The most commonly used tool in a nuclear power plant is the run chart. A run chart plots one or more variables against time. To compare variables, a scatter plot is used. An example of a scatter plot is shown in Figure 1.1. This type of plot is useful for identifying correlations in the data. In this example, there is a strong correlation between viscosity and weight, that is, as viscosity increases, so does the weight. One drawback of simple scatter plots is that they can only relate two variables at once. Typically when dealing with a complex process such as a NPP, an analyst will be faced with hundreds if not

thousands of variables. This quickly leads to data overload. One method for analysing datasets with many correlated variables is Principal Component Analysis (PCA) [7]. PCA is basically a technique for transforming a group of correlated variables via linear combinations of the original variables into a new group of uncorrelated variables. It can also be used to reduce the dimension of a data matrix. This method can greatly help in the analysis of large, ill-conditioned datasets, as will be expanded on in Chapter 2.

1.2.2 Process Monitoring

As mentioned previously, one key to improving the operation of a process is the ability to promptly detect and diagnose faults in the process which cause the process variables to move away from their desired values. On the surface, the task of fault detection and diagnosis or isolation (FDD or FDI) would appear to be a relatively straightforward one. A simple strategy for FDD is shown in Figure 1.2. Basically, the process should be monitored in some fashion and when it is not operating correctly, the cause of the fault or problem diagnosed. Once the cause of the fault is known, its severity must be evaluated and a decision is made as to whether the process should be stopped and the fault fixed or if the operation of the process should be changed. This procedure is performed routinely, although informally, in many aspects of everyday life, such as operating an automobile. However, when dealing with complex processes such as nuclear power plants, there are several underlying challenges which need to be addressed, some of which will be outlined in the next section.

1.3 Data Analysis and Process Monitoring Challenges in the Nuclear Industry

1.3.1 Instrumentation Faults versus Process Faults

When attempting to monitor a process, typically one would use all possible sources of information available. In the chemical industry, this would include not only data but also the sight, sound, and perhaps feel of the process. However, due to the nature of nuclear power, there are many areas of the plant which are inaccessible to the operators and engineers. Therefore, they must rely primarily on measured data collected from important process variables using various types of instrumentation. When something appears to be wrong, the obvious question is raised, "Is there actually a problem with the process or is the problem with the instrumentation and sensors used to measure the process variables". The standard method used in the nuclear industry to address this concern is to install redundant sensors for each measured variable. This is commonly referred to as a hardware redundancy. Then a method of voting will be used to determine if a process fault is present. For example, if 2 out of 3 sensors indicate a fault is present, then it would be assumed that there is indeed something wrong with the process. If only one sensor indicates a problem, then that sensor would be considered suspect. However, the drawback of hardware redundancy, apart from the obvious increase in capital cost, maintenance and testing, is that it adds to the number of variables presented to the operators and engineers and therefore adds to data overload.

1.3.2 Data and Information Overload

The issue of data overload has already been mentioned in the previous section. As stated above, the use of redundant hardware can result in 1000's of measurements or data signals that will cause data overload to the analyst in general. However, the effect of data overload can also be examined from the perspective of different functional groups or levels within the plant. For example, an Electrical/Instrumentation technician responsible for detecting small sensor calibration errors for a specific sub-system could become overloaded if he were also monitoring the entire plant for process faults. At the other extreme, the plant manager would not want to be informed every time one sensor in one sub-system needed re-calibration. Generally, the data or information needs to be distilled as it moves up the functional ladder. Also, it is very difficult for the control room operators to assimilate all the measured data and determine, in an on-line manner, if the process is operating correctly. For this reason, typically control room operators will chose to monitor only a few variables which they feel are important and indicative of the entire process. However, in this case, they are discarding some potentially useful information which may be contained in the entire set of measured variables. Ideally, to do effective FDD, this massive amount of measured data must be condensed into useful information and the proper information must be supplied to the different functional levels in the plant.

There are several different methodologies for performing FDD, some of which address some of the challenges outlined above. They are basically divided up into two categories,

those which use mathematical models of the process and those which do not. These methodologies will be discussed in detail in Chapter 2. One specific method which has flourished in the chemical industry is Statistical Process Control (SPC). As SPC will play a significant role in this research work, a brief introduction is presented in the next section.

1.4 Introduction to Statistical Process Control (SPC)

Statistical Process Control (SPC) involves setting up control charts which are used to monitor the process variables for faults. The control charts use control limits which are based on the inherent or "common cause" variability which affects the process variables at all times. This inherent or natural variability is considered a natural part of the process which cannot be eliminated. The task of the control charts is to distinguish between the natural variation in the process, which cannot be avoided, and faults which have an assignable cause. The criterion for measuring the performance of the control charts is based on two types of errors. If the control chart indicates a fault is present when the process is, in fact, in control, a false alarm has occurred. This type of error is known as a Type I error. If the control chart fails to detect a fault which is actually present, a Type II error has occurred. The ideal control chart scheme will minimize both types of errors.

SPC was first developed in the 1930's in the chemical industry. In its early applications, control charts were used to monitor a few product quality variables, such as density, viscosity or colour. These variables were assumed to be independent of each other. For

this reason, it was quite acceptable to use univariate control charts which were manually updated. Any data collected from the process, such as temperatures, pressures or flowrates would simply be ignored. This could be one reason why SPC has had limited applications in the nuclear industry. As stated above, nuclear power plant data consists of thousands of process measurements. Clearly, it would not be feasible to manually track thousands of variables individually and a multivariate method is required. Also compounding the problem is the fact that the process variables are not independent. Typically, there are only a few underlying events driving the process and each measured variable gives a little different information on the events. This causes the process variables to be highly correlated with one another. Hence, the covariance matrix of the process dataset will be nearly singular, as some variables will be approximately scalar multiples of others. This causes computational difficulties for standard multivariate techniques which rely on inverting the covariance matrix of the process dataset. Recently, significant advances have been made in the area of multivariate SPC using projection methods to handle large, ill-conditioned datasets [8]. Again, these advances will be reviewed in detail in Chapter 2. These methods break the dataset down into uncorrelated variables, or principal components, which can then be monitored for assignable cause events. The general method used to develop an SPC monitoring scheme is as follows. First, historical data is collected from the process when operating normally. It is important at this step to remove any data which represent faults that should be detected in the future. Therefore, the data used to develop the monitoring scheme should contain only inherent variability. Next, a statistical model is developed which accurately

describes this process data. Finally, new data can be compared to the model to determine if the process is continuing to operate normally or if there is a fault present. If a fault is detected, the data and/or process must be examined in more detail to diagnose the cause.

1.5 Hypothesis for Research Work

To address the challenges outlined above, three main hypotheses will be investigated in this research work. They will be discussed below.

The first hypothesis is as follows:

"Established Multivariate SPC techniques can be used for the analysis of historical datasets generated from CANDU nuclear power plants."

Some success criteria for this first hypothesis are:

- the techniques should be able to handle the historical data generated from a NPP
- the techniques should provide insight into the operation
- the techniques should allow the user to perform some basic diagnostics on any anomalies found in the data

The second hypothesis is:

" A hierarchical process monitoring methodology can be developed which will have the ability to deliver relevant information to different functional groups within a NPP"

Some success criteria for the second hypothesis are:

- the monitoring system should be sensitive to both small instrumentation faults and overall process faults

- the monitoring system should be able to provide the required information to different functional levels in the NPP
- the monitoring system should allow the user to perform some basic diagnostic tasks in an efficient manner

The final hypothesis is as follows:

"A systematic methodology can be developed to quantify the sensitivity of a specific process monitoring application"

Finally, some success criteria for this third hypothesis are:

- the sensitivity of a specific monitoring model should be estimated using a rigorous, systematic method or approach
- the systematic method should be applicable to any specific monitoring model

1.6 Chapter Organization

Following is a brief description of the contents of the following chapters:

Chapter 2: Literature Review of FDD Techniques

Chapter 2 will provide an overview of the most current techniques for data analysis and FDD and review some applications. It will give a justification for the use of SPC over methods which employ mathematical models of the process.

Chapter 3: CANDU Nuclear Power Plant Design and Project Data

Chapter 3 will describe both the process and actual data collected from an operational CANDU nuclear power. The chapter will also give detailed descriptions of the magnitudes of sensor faults which the monitoring system should be sensitive to.

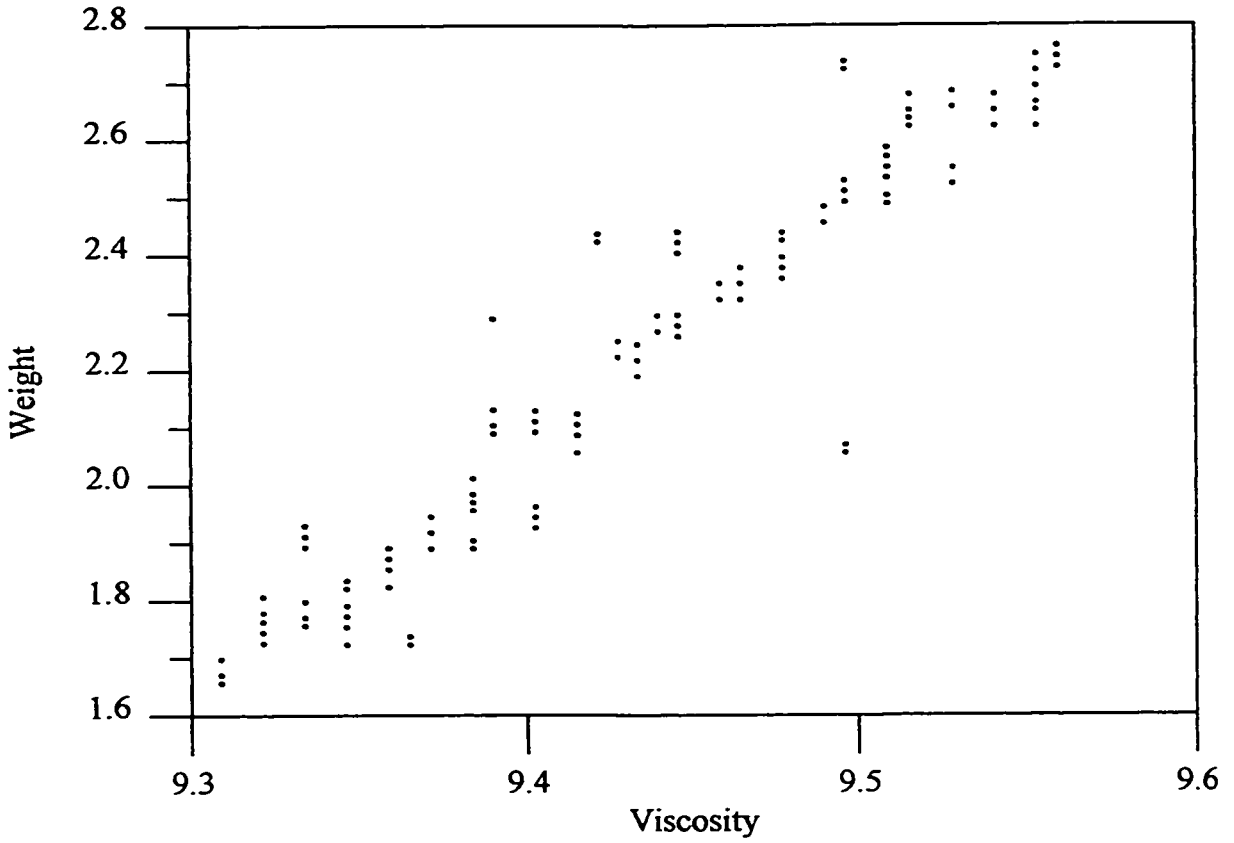
Chapter 4: Development of the Multi-Block, Multi-Level PCA Methodology

This chapter will provide a theoretical development of consensus principal component analysis (CPCA). An extension of CPCA to a multi-level format will also be developed. Finally, a norming and deflation analysis for the CPCA algorithm will be provided.

Chapter 5: NPP Analysis and Monitoring Using Various PCA Techniques

Chapter 5 will focus on three areas. First a preliminary analysis using the standard PCA will be presented. Next, process analysis using the multivariate techniques will be presented. Finally, the results from the process monitoring investigation will be presented. The first two topics above will basically investigate the analysis of historical data while the third topic will investigate process monitoring, including detecting sensor faults vs. process faults.

Chapter 6: Conclusions and Future Work



A Scatter Plot:

- Reveals Relationships Between Pairs of Variables
- Gives Insight Into Patterns in the Sample Data

Figure 1.1 Scatter plot of weight and viscosity

Fault Detection in Nuclear Systems

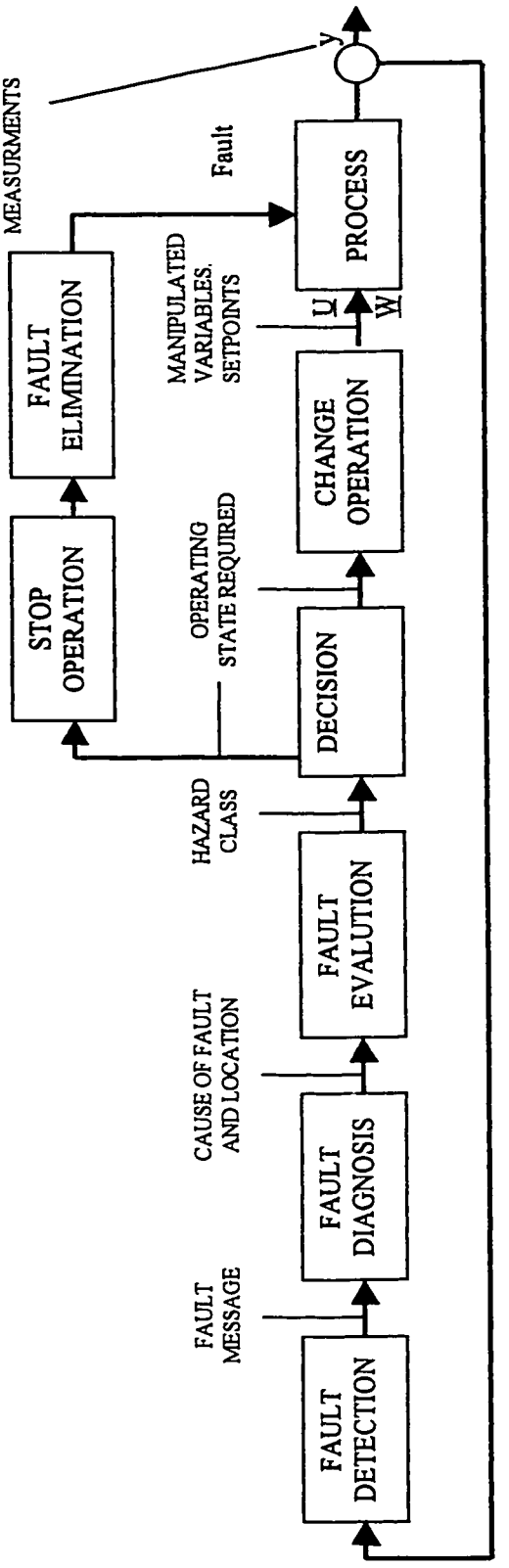


Figure 1.2 Supervision loop (on appearance of a fault).

CHAPTER 2

LITERATURE REVIEW OF FDD TECHNIQUES

2.1 Introduction

As stated in Chapter 1, the task of FDD is a relatively straightforward one. However, when monitoring a complex process, there are usually a large number of process variables that are measured on a very frequent basis. This massive amount of data makes on-line FDD by an operator very difficult. It is also very difficult to manually extract useful information from the large historical databases that are becoming available. These difficulties have led to the development of FDD techniques that can be implemented on the computer. Extensive research has been completed in the area of FDD over the past 20 years resulting in numerous publications. Of note, several survey papers on FDD have been presented [9,10,11,12,13,14]. The most common computer-based technique for FDD is to compare the measured variables of a process to a model of the process. This comparison generates residuals that can be used to detect faults. A very simple diagram of model based FDD is shown in Figure 2.1. The model used to generate the residuals may be one of two basic types. Methods that use models that are derived from first principles are generally referred to as analytical redundancy techniques. In this respect,

the analytical model provides the analytical redundancy much in the same way as additional sensors that measure the same variable provide hardware redundancy. If historical data from the process is used to develop an empirical model and associated control charts, the technique is generally referred to as Statistical Process Control (SPC). Artificial neural networks (ANN) can also be used to generate an empirical model to be used for FDD. If a process model is not available, either analytical or empirical, then other techniques such as simple limit checking or knowledge based systems may be used for FDD. The first part of this chapter will provide a detailed review of all of the techniques listed above, with a particular emphasis on SPC techniques. The second part of the chapter will review applications of FDD techniques that have been reported on in the nuclear industry.

2.2 FDD Background and Techniques

The task of FDD is really a two step process, as shown in Figure 1.2. First the fault must be detected and then it must be diagnosed. There are various methods that can be used for each step. The next sections will review the methods for both detection and diagnosis but first, some relevant terms and definitions will be reviewed.

In a field as well researched as FDD, there is bound to be some confusion with different terms and definitions. Isermann and Balle attempted to clarify the various terms and definitions [14]. Some of the more relevant terms for this research work are listed in Table 2.1. As stated in the previous section, the most common FDD technique is to

compare the output of the process to the output or prediction from a fault-free model of the process and analyse the residuals. The model can be either empirical or analytical. The next sections will review the theoretical background for both types of model techniques as well as techniques which do not use process models.

Term	Meaning
Fault	An unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable / usual / standard condition.
Failure	A permanent interruption of a system's ability to perform a required function under specified operating conditions.
Error	A deviation between a measured or computed value (of an output variable) and the true, specified or theoretically correct value.
Residual	A fault indicator, based on a deviation between measurements and model-equation-based computations.
Fault Detection	Determination of the faults present in a system and the time of detection
Isolation	Determination of the kind, location and time of detection of a fault. Follows fault detection.
Diagnosis	Determination of the kind, size, location and time of detection of a fault. Follows fault detection. Includes fault isolation and identification.
Monitoring	A continuous real-time task of determining the conditions of a physical system, by recording information, recognizing and indicating anomalies in the behavior.
Analytical Redundancy	Use of two or more (but not necessarily identical) ways to determine a variable, where one way uses a mathematical process model in analytical form.

Table 2.1: Terminology Used in the Field of Fault Detection and Diagnosis [adapted from 14]

2.2.1 Statistical Process Control

The basic tool for SPC is the control chart. The control charts and the associated limits are based on an empirical model of the process. The model is developed using historical data. There are several different types of control charts, both univariate and multivariate.

2.2.1.1 Univariate Control Charts

As the name implies, univariate control charts are used to monitor individual variables one at a time. Usually, control charts are used to monitor the mean and standard deviation or range of the variable. The standard deviation and range of the data are measures of the variation or variability in the data. They are calculated as follows:

Range = largest observation in a sample minus the smallest observation

$$\text{Standard Deviation (std)} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \qquad 2.1$$

Where: n = number of observations in the sample

x_i = value of x for observation i

$\bar{x}$ = average of the n observations in the sample

Control charts can be used to monitor individual observations or the data can be divided into subgroups or samples. Dividing the data into samples has certain advantages that will be outlined later. However, the data must be acquired at an acceptable rate in order to make subgrouping feasible.

There are several different choices for univariate control charts, based on the type of data available from the process. There are control charts for both measured numerical data and attribute or count data. The three most common types used for monitoring the mean or target are the Shewhart Control chart, cumulative summation (CUSUM) control chart and the exponentially weighted moving average chart. Dr. W.A. Shewhart, who is considered to be the father of statistical quality control, developed the Shewhart chart in 1931 [18, 56]. It plots successive sample averages and typically has control limits set at $\bar{x} \pm 3\sigma_{\bar{x}}$, where $\bar{x}$ is the overall average of the sample averages and $\sigma_{\bar{x}}$ is an estimate of the standard deviation of the sample averages. An example of a Shewhart chart is shown in Figure 2.2. These charts are effective in quickly detecting large changes in the variable mean, on the order of 1.5 to approximately 2 standard deviations. However, they are relatively insensitive to persistent moderate shifts in the mean, on the order of 1 standard deviation [6]. Quite often, these types of shifts are common and are a first indication that a fault has occurred. The two other charts listed above are better suited for detecting these shifts promptly and accurately. The exponentially weighted moving chart was developed by S.W. Roberts in 1959. This chart plots the value of a statistic, w_t, where $w_t = r * x_t + (1 - r) * w_{t-1}$. In this expression, the weight r has a value between 0 and 1. There are specific formulas for calculating the control limits. The CUSUM chart is a more popular chart for detecting small persistent changes. E.S. Page first introduced this chart in 1954 [15]. As the name implies, this type of chart cumulates deviations of the

sample averages from the target or desired value. Once these cumulations reach either a high or low limit, an out-of-control signal is given. Again, there are specific methods for calculated the control limits for CUSUM charts [16].

As stated in Chapter 1, control charts are developed using historical data from the process. This historical data must contain only natural or common cause variability. Any outliers in the dataset that have assignable causes must be removed before the control charts can be set up. There are two additional assumptions made when setting up control charts in the standard fashion. It is assumed that the observations from the process are normally distributed and independent. The normal distribution assumption is used to calculate control limits. If the observations do not appear to be normally distributed, then two methods can be used to address this. First, if there is a large historical data base, then a reference distribution based on the historical dataset can be used to calculate the control limits [17]. Secondly, the data can be divided into subgroups, as mentioned above, and the control chart limits can be calculated using the subgroup averages and standard deviation. By virtue of the Central Limit Theorem, the distribution of the subgroup averages will be better approximated by a normal distribution than the individual observations. Also, as the size of the subgroups increases, the assumption will become more accurate. For observations to be independent, there must not be either auto-correlation or cross-correlation in the data. Ryan showed the effects of auto-correlated data on the calculation of control chart parameters [18]. Basically, auto-correlation will cause the standard deviation to be under estimated and hence an unacceptable number of

false alarms will be generated. Harris et al suggested some methods for handling auto-correlated data [19]. Cross-correlation can cause faults to be missed in methodologies that assume no cross-correlation. This is shown in Figure 2.3. As observed in Figure 2.3, point YY falls within the control limits for both X1 and X2. However, if X1 is plotted against X2, it is obvious that point YY is different from the other observations. This plot is revealing that for point YY, the correlation between X1 and X2 has been broken. Even with these limitations, the standard univariate control charts are still the most widely used control charts in industry [8].

2.2.1.2 Multivariate Control Charts

There are multivariate extensions to the three basic univariate control charts described in the previous section. These are described by Kourti and will be outlined below [8]. The multivariate extension to the Shewhart chart is based on Hotelling's T2 statistic [7,18].

This is calculated as:

$$T^2 = (y - \bar{y})^T S_y^{-1} (y - \bar{y}) \qquad 2.2$$

Where: y is a multivatiate observation (a vector)

$\bar{y}$ is the mean or target value for y

S_y is the covariance matrix of the historical dataset = $\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})(y_i - \bar{y})^T$

The above equations are for individual observations. If the data has been subgrouped, the above definition of T^2 and its control limit must be redefined [8]. It should be noted that

the solution to the above equations for two variables represents an ellipse that is represented in Figure 2.3.

To extend the CUSUM charts, Crosier proposed calculating T^2 for each new observation and then computing the CUSUM of T2 as [20]:

$$C_i = \max[0, C_{i-1} + T_i - k]$$

where :

$$C_0 = 0$$

$$T_i = \sqrt{T^2} = \sqrt{(y_i - \bar{y})^T S_y^{-1} (y_i - \bar{y})}$$

k = is the allowable slack in the process, similar to the univariate CUSUM

An out of control signal is generated when C_i becomes greater than a certain limit, h. Finally, Lowry proposed an extension to the exponentially weighted moving average chart [21]. All of the above multivariate control charts involve inverting the covariance matrix (S) obtained from the historical dataset. Normally, the process variables will not be independent of one another. Usually, there are only a few underlying events driving the process and each measured variable contributes a little different information on the events. This causes the data matrix to be less than full rank, meaning the rank or number of independent columns is less than the number of variables. Hence the covariance matrix will be singular or very ill conditioned and cannot be inverted. Also, process data will typically contain lots of holes where the measured data were not available. This missing data in either the historical dataset or the new observations will cause the computations to crash if it is not handled in a proper fashion. There are a number of projection methods that can be used for dealing with large, ill-conditioned datasets by reducing the dimension of the problem. One method that has recently received much

attention in multivariate SPC methods is Principal Component Analysis (PCA). This method breaks the dataset down into uncorrelated variables, or principal components, which can be monitored for faults. The next section will describe PCA and how it can be used to detect faults.

2.2.1.3 Multivariate SPC Based on PCA

2.2.1.3.1 Principal Component Analysis

Consider a data matrix **X** which contains k variables. The central idea of PCA is to reduce the dimensionality of **X** while preserving as much of the information about the variances of the k variables and the covariances or correlations between the k variables as possible. In other words, PCA retains as much of the original variation in **X** as possible. This is achieved by transforming the original, correlated variables into a new group of uncorrelated variables, the principal components, which are ordered such that the first few retain most of the variation present in all of the original variables. The main concepts of PCA will first be presented using 2 and 3-dimensional geometrical examples and then the method for determining the principal components will be derived.

First, a two dimensional example presented in Jackson will be discussed [7]. The two variables, x_1 and x_2, are plotted in Figure 2.4. A typical analysis used to describe this data would be a least squares linear regression. The two lines associated with the least squares fit are shown in Figure 2.4. However, one may want to do the prediction in either direction, that is, consider the two variables as interchangeable. In this case, an

orthogonal regression line is required. An orthogonal regression line minimizes the deviations perpendicular to the line itself. This line is shown in Figure 2.4 and is known as the first principal component of the dataset. The first principal component is in a direction such that it explains the maximum amount of variation in the original dataset with a linear combination of the original variables, as described above. The second principal component, also shown in Figure 2.4, explains the next largest amount of the variation with a linear combination subject to the condition it is orthogonal to the first principal component. Geometrically, this represents a rotation of the principal axis system. As can be seen in Figure 2.4, the data are uncorrelated with respect to the new axis system. Figure 2.5 shows a PCA for a three dimensional example. As observed in Figure 2.5, the first two principal components represent a plane in the original three-dimensional system.

For systems larger than three dimensions, a geometrical interpretation is difficult to do. However, the system can be described in a manner shown in Figure 2.6. Two terms commonly used with PCA are loadings and scores. These vectors are shown in Figure 2.6. The first loading vector, $\mathbf{p}_1^T$ defines the direction of the first principal component with respect to each of the original coordinate axes. The size of each element in the $\mathbf{p}_1^T$ vector shows the relative importance of the associated original variable to the first principal component. The first score vector, $\mathbf{t}_1$, is the linear combination of the first loading vector and the **X** matrix, that is,

$$t_1 = Xp_1 \qquad 2.3$$

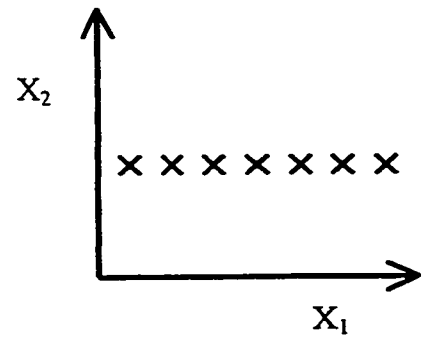
The first score vector represents the location of the individual observations on the first principal component. The a^{th} loading and score vectors are calculated and interpreted in a similar manner. There are as many loading and score vectors as there are original variables in the data matrix X.

In order to define the principal components, the numerical values in the loading vectors must be determined. The derivation of how the loadings are calculated or determined will follow Jolliffe [22]. The first step in PCA is to find the values of p_1 such that the linear combination Xp_1 has a maximum variance. This idea can be illustrated by the extreme 2-dimensional case shown in Graph 2.1. The variances for X_1 and X_2 and the covariance between X_1 and X_2 are calculated as follows:

$$\begin{aligned}
\sigma_{11}^2 &= \frac{1}{n-1}\sum_{i=1}^{n}(x_{i,1} - \bar{x}_1)^2 = \text{some finite value} \\
\sigma_{22}^2 &= \frac{1}{n-1}\sum_{i=1}^{n}(x_{i,2} - \bar{x}_2)^2 = 0 \qquad 2.4 \\
\sigma_{12}^2 &= \sigma_{21}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_{i,1} - \bar{x}_1)(x_{i,2} - \bar{x}_2) = 0
\end{aligned}$$

Clearly, from the above equations, the direction of maximum variance will be in the direction of the horizontal line in the X_1 direction and this would be the direction of the first principal component. Therefore, to determine p_1; variance(Xp_1) must be maximized. The expression variance (Xp_1) can also be written as:

$$\text{var}(Xp_1) = p_1^T \text{var}(X) p_1 = p_1^T \Sigma p_1 \qquad 2.5$$



Graph 2.1: Extreme Two-Dimensional Example

In order to find a solution, a normalization constraint must be imposed. The most common normalization constraint is to set the sum of the squares of p_1 equal to 1.0. Therefore, the problem can now be written as follows:

find p_1 such that :
$p_1^T \Sigma p_1$ is a maximum subject to the constraint $p_1^T p_1 = 1.0$
NOTE : for 2 – dimensions

$$\begin{bmatrix} p_{11} & p_{21} \end{bmatrix}^T \begin{bmatrix} p_{11} \\ p_{21} \end{bmatrix} = p_{11}p_{11} + p_{21}p_{21} = p_{11}^2 + p_{21}^2$$

The method of Lagrange Multipliers addresses problems in constrained maxima and minima. For example, if the minimum of the function $f(x, y, z) = x^2 + y^2 + z^2$ subject to the constraint $2x + 3y - z = 1$ is sought, a new variable, λ (Lagrange Multiplier) and function can be developed, as follows:

$$F(x, y, z, \lambda) = (x^2 + y^2 + z^2) + \lambda(2x + 3y - z - 1) \qquad 2.6$$

Now, the critical points of F are found. In the case of PCA, the function which needs to be maximized is:

$$\phi = p_1^T \Sigma p_1 - \lambda(p_1^T p_1 - 1.0) \qquad 2.7$$

Differentiating with respect to p_1 yields:

$$\frac{\partial \phi}{\partial p_1} = 2\Sigma p_1 - 2\lambda p_1 + 0 = 0$$
$$(\Sigma - \lambda I)p_1 = 0 \qquad 2.8$$

This is an eigenvalue problem, where λ is an eigenvalue of Σ and p_1 is the associated eigenvector. Finally, it must be determined which of the k eigenvectors is the maximizing value of p_1. From (2.8):

$$\Sigma p_1 = \lambda p_1$$
$$p_1^T \Sigma p_1 = p_1^T \lambda p_1 = \lambda p_1^T p_1 \qquad 2.9$$
$$p_1^T \Sigma p_1 = \lambda$$

The term $p_1^T \Sigma p_1$ must be a maximum, therefore λ must be a maximum and hence λ_1 must be the largest eigenvalue of Σ. Also, p_1 is the eigenvector associated with the largest eigenvalue, λ_1. In a similar fashion, it can be shown that the second principal component, p_2, is the eigenvector associated with the next largest eigenvalue of Σ. In this case, two constraints are imposed; $p_2^T p_2 = 1.0$, and, as stated above, the principal components must be uncorrelated with each other meaning the covariance between t1 and t2 must be equal to 0. The second constraint is equivalent to:

$$p_1^T p_2 = 0 \qquad 2.10$$

It is interesting to note that the eigenvalues are equal to the variances of the score vectors, as follows:

$$
\begin{aligned}
&Xp_1 = t_1 \\
&\therefore \mathrm{var}(Xp_1) = \mathrm{var}(t_1) = p_1^T \Sigma p_1 \\
&\text{from}(7): p_1^T \Sigma p_1 = \lambda_1 \\
&\therefore \mathrm{var}(t_1) = \lambda_1
\end{aligned}
\qquad 2.11
$$

If the sum of the variances of all the variables is used as a measure of the overall variability of the dataset, the eigenvalues may be used to calculate the amount of variability explained by the principal component. For example, the ratio of the first or largest eigenvalue over the sum of all the eigenvalues will be the fraction of the variability explained by the first principal component.

PCA is scale-dependent, meaning that the contribution to the total variance of a dataset for a specific variable is a function of the units of measurement of that variable [23]. In order not to have one variable dominate the analysis due to its large variance, the variables must be scaled in some meaningful way. Typically, the starting place for scaling is to mean-center and auto-scale the data. Auto-scaling means dividing each observation for each variable by the standard deviation of the variable. Hence, each variable has unit variance. It should be noted that if a variable has very small variance or standard deviation, other methods of scaling may be required. In the extreme case, if a variable has zero variance, auto-scaling will cause a "divide by zero" error in the algorithm. Kersta noted this issue in his paper in 1991 [23].

2.2.1.3.2 Fault detection Using PCA

PCA can be used to detect faults in a process in the following manner. By rearranging equation 2.3, the PCA model using all the principal components can be written as:

$$X = TP^T \qquad 2.12$$

where: X - historical dataset containing only inherent variability

If X contains many highly correlated variables, usually the first few principal components (i.e. 1, 2, ... A) will explain most of the significant variability in the system. They will be characterized by large, well separated eigenvalues and represent variability that can be attributed to natural correlations present in the data. These principal components should be retained in the model for monitoring purposes. The remaining principal components can be discarded. Therefore, the PCA model can be written as:

$$X = \sum_{i=1}^{A} t_i p_i^T + \sum_{i=A+1}^{k} t_i p_i^T \equiv \hat{X} + \text{Error}$$

$$\text{where: } \hat{X} = \sum_{i=1}^{A} t_i p_i^T \qquad 2.13$$

$$\text{Error} = \sum_{i=A+1}^{k} t_i p_i^T$$

As seen from equation 2.13, the X matrix is broken down into a prediction, $\hat{X}$, using the "A" principal components retained in the model and a residual error. Development of the PCA model involves determining two items; the number of principal components to be retained and the loadings associated with each retained principal component. There are several statistical tests that can be used to determine the number of principal components to retain. They include plotting the eigenvalues and looking for a break, evaluating the

size of the eigenvalues or cross-validation [7,24]. Briefly, the cross-validation algorithm for determining how many principal components should be retained in the model is as follows:

1. Calculate the sum of squares of the historical dataset X: SS_X
2. Divide X randomly into G groups, where each element of X is used only once.
3. Delete the first group from X and calculate the first principal component using the remaining elements
4. Calculate the predicted values of the deleted elements of X using the first principal component (as outlined below)
5. Calculate the Squared Prediction Error (SPE) for the deleted elements (as outlined below)
6. Replace the first group in X
7. Repeat steps 3-6 until all groups have been removed from X once.
8. Calculate the total sum of the SPE's: $\text{SPE(Total)} = \sum_{i=1}^{G} \text{SPE}_i$
9. Calculate the ratio: $\frac{\text{SPE(Total)}}{SS_X}$

If the ratio in Step 9 is greater than 1, that is, the total SPE is greater than the original sum of squares, the calculation should be stopped. Otherwise, the second or next principal component should be calculated. This method was used in the PCA analysis discussed in Section 5.3.1. The loadings can be determined by calculating the eigenvectors and eigenvalues of the covariance matrix. However, typically, only the first

few principal components and hence eigenvectors and eigenvalues are required. Therefore, it is desirable to calculate the principal components sequentially. One popular sequential method for calculating the principal components, which was used in this research work, is the NIPALS algorithm [25].

Once a PCA model has been developed from historical data, it can be used to monitor the process for future faults. In order to do this, two items must be monitored; the scores retained in the model and the error between the model and the new observation. The scores and error are calculated as follows:

1. Calculate the scores (t_i's) for each of the principal components, as follows:

for i =1: A

$$t_i = X_{NEW} * p_i \qquad 2.14$$

$$X_{NEW} = X_{NEW} - t_i p_i^T$$

end

2. Calculate the Squared Prediction Error (SPE) between the model and the new observation, as follows:

$$SPE = \left(X_{NEW}(\text{Original, not deflated as shown above}) - \sum_{i=1}^{A} t_i p_i^T \right)^2 \qquad 2.15$$

Referring again to Figure 2.5, which represents the case where there are 3 variables in the X matrix, it was noted that when two principal components are used in the model, they

represent a plane. The SPE represents the distance from the new observation to the plane. This is also shown in Figure 2.5.

Process faults are reflected in the scores and SPE in the following manner. If the new observation represents normal operating data, all the scores and the SPE will remain below their control limits. If the new observation represents an event that was not included in the historical dataset, the correlations between the variables will be changed and the covariance structure will be changed. This will cause the new observation to move further away from the plane than normal and will be detected by a high SPE value. If the new observation represents an event which causes larger than normal variations in the principal components used in the model but the basic correlations between the variables does not change, it will be detected in a shift in the scores.

In order to detect faults, control limits are required for both the scores and the SPE. As mentioned, the limits could be based on the historical reference distribution, that is, the historical dataset used to develop the PCA model. For example, if 95% control limits were desired, the actual limit would be set to include only 5% of the data in the tails of the histograms for the scores and SPE's. Alternatively, the control limits could be set by making some assumptions about the distributions of the scores and SPE. For example, the scores are a linear combination of many original variables. One would expect the scores to be normally distributed by virtue of the Central Limit Theorem. This arises from the fact that the scores are linear combinations of a number of variables. Therefore,

the individual scores could be monitored using standard Shewhart control charts, with the limits set at $\pm 3\sigma_{t_a}$. The scores could also be monitored using the multivariate statistic $T_A^2 = \sum_{a=1}^{A} \frac{t_a^2}{s_{t_a}^2}$ [8]. In this case, the limit for T_A^2 is calculated as a function of the F distribution, in a similar manner as to how the limit is calculated for the multivariate Hotellings T^2 limit:

$$T_{A,UCL}^2 = \frac{(N-1)(N+1)A}{N(N-A)} F_\alpha(A, N-A) \qquad 2.16$$

(The ratio of two independent variables which follow the chi-square distribution, which have been divided by their respective degrees of freedom, will be distributed according to the F distribution [18]. The chi-square distribution is a special case of the gamma distribution [18].)

The limit for the SPE can be determined by assuming a reduced Chi-Squared distribution [26].

Some limitations of the above method should be noted. First, as stated previously, if the historical dataset that is used to develop the model contains faults, then these faults will not be detected in future data. These faults will be considered normal operations and will be built into the model. Therefore, it is extremely important that the historical dataset contain only natural, inherent variability. Secondly, PCA is modeling the correlation structure present in the historical data. If the process is changed and the correlations are broken, the model will no longer be valid. If the change is done as part of normal

operations, then a new model will need to be developed based on the new steady state conditions.

2.2.1.4 Extensions to Multivariate SPC Methods

Several extensions have been suggested in the literature to the basic multivariate SPC monitoring procedure using PCA described above. Three of these extensions will be outlined below.

In some cases, the data matrix X shown in Figure 2.6 will be a three dimensional array, as shown in Figure 2.7. In this case, multi-way PCA can be used to analyse the data. The multi-way analysis is described in detail by Wold [27]. This method is very useful for analyzing data that has been collected in a sequential manner. For example, in batch processes, variables are tracked over the course of each batch. Assuming historical data are available from a significant number of batches, multi-way PCA can be used to monitor future batches, as described by Nomikos [26]. This is done by unfolding the three dimensional array in one of several possible ways and performing normal PCA on the resulting large matrix. For batch analysis, the 3-dimensional array is unfolded as shown in Figure 2.7. In this case, the resulting 2-dimensional matrix contains deviations about the mean trajectories for each variable [28]. Another application of multi-way analysis is multivariate image analysis [29].

The second extension of PCA is the inclusion of dynamic behavior or the variability caused by auto-correlation in the variables. In this method, the variables are "time

lagged" and added to the main data matrix X. This is shown in Figure 2.8. This method has the effect of removing the major dynamics from the data that result in residuals that are much better behaved or uncorrelated [28]. In order to use this model, the number of time lags and principal components to be used in the model must be determined. A simple procedure for doing this has been outlined by Ku [30].

The final extension is multi-block PCA. This extension is a major part of this research work and will be discussed in detail in Chapter 4. A basic diagram of multi-block PCA is shown in Figure 2.9. The main purpose of multi-block PCA is to help with the interpretation of the PCA model. Basically, the X matrix is divided into blocks such that the variables within the blocks are highly correlated while there is less correlation between the blocks.

2.2.2 Process Monitoring Using Artificial Neural Networks

An artificial neural network (ANN) or neural network can be defined as a massively parallel distributed processor that can store experimental knowledge and make it available for future use [31]. The knowledge is acquired via a learning process and stored in inter-neuron connection strengths known as synaptic weights. The learning process can either be supervised or unsupervised. There are two basic types of network architectures; the multi-layer perceptron (MLP) network and the radial-basis function (RBF) network. These two architectures are shown in Figures 2.10 and 2.11. The MLP network is typically trained using the back propagation algorithm while the RBF network is trained

by choosing the number, location and widths of the center in the hidden layers. Detailed training algorithms can be found in Haykin [31]. The task of fault detection using ANN is basically a pattern recognition problem. The network is trained with data that represents acceptable steady state operation and data that represents fault data. Once training is completed, the network can classify new data based on the information contained in its synaptic weights. This approach combines both fault detection and fault diagnosis into one task. There have been several papers published on the use of ANN's for FDD [32,33,34]. Also, it has been reported that the RBF network outperforms the MLP network for this type of application [35].

There are some limitations with the use of ANN's for FDD. The first limitation has to do with the training of the network. In order to follow the procedure outlined above all the different possible faults must be known and there must be data available for all of the faults. Clearly, enumerating all the possible faults in a complex process is not a realistic task. Also, ordinarily there will be an abundance of normal, steady state, operational data but very little fault data. One solution to this training problem would be to train the network on only the normal, steady state data. In this case, there would be only two outputs from the network, either the plant is operating acceptably or there is a fault. In this case, fault diagnosis would not be possible, which leads into the second limitation. It is very difficult to understand how an ANN arrives at its answer. In other words, it can not tell you why it arrived at the final output or classification. This is because the information is stored in the synaptic weights in the model. There are multiple solutions

that the network could find during the training phase. Different solutions can be found using different starting weights and sometimes the starting weights can cause the network not to converge. Each solution will produce a different set of weights and therefore, the weights cannot be used to interrogate the model. Also, ANN's are usually over-parameterized as compared to PCA models and hence the model becomes more confusing.

2.2.3 Analytical Redundancy Techniques

Analytical redundancy techniques or model-based methods for FDD involve the use of a mathematical model of the process. Here, a distinction is made between models based on first principles or analytical models and empirical models such as multivariate statistical models and ANN's. Model-based methods detect faults in the entire process, including the actuators and sensors, by measuring available input and output variables. A general scheme for model-based fault detection is shown in Figure 2.12 [37]. As seen in Figure 2.12, the models generate features that are compared to normal behaviour. If a change is detected between the normal and calculated features, a fault has been detected. There are three general types of features that can be generated with the models; estimates of unmeasured state variables $(\hat{x})$, parameter estimates $(\hat{\Theta})$ and residuals from parity equations (r). For the parity equation method, a fixed model is run parallel to the process and an output error is calculated.

One of the main limitations of the model-based techniques is that an accurate mathematical model of the process must be available. In practice, an accurate and complete mathematical description of the process is never available [36]. Typical modeling problems include unknown structures of the dynamic system, unknown parameters or parameters which are only known over a limited range of the plant's operation. The problem of a "model-reality mismatch" is known as a robustness problem. The goal of robust model-based FDD scheme is to discriminate between the fault effects and the effects of uncertainties in the model. Another limitation of the model-based methods is that the techniques are not as well known as other traditional methods such as limit checking. As a result of this unfamiliarity, the users may consider the methods a black-box and be skeptical of their results. This also leads to a fear of economic loss if the models generate an unacceptable number of false alarms due to modeling errors.

2.2.4 FDD Methods Which Do Not Use Process Models

2.2.4.1 Knowledge-Based Systems (Expert Systems)

With respect to FDD, knowledge-based systems are used primarily for fault diagnosis. In this manner, a knowledge-based system can be used to determine the cause of a fault given a set of symptoms. The main components of a knowledge-based system are shown in Figure 2.13. They are a knowledge base, inference engine and user interface. The knowledge base contains the knowledge that is stored in the form of IF-THEN rules. Knowledge can be broadly classified into two types: shallow or diagnostic knowledge and deep or behavioral knowledge. Shallow knowledge is knowledge gained by

experience. This is also referred to as heuristic knowledge or "rules of thumb". For example, a rule of thumb could be that pipes would typically leak at joints. Shallow knowledge does not consider underlying reasons or principles where as deep knowledge contains rules based on first principles. The inference engine is the mechanism by which the system interprets and applies the rules contained in the knowledge base. There are two general strategies for applying the rules: forward chaining or backward chaining, as shown in Figure 2.14. Forward chaining is data-driven and backward chaining is goal-driven. Diagnostic problems are better suited for forward chaining where the available information is used to derive as many facts as possible. The rules are selected and applied in response to the current fact base.

Conventional expert systems used for fault diagnosis typically have diagnostic knowledge in their knowledge base and use a forward chaining inference engine. Knowledge based systems have several advantages. Most importantly, they can justify their conclusions, usually by a trace through the rules that have fired. Their scope can be gradually increased over time by adding new rules to the knowledge base. Knowledge based systems can be used in situations where only heuristic solutions are available. Finally, the knowledge is easy to code, verify and revise. However, they also have several disadvantages. First, to be 100% effective, the knowledge base must contain information about every conceivable fault that the system must diagnose. The system may not be able to handle an input dataset for which it has not explicit rules and will not degrade gracefully. Secondly, usually the knowledge or rules used in the system will involve hard

limits. For example, a pressure may be considered HIGH if it is above 11 MPa. If the pressure is marginally below 11 MPa, the system will not consider the pressure HIGH, even though the pressure may still be higher than normal. This trait of expert systems is commonly referred to as brittleness. It should be noted that setpoints used in typical nuclear operations would also be considered brittle. This issue can be addressed with the use of fuzzy systems. In a fuzzy system, the hard limit of 11 MPa would be replaced with a series of values from a specific distribution, such as a triangular or normal distribution. The fuzzy system would then use rules with adjectives such as NORMAL, MODERATELY HIGH, HIGH, VERY HIGH, etc. Thirdly, if shallow knowledge is used, the conclusion justification will simply be a regurgitation of the heuristic rules that led from the observations to the conclusion. In other words, the system has no understanding of the knowledge. Finally, it is both timely and costly to develop knowledge based systems. The main cost is the knowledge acquisition. There are some methods for improving conventional expert systems. These include converting shallow knowledge to deep knowledge, using solved problems to assist in the solution of future problems (case-based reasoning) and creating new shallow rules based on common patterns between symptom and conclusions (rule induction).

2.2.4.2 Limit Checking

Classical limit checking is, in essence, the same as standard univariate SPC. The only difference is that the limits are usually based on safety requirements as opposed to statistical considerations. As with SPC, the limits must be set to strike an acceptable

balance between missed faults and false alarms. The main advantage of classical limit-based FDD methods is their simplicity and reliability [37]. They work especially well if the process operates in a steady state fashion. However, the limits are typically set to react to only relatively large changes in the variables being monitored. This prevents the system from proactively detecting small faults. Another disadvantage of classical limit checking is that in-depth fault diagnosis is generally not possible.

2.2.4.3 Frequency or Noise Analysis

Frequency analysis is an example of a signal-model-based method. These methods are used where only output signals can be measured. Frequency analysis can be used to detect vibrations that are related to rotating machinery. The machinery may have a typical frequency spectrum under normal operating conditions. A deviation from this spectrum would indicate a fault is present. The main disadvantage of this method is its lack of familiarity with respect to the operating staff of a plant, similar to process model methods and ANN's.

2.3 Applications of Various FDD Techniques in the Nuclear Industry

In order to gain insight into the current status of FDD systems in the nuclear industry, a literature search was completed. The goal was to understand which FDD methodologies are popular and currently being used in the nuclear industry.

Reifman completed an extensive survey of artificial intelligence methods for detection and identification of component faults in nuclear power plants [38]. The survey focused on systems using either expert systems, ANN's or hybrids of the two to detect and identify component faults in thermalhydraulic systems. It should be noted that Reifman also included systems that used numerical simulation programs based on first principles to detect faults in the review. He considered systems that detect faults by comparing the predicted results from a reference model to measured plant parameters a form of artificial intelligence. Systems that were designed to detect and identify sensor faults were not included. The survey reviewed 95 papers written between 1982 and 1997 and the results are summarized in Table 2.2. Several important points were noted in the paper. First, it was found that the earlier systems used expert systems and the most recent systems typically used ANN's, which follows the chronological development of the two methods. Secondly, it concluded that, as of 1997, there is not a mainstream artificial intelligence approach.

FDD System	Number of Papers Reviewed
Expert Systems	49
ANN's	33
Numerical Simulation	13
Hybrids	13

Table 2.2: Survey Results for Artificial Intelligence FDD Methods Used in NPP's

Thirdly, it concluded that the development of on-line FDD systems for nuclear power plants is still in the research stage. The bulk of the proposed systems that were reviewed were research-oriented types of applications. Practical applications were limited to incorporation into simulators used for operator training and only a couple of systems were actually installed in an operating power plant (none in the US) [38]. Finally, it is interesting to note that Riefman cited four factors that are limiting the use of model-based systems in NPP's. They were: measurement noise, model inaccuracy, drifting of the process parameters and the requirement to run faster than or in real time. These factors agree well with the disadvantages of model-based systems discussed in Section 2.2.3.

A second survey paper reviewed published applications of model-based systems from 1991-1995 [14]. This survey reviewed 110 papers and only three were related to the nuclear industry. Two dealt with detecting faults in reactor coolant pumps and one dealt with FDD in a pressurized water reactor. Again, this shows relatively few applications of model-based FDD in the nuclear industry.

Several other publications and journals were searched for nuclear applications of systems other than model-based systems, expert systems or ANN's. B. Wise et al reported on using multivariate SPC for monitoring a nuclear waste storage tank [28]. Upadhyaya and Erbay developed an on-line signal validation system using generalized consistency

checking and sequential probability ratio tests in conjunction with other methods such as ANN's [39]. The sequential probability ratio test is the basis for the CUSUM control chart discussed in Section 2.2.1.1. Golerter reported on a noise analysis application for Ontario Hydro [40].

The results of the literature search generated two important conclusions. First, as stated by Reifman, actual implementation of FDD systems in the nuclear industry is limited and the entire subject is still in the research stage. Secondly, the amount of work done with multivariate SPC in the nuclear industry appears to be minimal. Both these conclusions support the need and direction for this research, as will be discussed in the next section.

2.4 Justification For Research Into The Use of Multivariate SPC in the Nuclear Industry

From the above discussion, it is evident that research into the area of FDD methods is relevant, especially in the nuclear industry. This provides justification for the current research as it is planned to use actual plant data in the project. Also from the above review, it is evident that all the FDD methods discussed have their own advantages and disadvantages. The question remained as to which method or combination of methods would be best suited for this research. It was decided to use the new multivariate SPC methods based on PCA for the following reasons. First, when considering a NPP, one must realize the mathematical modeling of such a process is a very complex task. Assuming a mathematical model is available, it could take up to two years to properly tune the model for this type of application [41]. As discussed previously, the lack of an

accurate process model is one of the main disadvantages of the model-based method and hence this method was not considered. ANN's were not considered because, typically, the process operators lack familiarity with the techniques and hence the monitoring methods are viewed as black boxes from an operations point of view. This presents barriers for actual implementation in plants. Also, as stated previously, ANN's cannot provide justification for their conclusions, which is required in the nuclear energy environment. Finally, the development of knowledge-based systems is costly and time consuming for a large complex process such as a NPP. On the other hand, the SPC models are relatively easy to develop with the proper data and can be interrogated to help explain or justify their results. The multivariate SPC methods can be explained or developed from simple univariate control charts whose concepts are relatively straight forward. Also, there is an opportunity to learn about the process as the SPC models are being developed. Finally, there has been a minimal amount of work done in the area of applying multivariate SPC methods to a NPP.

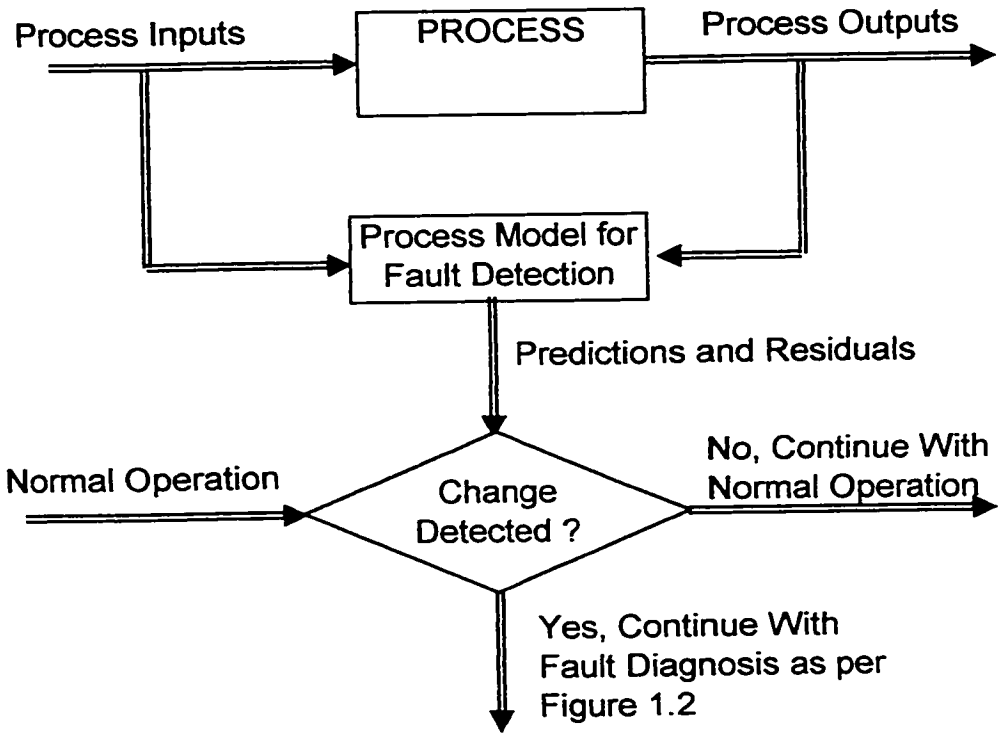


Figure 2.1: Basic Scheme for Model-Based FDD

Shewhart Control Chart

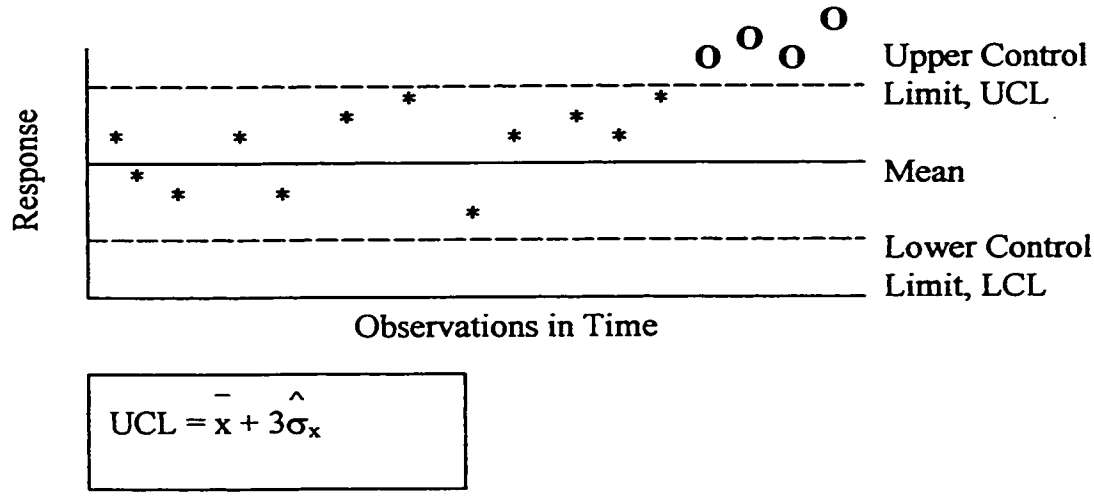


Figure 2.2: Shewhart Chart

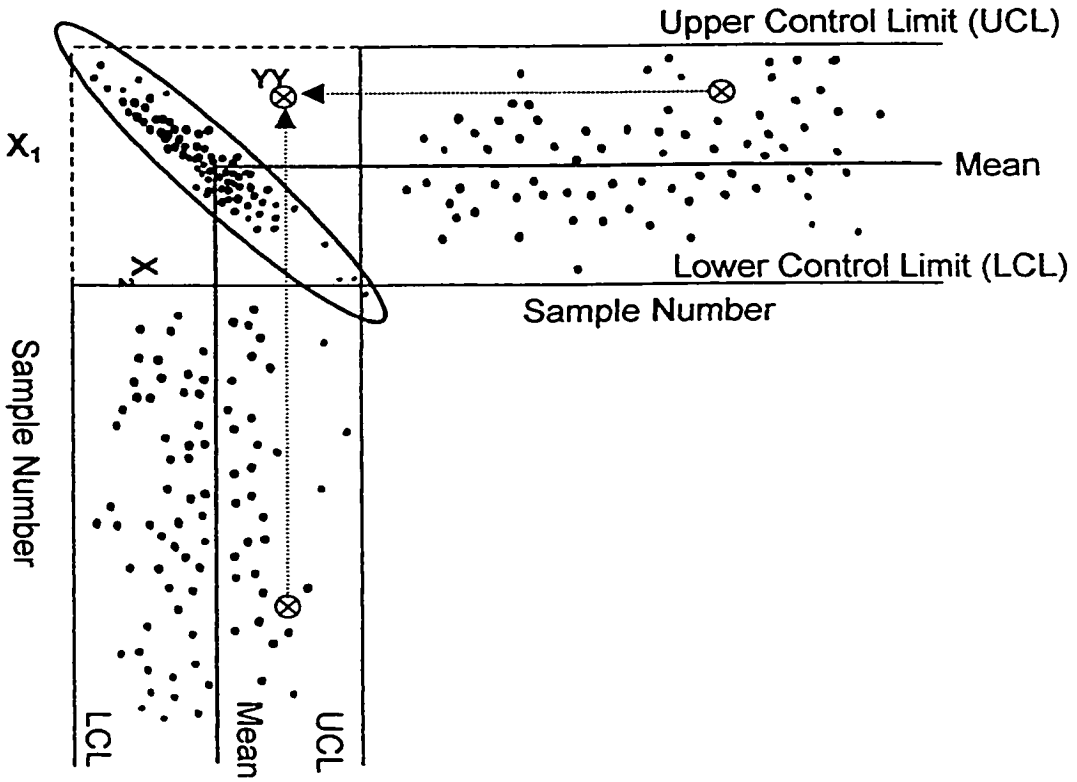


Figure 2.3: Missing Fault in Cross-Correlated Data [8]

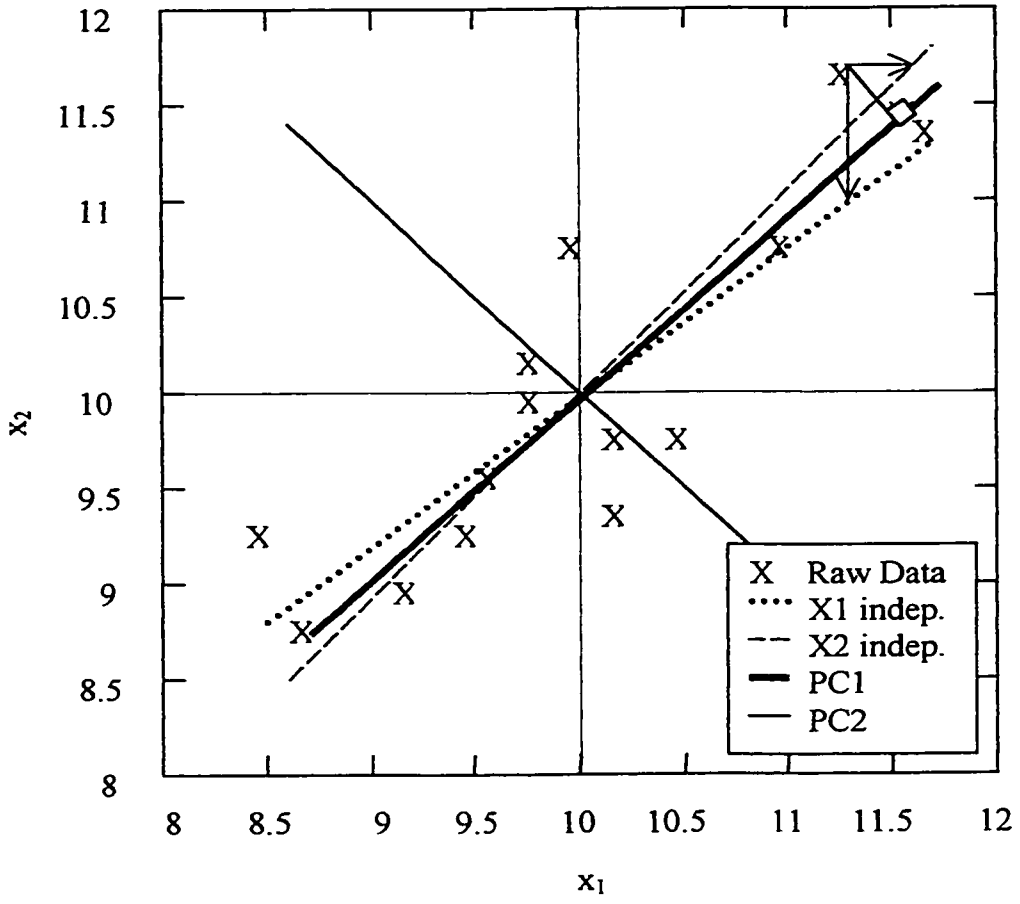


Figure 2.4: Two-D PCA Plot

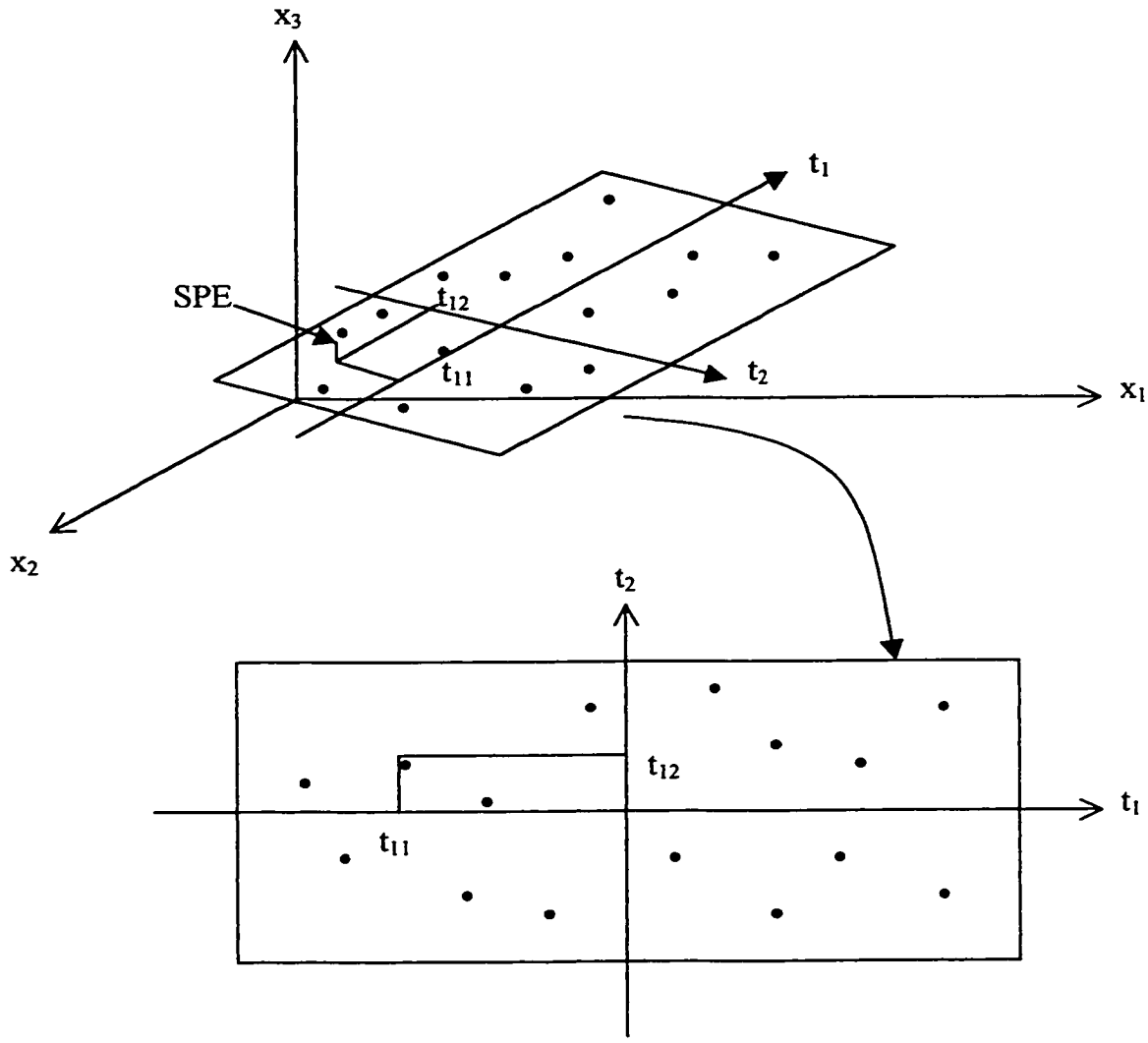


Figure 2.5: Three-D PCA Plot

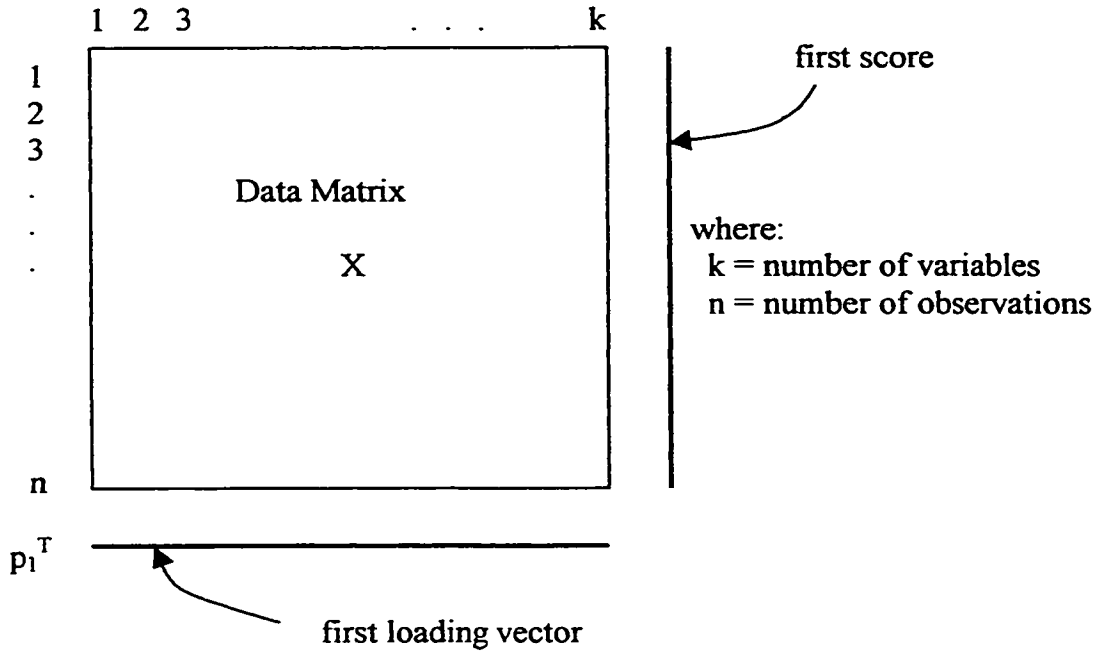


Figure 2.6: PCA Nomenclature

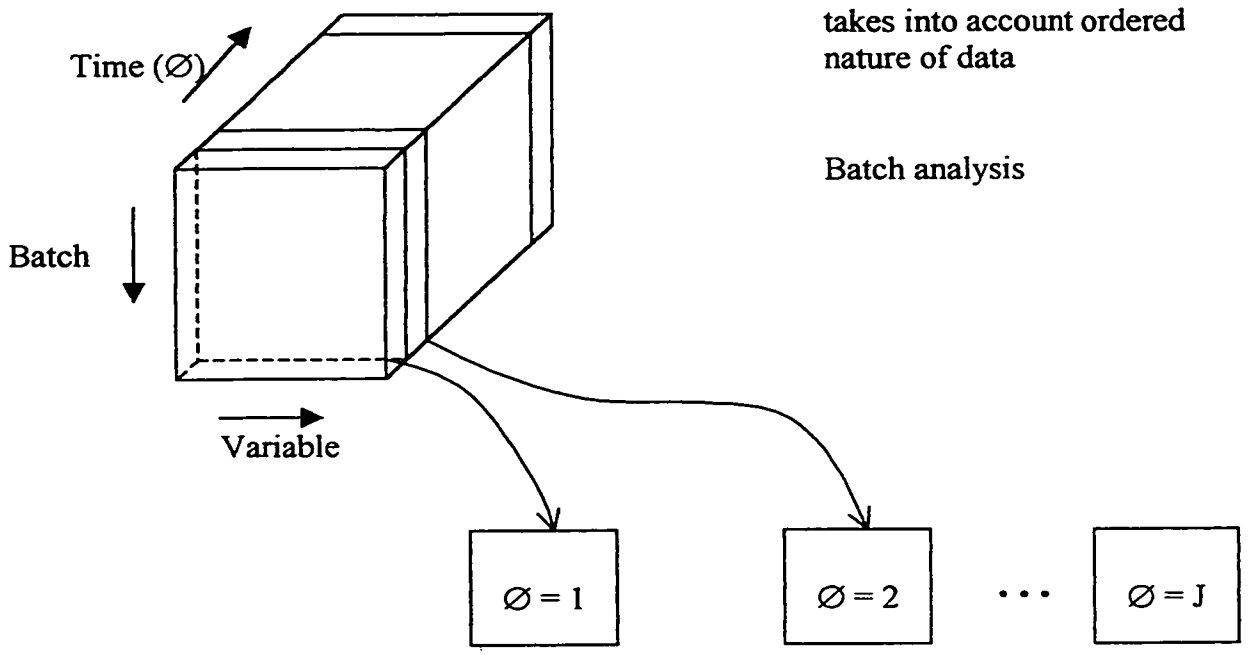


Figure 2.7: Multi-Way PCA

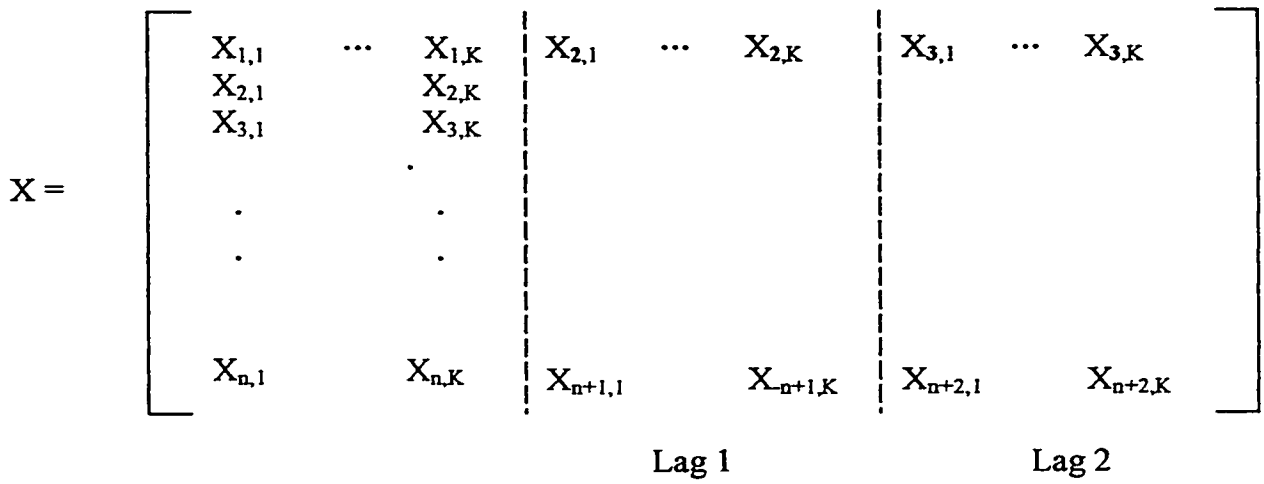


Figure 2.8: Dynamic PCA

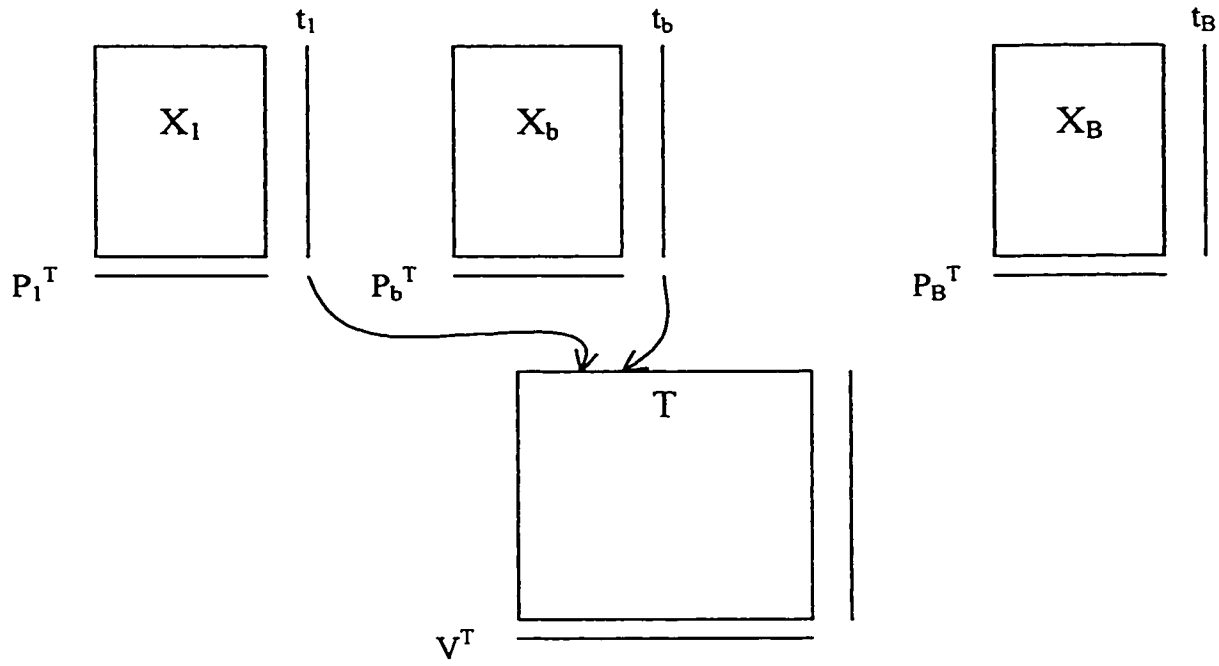


Figure 2.9: Basic Multi-block PCA

1. Multi-Layer Perceptron Network

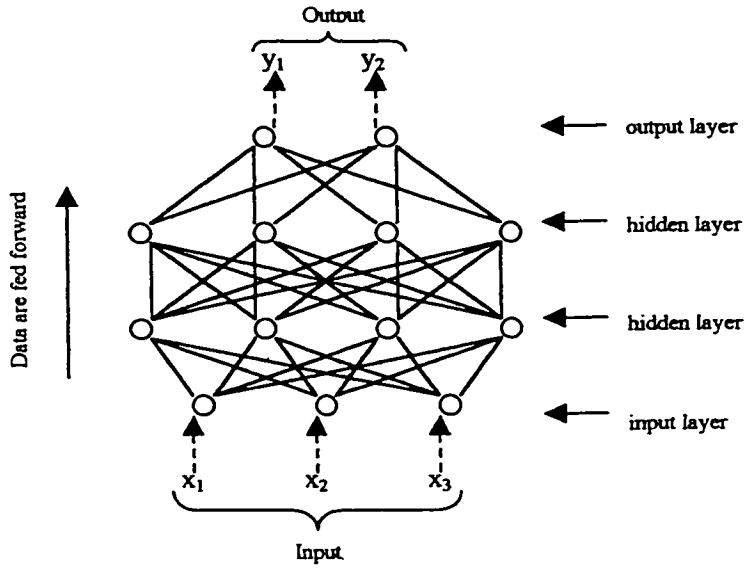
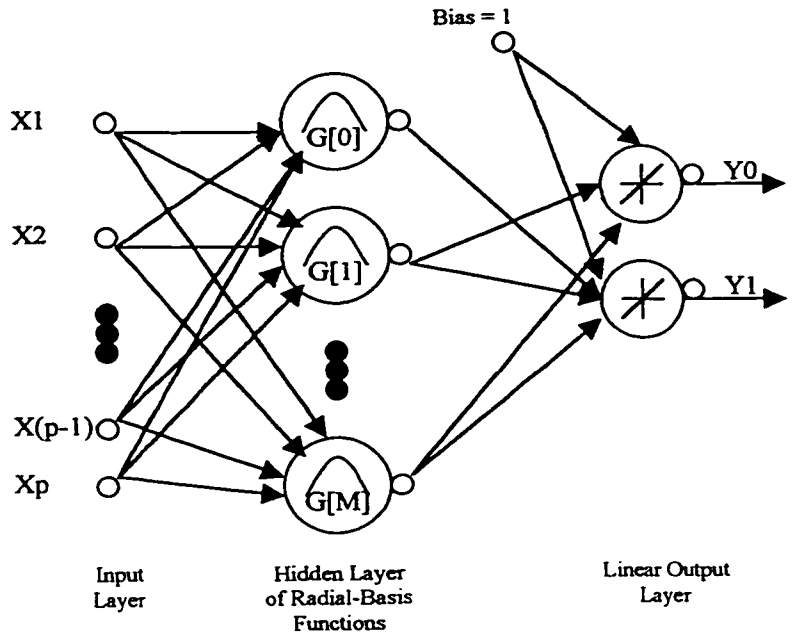


Figure 2.10: ANN Multi-Layer Perceptron Architecture

2. Radial Basis Functions Networks



Nodes are Gaussian Functions centered at t_i

$$G(\|x-t_i\|^2) = \exp\left[\frac{-\|x-t_i\|^2}{\sigma_i}\right]$$

Figure 2.11: ANN Radial-Basis Function Architecture

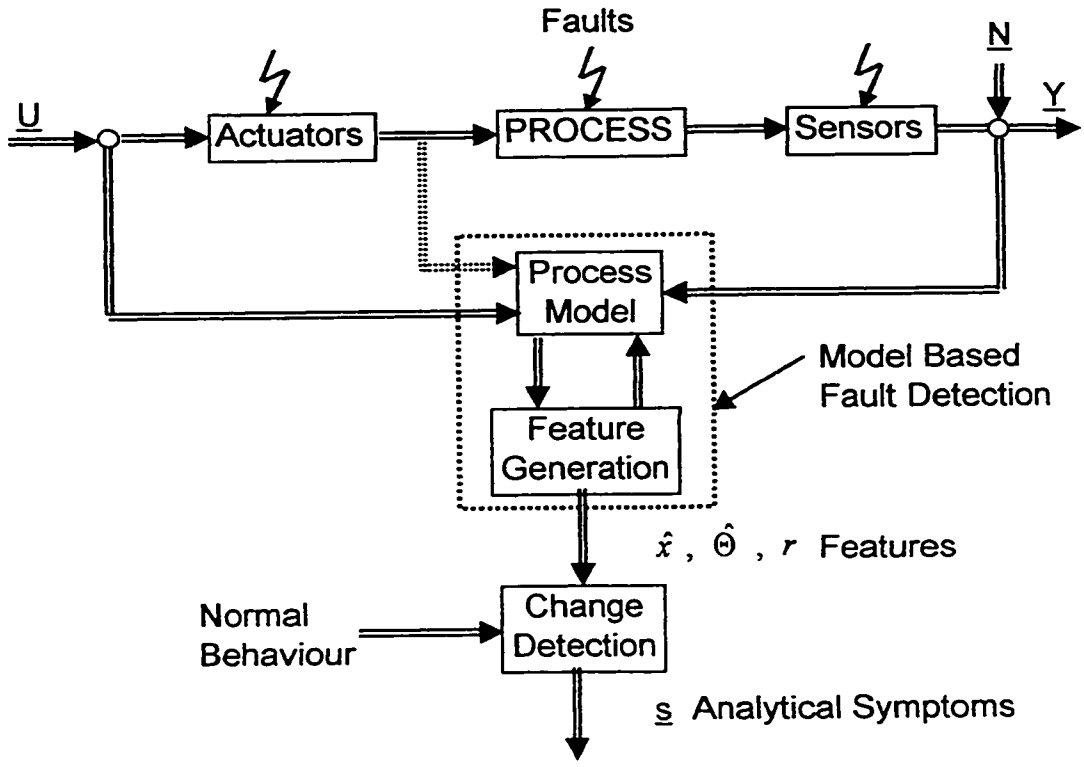


Figure 2.12: General Scheme of Model-Based Fault Detection [37]

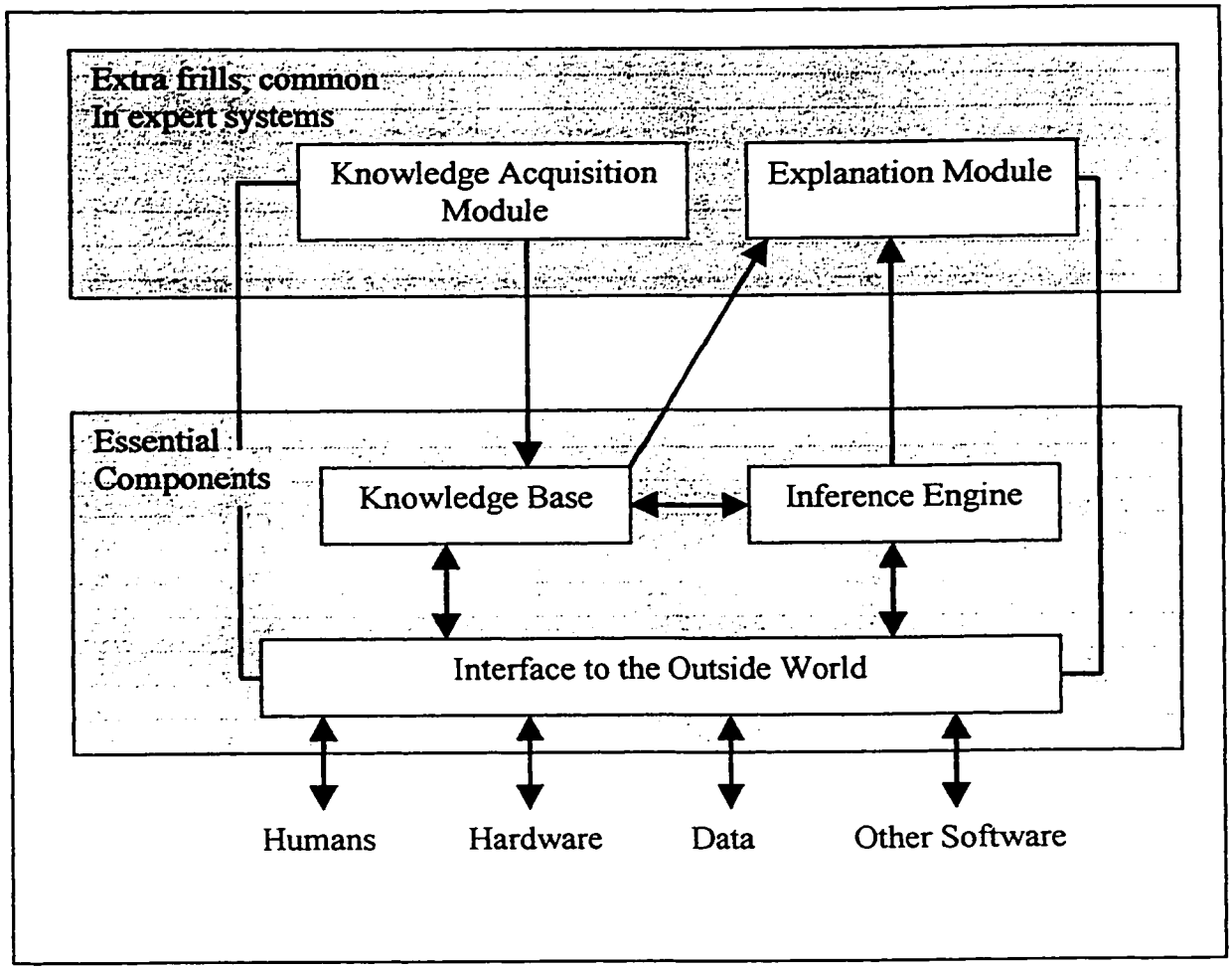
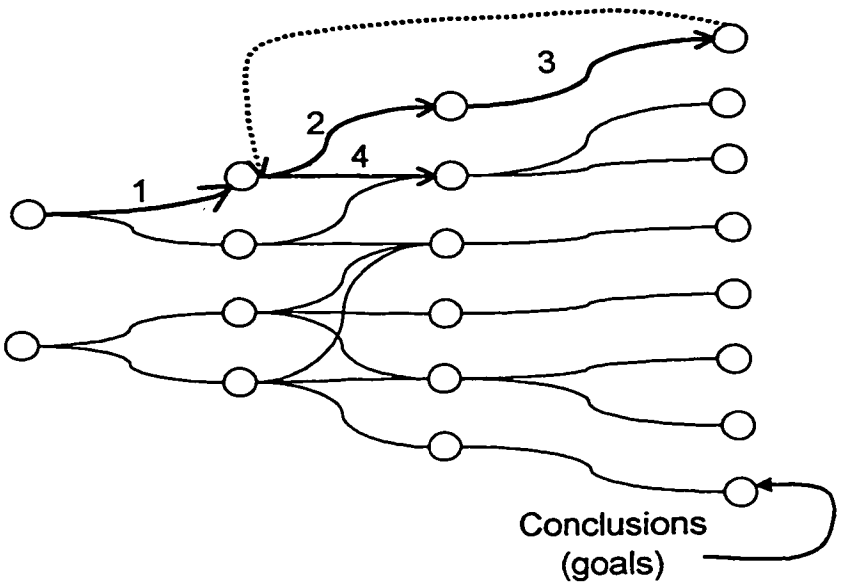
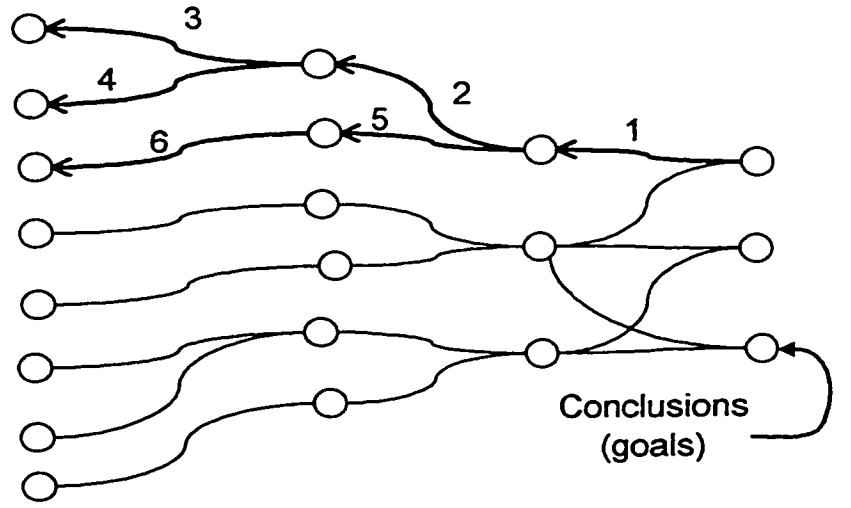


Figure 2.13: General Knowledge-Based System



Forward Chaining



Backward Chaining

Figure 2.14: Search Strategies for Inference Engines

CHAPTER 3

CANDU NUCLEAR POWER PLANT DESIGN and PROJECT DATA

3.1 Introduction

One of the main goals of this research is to investigate the application of SPC techniques to a CANDU NPP. In order to do this one must have a good understanding of the process and some process data. This chapter will be divided into four main sections. First the general design of the CANDU NPP will be reviewed. Secondly the data used for the project will be discussed. Thirdly, some current work in the area of instrumentation monitoring will be presented. Finally, the motivation for the investigation of the multi-block PCA algorithm will be presented.

3.2 General CANDU NPP Design

A NPP is basically a method of producing electricity through the generation of steam. The plant consists of two primary parts, the nuclear reactor and the balance of the plant, which contains all the remaining non-nuclear components. The overall plant can also be divided into the primary side and the secondary side. A simple diagram of a CANDU NPP is shown in Figure 3.1. The basic operation of the plant is to pass coolant through the reactor core, where it is heated. This coolant then passes through a steam generator

where the heat removed from the reactor is used to generate steam. The cycle is known as the primary side heat transport system. On the secondary side heat transport system, water flows through the steam generator where it is converted to steam. The steam is then passed through a turbine (producing electricity via the turbine generator set) and subsequently to a condenser. The condensate is then returned to the steam generators via a pump. The CANDU plants have some unique features which have earned them the reputation as one of the world's most successful reactors. Some of these unique features will be discussed in the next section.

3.2.1 CANDU NPP Design

The CANDU NPP is a pressurized heavy water reactor. Pressurized means that the primary side HTS is maintained at a high pressure (roughly 100 atm.). In this design, heavy water (D_2O) is used as the coolant on the primary side heat transport system (HTS). Another feature of the CANDU design is that it uses natural uranium as opposed to enriched uranium. This leads to reduced fuel costs as enrichment services are not required. Finally, the CANDU reactor can be refueled on-line during full power operation. This is a unique feature of the CANDU system and has contributed to the high reliability and capacity factors.

Figure 3.2 is a diagram of the primary HTS for the plant used in this study. As observed there are four reactor inlet headers, two located on each side of the reactor (RIH 1, 2, 3, 4). The purpose of two inlet headers on either side of the reactor is to break the flow up as it enters the reactor core. Some of the flow is fed directly to the outer part of the core

while some of the flow is passed through preheaters before it enters the center of the core. There are two reactor outlet headers, also located on opposite ends of the reactor (ROH 1, 2). Each ROH feeds four steam generators for a total of eight steam generators in the plant. The water exiting the steam generators then passes through two primary HTS pumps before it enters the RIH's. The presurizer shown in Figure 3.2 is used to control the pressure in the primary HTS. This is done by either adding or removing heavy water from the system as required. Finally, it should be noted that the entire reactor core is contained in a vessel called the calandria. The calandria is used to surround the core with additional heavy water, known as the moderator. The moderator is used to slow down the neutrons released in the fission process to thermal speeds. This is a requirement of all thermal reactors.

Safety is a priority in the operation of any NPP and CANDU reactors have exceptional safety records. The CANDU safety approach involves a defense in depth philosophy. The design principles state that the process systems should be independent of the safety systems and safety systems should be independent of one another [42]. The CANDU system has three basic safety systems, the reactor shutdown systems, the emergency core cooling systems and the containment systems. The reactor shutdown systems consist of two independent systems shown in Figure 3.3. The two systems are known as shutdown system 1 and shutdown system 2 (SDS1, SDS2). SDS1consists of 20-30 cadmium shutdown rods which drop by gravity with spring assistance into the reactor core. SDS2 consists of a concentrated gadolinium nitrate solution which is injected into the

moderator. Both the shutdown rods and gadolinium are very strong neutron absorbers and therefore stop the chain reaction as soon as they enter the core. These two independent systems are an example of defense in depth. The data used for this project is associated with SDS1 and SDS2 and will be discussed in detail in the next section.

3.3 Process Data Associated With SDS1 and SDS2

In order for the safety shutdown systems to work properly, they must know when a fault is present and hence, should activate. In essence, this is the task of fault detection. SDS1 and SDS2 are activated when certain important process variables rise above or drop below specified setpoints. This is the method of hard limit checking, described in Chapter 2. Table 3.1 shows the variables associated with the two safety shutdown systems used in the plant for this study. The general location of the measured variables are shown in Figure 3.2. As observed from Table 3.1, there are a total of 15 measured variables; 13 measured variables for SDS1 and 12 measured variables for SDS2. Each variable is measured redundantly either 3 or 6 times. This is an example of hardware redundancy which is used for several purposes. First, by using redundant sensors, the chance of missing a fault due to sensor malfunction is reduced. Secondly, the number of false alarms due to sensor error or mis-calibration is also reduced. Finally, redundant hardware permits on-line testing and repair of the instruments. There are three independent instrumentation channels associated with each of the two shutdown systems. Channels D, E, and F are associated with SDS1 while channels G, H, and J are associated with SDS2.

Var. #	Process Variable	Ch D	SDS1 Ch E	Ch F	Ch G	SDS2 Ch H	Ch J	Total
1	HT Pressure Header 1 (MPa)	X	X	X	X	X	X	6
2	HT Pressure Header 2 (MPa)	X	X	X	X	X	X	6
3	Pressurizer Level (m)	X	X	X	X	X	X	6
4	Boiler #2 Level (m)	X	X	X	X	X	X	6
5	Boiler #3 Level (m)	X	X	X	X	X	X	6
6	Boiler #6 Level (m)	X	X	X	X	X	X	6
7	Boiler #7 Level (m)	X	X	X	X	X	X	6
8	Boiler Feedline Pressure (MPa)	X	X	X	X	X	X	6
9	HT Flow 1 (Kg/sec)	X	X	X				3
10	HT Flow 2 (Kg/sec)	X	X	X				3
11	Moderator Temperature (°C)	X	X	X				3
12	Log N (decades)	X	X	X	X	X	X	6
13	Log N Rate (%/sec)	X	X	X	X	X	X	6
14	HDR 1-4 Differential Pressure (Mpa)				X	X	X	3
15	HDR 2-3 Differential Pressure (Mpa)				X	X	X	3
	TOTALS	**13**	**13**	**13**	**12**	**12**	**12**	**75**

Table 3.1: Measured Process Variables for SDS1 and SDS2

A brief description of each of the process variables will now be presented. The level measurements are straightforward and are shown in Figure 3.2. The header pressures measure the ROH pressures, again shown in Figure 3.2. The differential pressures, variables 14 and 15, measure the differential pressures between the specified outlet and inlet headers. This measures the pressure difference across the reactor core. The flow rates measure the flows through various channels throughout the core. Finally, the boiler feedline pressure measures the pressure at the inlet to the steam generators on the

secondary side. All of the above variables are measured using pressure-based transmitters or sensors. Log N and Log N rate are measures of power and the rate of power change respectively. They are measured using neutron ion chambers. Finally, the moderator temperature is measured using a thermocouple. The data for this research was acquired from an operational CANDU reactor as a result of the Transmitter Accuracy Monitoring System (TAMS) project initiated by Atomic Energy of Canada Limited (AECL) [43]. This system was temporarily installed in the plant used for this study and will be reviewed in detail in the next section. The data was acquired from the plant using the LabView data acquisition software package [44]. LabView saved the data in a raw 16-bit binary integer format. Data were acquired approximately every two seconds. The raw data were transferred on 230 Meg optical disks. Each disk contained approximately 400 one-hour files for a total of approximately two weeks of data. The data was decoded using a combination of C and MATLAB functions. The C function decoded the raw data, calculated averages and saved the averaged data in a new binary formatted file. This file was then read into MATLAB where the analysis was completed. The C and MATLAB functions used to decode the data are discussed in detail in Appendix A.

3.4 TAMS Project

The purpose of the TAMS was to use continuous computerized monitoring of safety system signals to verify the calibration accuracy and functionality of the pressure-based transmitters [45]. The algorithm for this system was to compare each individual sensor for a specific variable to an estimate of the true value. The true value was estimated by

calculating the mean of all the consistent and rational sensors for each new observation or time step. Some guidelines for the size of small offset errors which should be detected were suggested by AECL [46]. These guidelines were based on the quantization levels of the sensors. Here, the quantization level is considered to be the minimum interval between two adjacent digital values. AECL suggested that offsets larger than 5 quantization levels should be considered significant and should initiate further investigation by the System Responsible Engineer. Table 3.2 below summarizes the offset errors which should be detected for each variable measured with pressure-based transmitters.

Variable	Number of Sensors	Offset Error to be Detected
Header Pressures	12	+/- 50 kPa
Differential Pressures	6	+/- 13.5 kPa
Feedline Pressure	6	+/- 34 kPa
Boiler Levels	24	+/- 5 cm
Pressurizer Level	6	+/- 7 cm
Flowrates	6	+/- 0.14 kg/sec

Table 3.2: Offset Errors to be Detected

This method has been successful in detecting several calibration errors [43]. Also, it was found that both the difference mean and difference standard deviation remained roughly constant under normal steady state operation and during severe transients such as a shutdown. However, this can be both an advantage and disadvantage. It is an advantage

in that normal operations such as power changes which cause all the redundant sensors to move together do not cause false alarms. The disadvantage is that it will not detect process faults because it considers a change which effects all the redundant sensors in the same manner as normal. However, to be fair, TAMS was designed to detect calibration errors and not process faults. Also, each variable is considered individually and correlation among the different variables is not considered. Another disadvantage is that the method can not handle sensors which are not truly redundant, that is, sensors for the same variable which are not expected to behave in the same manner. This is the case for the flowrate measurements because they are measuring flows in different channels in the core which are not truly the same flow. After reviewing this work, it was hypothesized that the multi-block PCA method outlined in the previous chapter might be able to provide fault detection capabilities for both small sensor calibration errors and overall process faults. This was the motivation for the second hypothesis given in Chapter 1, as will be discussed in the next section.

3.5 Motivation for Multi-Block PCA

Ideally, a fault detection method would be sensitive to both small instrumentation errors and overall process errors. Also, as stated in Chapter 1, there is a need to distill information as it is presented to higher functional levels within the NPP. This idea is along the same lines as the information abstraction and problem solving strategy built into the OPUS performance support system developed at McMaster University [47]. This is illustrated in Figure 3.4. The development of the OPUS system identified the need for

different levels within the system with different processing capabilities, as shown in Figure 3.5. The lowest level was a technician level which performed operational tasks or data processing. The middle level was a supervisor level which performed tactical tasks or information processing. Finally, the upper level was a manager level which performed strategic tasks or knowledge processing. This multi-level framework was applied to the problem at hand as shown in Figure 3.6. As observed in Figure 3.6, the lowest level represents the technician level. In the NPP, this could represent the technician responsible for instrumentation calibration. The middle level represents the system responsible engineer who may be responsible for an entire sub-system of the plant, such as the primary HTS or the boiler pressure and level control systems. Finally, the top level represents the plant manager who is responsible for the entire plant. In order to implement this three level approach in multi-block PCA, the current algorithm had to be expanded to three levels. This work will be described in detail in the next chapter. Once this was completed, it was thought that small faults on individual sensors could be detected on the technician level but not on the two higher levels. Process faults which affected individual sub-systems would be detected by the first two levels and process faults which upset the entire plant would be detected by all three levels.

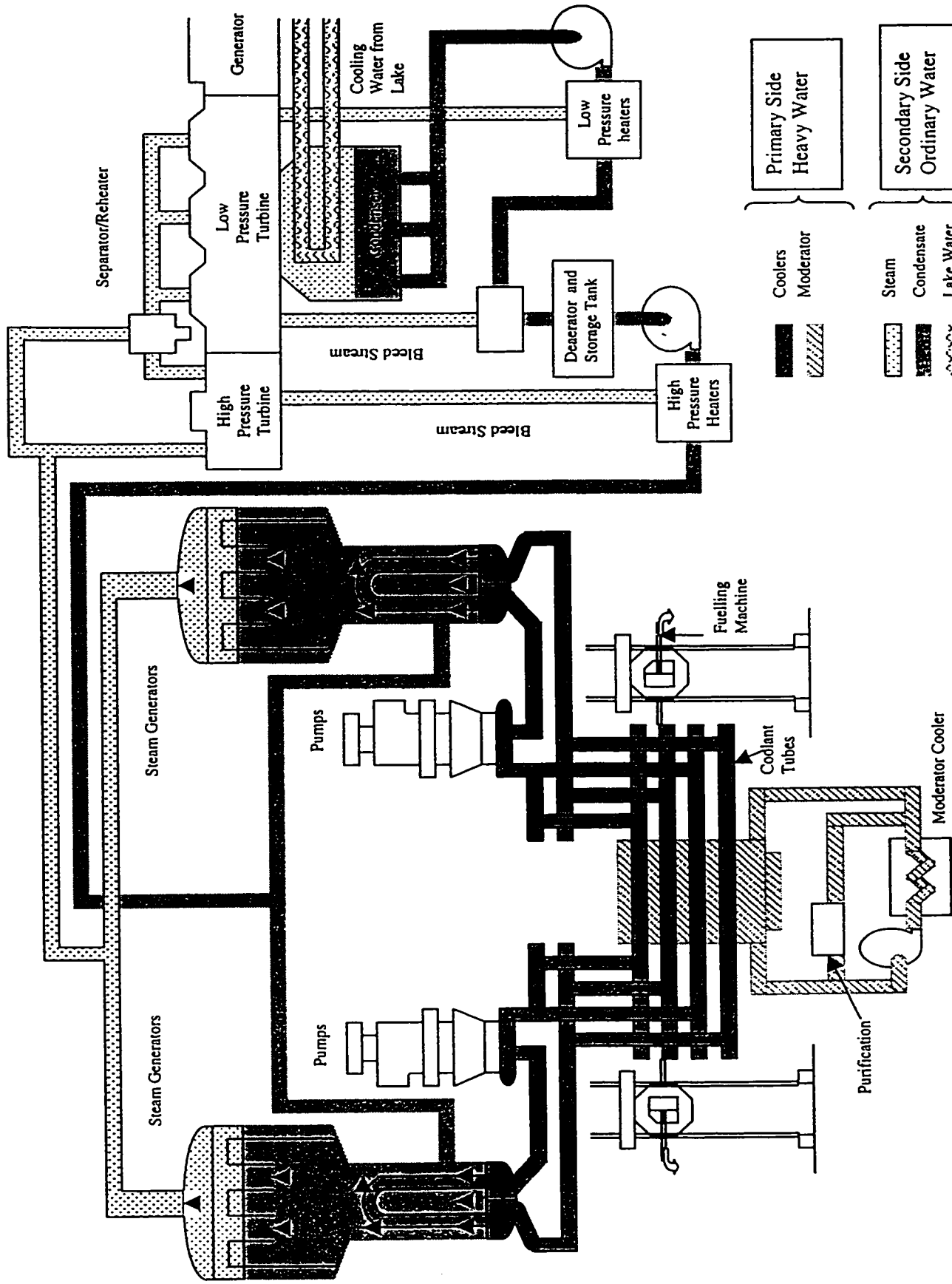


Figure 3.1: Basic CANDU NPP Design

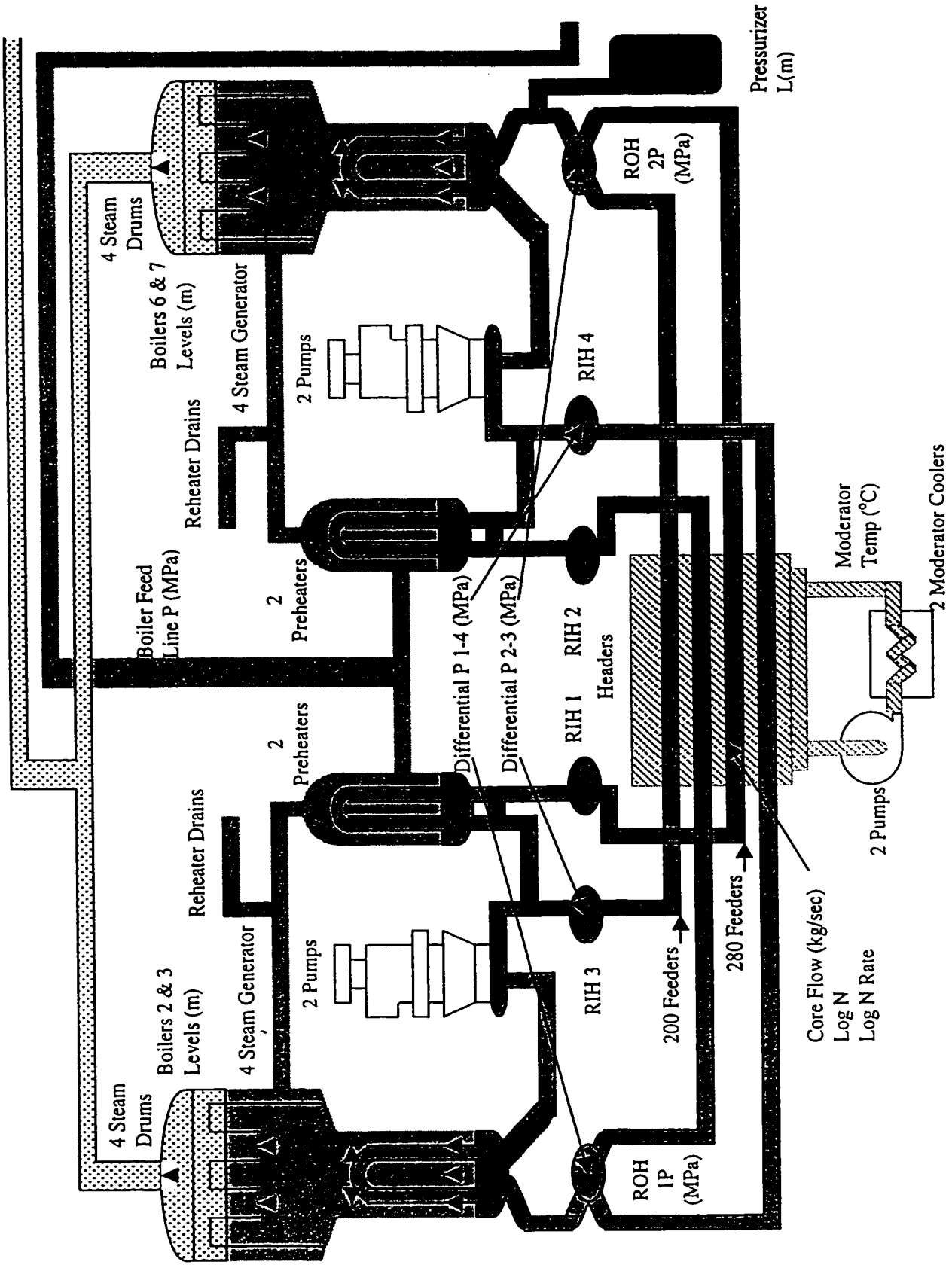


Figure 3.2 Detailed Diagram of Primary HTS [53]

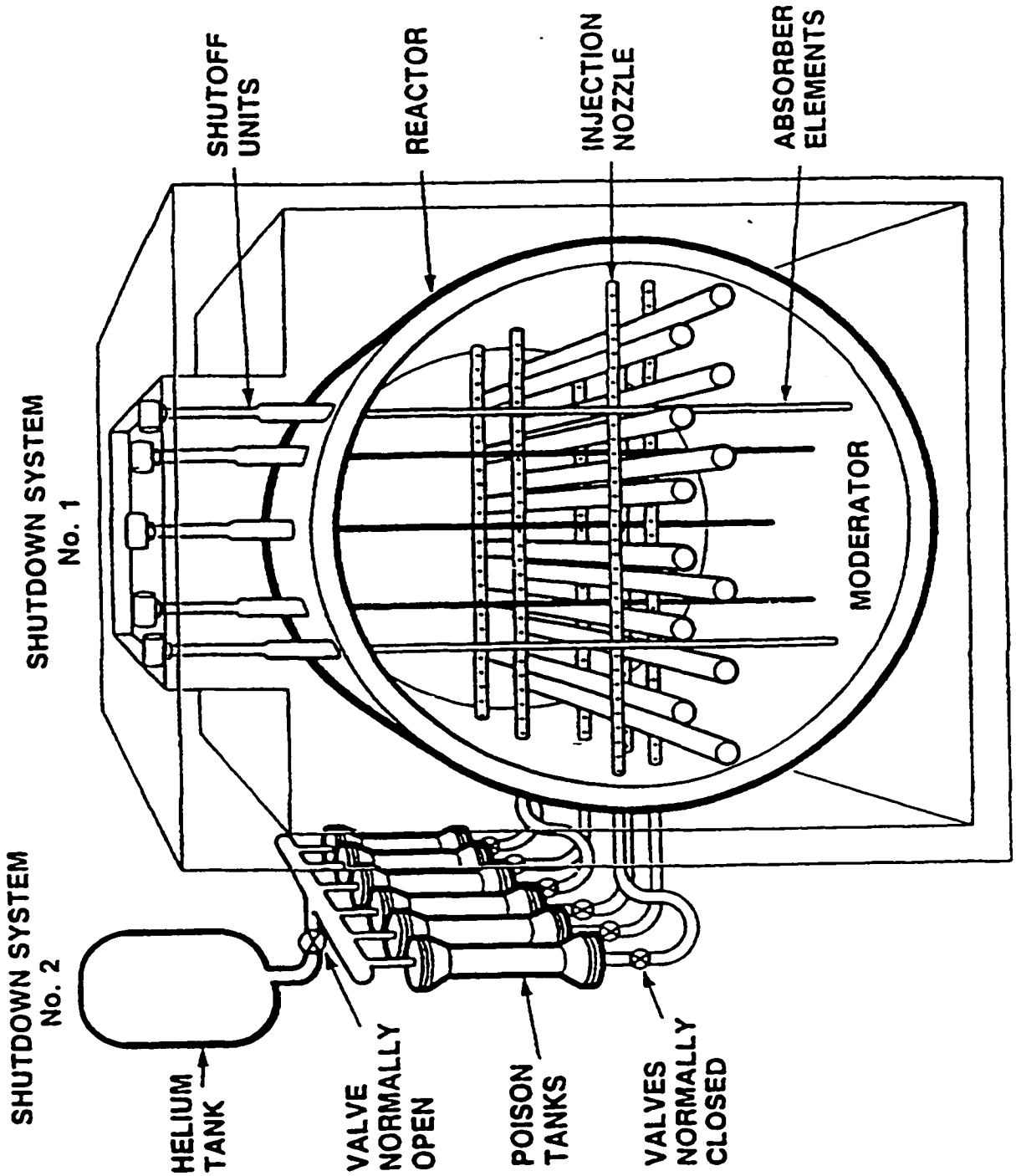


Figure 3.3: Reactor Shutdown Systems [54]

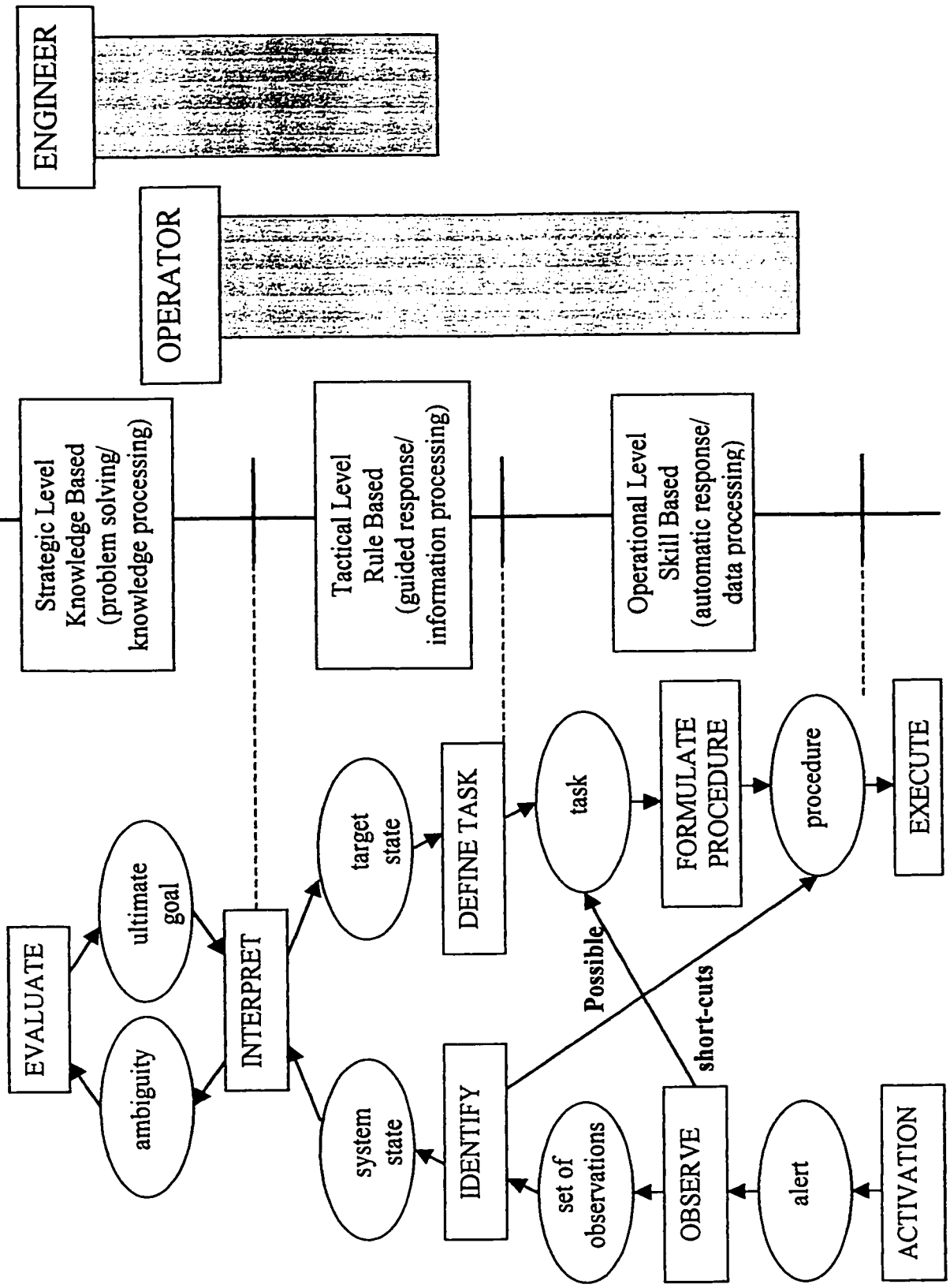


Figure 3.4: Information Abstraction and Problem Solving Strategy [47]

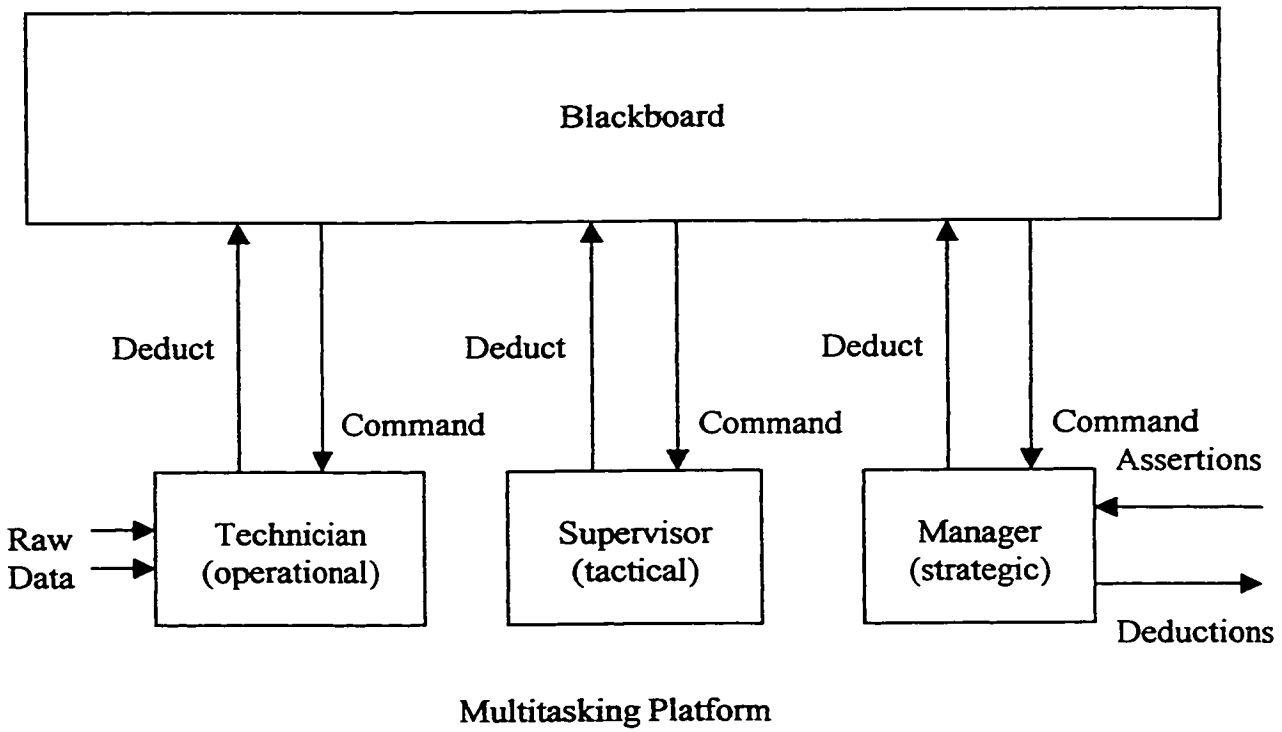


Figure 3.5: The OPUS Model [55]

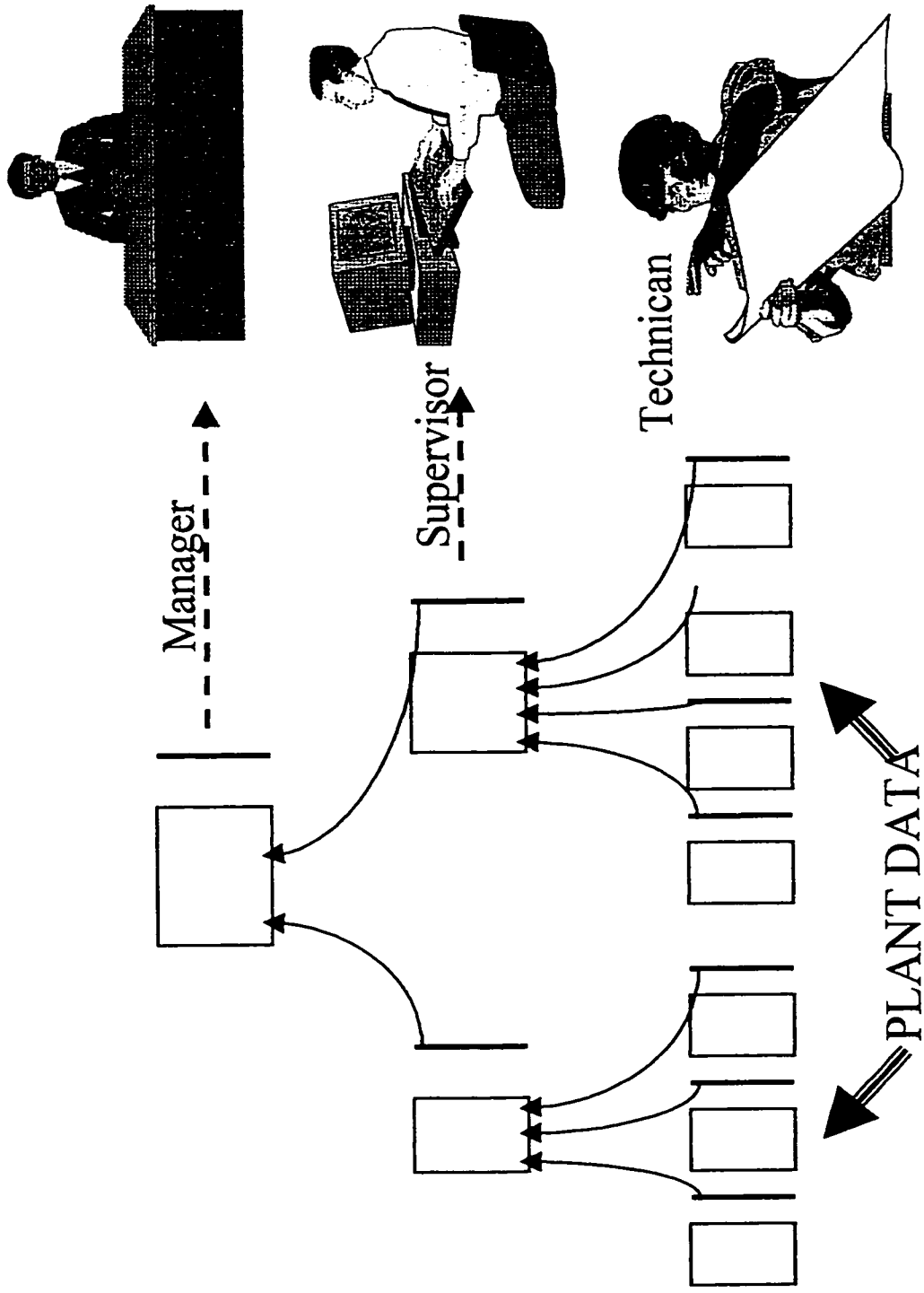


Figure 3.6: Mulit-Level Framework for NPP

CHAPTER 4

DEVELOPMENT OF THE MULTI-BLOCK, MULTI-LEVEL PCA METHODOLOGY

4.1 Introduction

In the last chapter, the need for a multi-block, multi-level algorithm has been developed. This chapter will provide a brief history of the multi-block PCA algorithm, discuss the development of the multi-level, multi-block PCA algorithm and present some analysis and characteristics of the algorithm.

4.2 History of the Multi-Block PCA Algorithm

The original multi-block PCA model (shown in Figure 2.9) and algorithm was first developed by H.Martens and S.Wold in 1984 and initially presented by S.Wold in 1987 [48]. The algorithm is presented in Appendix B along with the simple PCA algorithm. This algorithm was originally called a consensus PCA (CPCA). This code was first used in the current research in the spring of 1997. At that time, it was found that the code had convergence problems. These convergence problems were linked to the norming method used. As observed from the CPCA code in Appendix B, only the loadings in the consensus block were normed. It was found that using this method, the algorithm would

not converge. The algorithm did converge if the consensus scores were normed. This result appeared to be confirmed by a Wold et.al in 1996, when they published an updated version of CPCA which they called HPCA (Hierarchical PCA) [49]. This algorithm is also presented in Appendix B and shows that the consensus score is now normed. However, using the H-PCA algorithm, it was found that there was more than one possible solution and the solution depended on the initial starting guess for the consensus score. This result was found by running several test cases for data from a low-density polyethylene (LDPE) process simulation. The results of this work are summarized in Appendix C. It was found that in order for the algorithm to converge, both the consensus scores and the individual block scores had to be normed. These results were confirmed by Westerhuis et. al. who reviewed the algorithm for both norming the loadings, which they called CPCA and norming the scores, which they called HPCA [50]. Both the CPCA and HPCA codes presented by Westerhuis are also included in Appendix B. It should be noted that one change has been made to the HPCA algorithm, which will be discussed below.

The prediction method for the original codes was also investigated. Wold presented a prediction method in both the 1987 and 1996 papers. These prediction methods are also presented in Appendix B. However, the prediction methods did not work due to the errors in the algorithms. It was found that in order to do predictions, the block norming had to be carried through the calculations, in much the same fashion that the scaling of the original data has to be applied to the new data. Therefore, the predictions are calculated

as shown in Figure 4.1. Two points should be noted in this calculation. First, dividing the block scores by the appropriate norms follows from an inspection of the codes given in Appendix B, which are reproduced in Figures 4.2 and 4.3 for convenience. Consider HPCA, shown in Figure 4.2. It is noted that on the final iteration, t_b is calculated and normed. This value of t_b is used in T to calculate w_t and hence t_t. Therefore, in order to get the correct value of t_t to be used in deflation for the prediction, t_b must be divided by the norm of the block score calculated when building the model. A similar analysis follows for CPCA. From Figure 4.3, it is noted that on the final iteration, t_b is calculated using the normed value of p_b. This value of t_b is used in T to calculate w_t and hence t_t. Again, in order to get the correct value of t_t to be used in the deflation, t_b must be divided by the norm of the block loading calculated when building the model. The second point to note is that when calculating the consensus scores in the prediction, they do not need to be divided by the appropriate norm from the model. Again, this will be explained for both HPCA and CPCA. First consider HPCA. In the algorithm, one step was added to the code presented by Westerhuis. This is the step where w_t=w_t/norm_t_t. This step was presented in the code given by Wold in 1996. By doing this, the information about the norm for the consensus score in the model is included in w_t and hence t_t does not need to be divided by its norm in the prediction. This extra step could be left out of the algorithm for the model, but then it would need to be included in the prediction code. For CPCA, t_t is calculated using w_t which has been normed in the previous step. Therefore, t_t does not need to be divided by the consensus loading norm in the prediction code.

4.3 Multi-Level PCA

In this section, the extension of the two level multi-block PCA to three levels will be discussed. The extension to multi-levels was discussed briefly by Wold but no algorithms were given [49]. The general schematic of the proposed model was shown in Figure 3.8. A detailed diagram of the purposed model is shown in Figure 4.4. As observed in Figure 4.4, the three level algorithm includes eight steps. For each dimension, the level three consensus score is guessed at. This starting score is then regressed on each column of each block in level 1 to give the loadings for each block. Each block score in level 1 is then calculated as a linear combination of the rows in the block and the new loadings. These calculations are shown as steps 1 and 2 in Figure 4.4. The matrices for the second level are then created by grouping the appropriate block scores from first level. This is step 3 in Figure 4.4. Then steps one and two described above are repeated for each second level block, as shown by steps 4 and 5. Finally, each block score from the second level is collected into the third level matrix, step 6, and steps 1 and 2 are repeated again on the third level block, steps 7 and 8. This process is continued until the third level score converges. This is a relatively straight forward extension of the two level multi-block PCA and could easily be expanded to several levels. A detailed description of the actual code used to calculate the model and a copy of the prediction code are given in Appendix D. The prediction code follows the same pattern as the prediction for two levels discussed in the last section. It should be noted from the above discussion that there are various options for both norming and deflation in the model calculation. All these options were built into the code used to calculate the model and will be discussed in the next section.

4.4 Norming and Deflation Analysis

There are basically two methods for norming in the model algorithm, either the scores or the loadings must be normed. There are also three different options which can be used in the deflation step. Each block in the first level can be deflated by either the third level score, the score from the second level block associated with it or by its own score. These options are shown below.

$$\text{Option 1: } X_b\{level1\} = X_b\{level1\} - t_c\{level3\} * p_b^T\{level1\}$$

$$\text{Option 2: } X_b\{level1\} = X_b\{level1\} - t_b\{level2\} * p_b^T\{level1\} \qquad 4.1$$

$$\text{Option 3: } X_b\{level1\} = X_b\{level1\} - t_b\{level1\} * p_b^T\{level1\}$$

Therefore, there were six different norming/deflation combinations which could be used in the algorithm. Westerhuis investigated some of these options for a two level model [50]. To summarize, he found that when the underlying correlations or latent direction were spread out among all the blocks, norming either the scores (HPCA) or the loadings (CPCA) gave similar results. However, if there was a strong direction in only one block, CPCA found it while HPCA disregarded it in favour of a weaker direction present in multi blocks. Different deflation methods were investigated for multi-block partial least squares (MPLS) only. It was found that deflating by the block scores performed worse than deflating by the super score because of removal of information that is not used for the prediction. However, the goals of MPLS and CPCA or HPCA are not entirely the same. In general, PCA has one objective, to model the covariance structure in one matrix **X**. PLS, on the other hand, has two objectives. First, it tries to model the covariance in **X**

for process monitoring. At the same time, it is trying to model the relationship between **X** and **Y** so **Y** can be predicted from **X**. The conclusion given above that super score deflation should be used is based on the performance for predicting **Y** from **X**. This same conclusion may or may not be valid for multi-block PCA methods and so it was investigated.

In order to analyse different norming/deflation methods, a blocking strategy for the variables was required. Considering the objectives of the different levels described in Chapter 3, the blocking strategy shown in Figure 4.5 was developed. From Figure 4.5, it is observed that groups of redundant sensors are blocked together in level 1. The second level contains blocks of variables which should be highly correlated with one another. For example, the header pressures, differential pressures, and boiler levels are grouped in different blocks. While these groupings do not represent complete sub-systems in the plant, they could represent different areas of the plant for which different engineers may be responsible. Finally, the third level block contains all the scores from the second level blocks and should be representative of the entire plant. This blocking strategy resulted in 15 blocks in the first level, 8 blocks in the second level and one block in the third level. Finally, some actual data was required for the norming/deflation analysis. Three two week periods of data were obtained from AECL. They covered two weeks in Nov./95, March/96 and Sept./96. According to AECL, all data represented normal steady state operations. After some initial analysis, which will be described in the next chapter, this was found not to be the case. The November data was found to contain two strong

directions or underlying correlations in the data. The first affected the pressurizer level and Log N while the second affected both header pressures. It was decided to use this dataset in the analysis because of the known trends, similar to the analysis by Westerhuis. Ten days of the November data were decoded using 15 minute averages and obvious outliers were removed. This resulted in 843 observations. The method for screening the data for outliers will also be discussed in the next chapter.

The first step in this analysis was to build PCA, HPCA and CPCA models with two dimensions or principal components. The three different deflation techniques were used for both the HPCA and CPCA models. The loadings, scores and percent of the sum of the squares explained were then compared for the seven resulting models. Table 4.1 summarizes the sum of squares explained for each model.

Model	**% Sum of Squares Explained (SSexp)**		
PCA	**48.69%**		
		Deflation Method	
	Level 1	Level 2	Level3
CPCA (norming loadings)	54.36%	52.91%	48.69%
HPCA (norming scores)	80.26%	68.59%	44.54%

Table 4.1: Percent Sum of Squares Explained for Seven Models

Figure 4.6 shows the loadings for a PCA model and the level one loadings for the six multi-level, multi-block PCA models for the first two dimensions. Figure 4.7 shows the score for the PCA model and the level 3 scores for the six multi-level, multi-block PCA models. Several interesting insights can be gained from Table 4.1 and Figures 4.6 – 4.7.

First, in Table 4.1, it is noted that the sum of squares explained decreases as the deflation moves from level one to level three for both CPCA and HCPA. This occurs because more information for specific blocks is being used if they are deflated by the lower level scores. To explore this further, the sum of squares, loadings and scores for the header pressures and pressurizer levels were examined in detail for the CPCA case. These variables were chosen because, as seen in Figure 4.7, the third level score for the first PC looks like the general trend in the raw data for the pressurizer levels which is quite different from the general trend in the raw data for the header pressures. Plots of the raw data are given in Appendix E. On the other hand, the third level score for the second PC looks like the general trend in the header pressures. Table 4.2 gives the sum of squares explained for all of the blocks by the first PC when deflating with either the first or third level scores.

Block	**% Sum Squares Explained**	
	Deflation : Level 1	**Deflation : Level 3**
Header Pressure 1	22.4	5.4
Header Pressure 2	21.6	4.7
Pressurizer Level	70.4	82.6
Log N	53.2	51.9
Boiler 2 Level	52.1	35.0
Boiler 3 Level	52.0	35.0
Boiler 6 Level	44.2	24.0
Boiler 7 Level	58.5	46.4
Feed Line Pressure	26.3	11.5
Flow Rate 1	7.9	8.2
Flow Rate 2	16.1	20.2
Differential Pressure 1-4	44.4	49.5
Differential Pressure 2-3	35.3	39.1
Log N Rate	4.2	2.2
Moderator Temperature	8.1	7.2

Table 4.2: Percent Sum of Squares Explained for CPCA Model, 1st PC

Consider first the header pressures. Table 4.2 shows that the sum of squares explained increases from 5% for deflating with level 3 to 22% for deflating with level 1. From Equation 4.1, it is noted that the same loading factor is used regardless of the deflation method. For Header Pressure 1, Channel D, the loading was -0.0421. Hence the different sum of squares explained must be due solely to the different scores. Figure 4.8 shows the level three score along with the level one block score for header pressure 1. The score for the header pressure block looks like the general trend in all the header pressures. Therefore, one would expect deflation by this score to explain a larger portion of the sum of squares. However, because the score is being scaled down by –0.0421, only 22 % of the sum of squares is explained. The actual deflation sequence for header pressure 1, channel D for the deflating using the first level is shown in Figure 4.9. As observed, the product t1*p(HP1 block) has the same general trend but is very small when compared to the original data. It is interesting to note that when deflating header pressure channel D by the third level score, 5% of the sum of squares is still explained. This is despite the vastly different trend in the original data and the score. To understand how this happens, the deflation sequence for header pressure 1, channel D using the third level score for deflation for the first PC was plotted in Figure 4.10. As seen, the deflation causes a spike in the deflated data that is approximately the same magnitude as the original data, only in the different direction. Therefore, the sum of squares remains approximately constant for the initial part of the data where the third level score looks like the pressurizer level trend.

It appears that the third level score is able to explain a portion of the sum of squares in the later part of the data.

Next, the pressurizer levels were analyzed. The loading for pressurizer level, channel D is –0.2031. This loading is an order of magnitude larger than the header pressure loading. Hence, it is expected that more of the sun of squares will be explained. The score for the pressurizer block in level 1 is shown in Figure 4.8, along with the score for the header pressure block. The deflation sequences using both the block score and the third level score are shown in Figures 4.11 and 4.12 respectively. As seen in these figures and Table 4.2, deflating using the third level score actually explains more sum of squares than deflation using the block score. This is due to the fact that the spike in the overall score (tc in Figure 4.8) is larger than the block score (t1 Plev in Figure 4.8). For the second PC, the results for the header pressures and pressurizer level are generally reversed. Now, the header pressures have the larger weights and the pressurizer levels have the smaller weights. Hence the incremental sum of squares explained is larger for the header pressures.

The next interesting observation from Table 4.1 is that the HPCA algorithm explains more of the sum of squares for deflation using either levels one or two but less for level three. The same general trend was also found for the September data, as shown in Figure 4.13. In order to gain insight into this observation, a detailed comparison of the percent

sum of squares explained for Header Pressure 1 and Pressurizer Level using CPCA and HPCA was completed. This is shown in Table 4.3 along with the following calculations:

- ratio of the HPCA loading to the CPCA loading for Header Pressure 1, Channel D and pressurizer level, Channel D (Load ratio)
- average of the ratios of the HPCA score to the CPCA score for Header Pressure 1, Channel D and pressurizer level, Channel D (Score ratio)
- the product of the Load ratio and Score ratio (*).

Var.	Deflation with Level 1					Deflation with Level 3				
	SSexp CPCA	SSexp HCPA	Load ratio	Score ratio	*	SSexp CPCA	SSexp HCPA	Load ratio	Score ratio	*
HP1	22.4	53.1	185	.0142	2.6	5.5	10.3	185	0.007	1.2
P Lev	70.4	94.8	134	.0141	1.9	82.6	73.4	134	0.007	0.9

Table 4.3: Detailed Comparison of the Sum of Squares (SSexp) for CPCA and HCPA

Deflation using level 1 will be discussed first. Recall once again from Equation 4.1 that the deflation is a function of the block score and the loading. This is the rationale for comparing the loadings and scores for the two algorithms. If the deflation for the two algorithms were to be the same, the product of the ratios, as shown in Table 4.3, should be equal to 1.0. However, this is not the case. For the header pressure, the product is 2.6, indicating that the product of the loading and score for HPCA is 2.6 times as large as for CPCA. For the pressurizer level, the product is 1.9. Therefore, more of the sum of squares is explained when HPCA is used. This is shown in Figure 4.14 which shows the deflation sequence for the header pressure for HPCA and CPCA. However, the same

trend is not seen with deflation using level 3. In this case, the product of the ratios are closer to 1.0. Specifically, for the header pressure, the product is 1.2 and the difference in SSexp for HCPA and CPCA is much less than for deflation with level 1, where the product is 2.6. For the pressurizer level, the product is 0.9. In this case, CPCA actually explains more sum of squares and again the difference between CPCA and HPCA is much less than for deflation with level 1. The same general trend is expected for the other 13 blocks in level 1. In summary, the results for the two algorithms are somewhat as expected for deflation using level 3, where the two algorithms explain approximately the same amount of sum of squares. However, it appears that the HPCA explains more sum of squares when deflation with level one scores is used. This occurs because the product of the loading and score is consistently larger for HPCA, when compared to CPCA. However, it is not clear as to whether this observation is a function of the data used in this project or if it is indeed a characteristic of the algorithms. This is an area for future work, which will be discussed in Chapter 6.

The next interesting observation noted was that using CPCA and deflating with level 3 scores produced the same results as standard PCA. This was found by comparing the loadings and scores for the two models discussed above, shown in Figures 4.6 and 4.7 respectively, and finding that they were identical. It was also confirmed by tracking the overall sum of squares explained for each model for 2 PC's for both the November and September data and finding that, again, the values were the same. This was expected based on the proof given by Westerhuis that the score in PCA equals the consensus score

of CPCA for a two block 2 level model [50]. As discussed earlier, Weterhuis also found that HPCA will disregard a very strong direction in a single block to favor a weaker direction present in many blocks. This trait was also seen in this analysis. The strong direction in the header pressures was missed by HPCA in the second dimension, as shown in Figure 4.7. However, it was detected by CPCA in the second dimension. This result was similar to the standard PCA results.

Finally, it was observed that the sum of squares explained in each of the blocks in level one was simply the average of the sum of squares explained for the individual sensors in the respective block. Similarly, the sum of squares explained in the second level blocks equaled the average of the sum of squares explained for each associated block in level one. This is illustrated for the header pressures, pressurizer levels and Log N in Table 4.4.

Also, it is noted that the sum of squares explained for the third block is equal to the average sum of squares explained for all individual sensors. These observations are easily verified by examining the equations used to calculate the SSexp for each level. The equations used to calculate the SSexp for each level along with proofs for the above observations are presented in detail in Appendix F.

Variable	% Sum of Squares Explained				
	Individual Sensor	Average of Individual Sensors	Block 1 Values	Average of Block 1 Values	Block 2 Values
HP1, Ch. D	19.1%				
Ch. E	26.3%				
Ch. F	23.4%	22.4%	22.4%		
Ch. G	20.8%				
Ch. H	22.7%				
Ch. J	22.3%			22.0%	22.0%
HP2, Ch. D	21.8%				
Ch. E	21.9%				
Ch. F	21.7%	21.6%	21.6%		
Ch. G	21.3%				
Ch. H	21.3%				
Ch. J	21.2%				
PLev, Ch. D	71.6%				
Ch. E	72.7%				
Ch. F	72.2%	70.4%	70.4%		
Ch. G	72.7%				
Ch. H	60.2%				
Ch. J	72.8%			61.8%	61.8%
LogN, Ch. D	54.2%				
Ch. E	54.7%				
Ch. F	51.6%	53.2%	53.2%		
Ch. G	53.9%				
Ch. H	54.2%				
Ch. J	50.4%				

Table 4.4: Percent Sum of Squares Explained for CPCA Model, Deflation With Level 1

There were two main goals for the above analysis. The first goal was to provide insight into the different norming and deflating techniques. The second goal was to decide on the norming/deflating strategy to be used for this project. With regards to the norming strategy, it was decided to use CPCA. The rational for this decision was based on two key observations outlined above. The first reason for using CPCA is the characteristic of

HPCA to miss strong directions in individual blocks. Based on the general trends in the November data, it was thought that strong directions in individual blocks could be an important attribute of the project data in general. Therefore, using HPCA could result in deficiencies in future analysis. Secondly, the CPCA algorithm behavior is more similar to the standard PCA algorithm. This is shown in Figure 4.7 and by the fact that PCA and CPCA with deflation using the highest level score are equivalent. With regards to the deflation strategy, it was decided to use the level 1 scores because this resulted in more sum of squares being explained in each dimension. Also, for this project, there is no need to predict a Y (i.e. an output matrix) matrix from the process data matrix. Hence, there is not as great a concern about losing information by deflating with the level 1 block scores as discussed by Weterhuis.

Given a new observation : $x\,[(1 \times k)]$

1. Calculate the block scores as : $t_b = x_b p_b\ [(1 \times b)(b \times 1) = (1 \times 1)]$
2. Norm the block score :

CPCA : $$t_b = \frac{t_b}{\text{norm\_}p_b(\text{from model})}$$

HCPA : $$t_b = \frac{t_b}{\text{norm\_}t_b(\text{from model})}$$

3. Collect the block scores into matrix T
4. Calculate the consensus score :

CPCA or HPCA : $t_T = Tw_T\ [(1 \times b)(b \times 1) = (1 \times 1)]$

Figure 4.1: Prediction Code for CPCA and HPCA

Transform, center, and scale

For each dimension:

choose start t_T

loop

normalize t_T : $t_T = \frac{t_T}{\text{norm\_}t_T}$

calculate block loadings : $p_b = \frac{X_b^T \bullet t_T}{t_T^T \bullet t_T}$

Break if convergence of t_T

Calculate block scores : $t_b = X_b \bullet p_b$

Normalize t_b : $t_b = \frac{t_b}{\text{norm\_}t_b}$

Combine all block scores in T : $T = [t_1 \ldots t_b]$

Calculate consensus loadings : $w_T = \frac{T^T \cdot t_t}{t_T^T \cdot t_T}$

NOTE : FROM WOLD(1997) : $w_T = \frac{w_T}{\text{norm\_}t_T}$

Calculate consensus score : $t_T = T \cdot w_T$

end

Deflation : $X_b = X_b - t_T \cdot w_T$

end

4.2: Model Code for HPCA (adapted from [50])

Transform, center, and scale
For each dimension:
  choose start t_T
  loop
    calculate block loadings: $p_b = \frac{X_b^T \cdot t_T}{t_T^T \cdot t_T}$
    Break if convergence of t_T
    Normalize block loadings: $p_b = \frac{p_b}{norm\_p_b}$
    Calculate block scores: $t_b = X_b \cdot p_b$
    Combine all block scores in T: $T = [t_1 ... t_b]$
    Calculate consensus loadings: $w_T = \frac{T^T \cdot t_t}{t_T^T \cdot t_T}$
    Normalize consensus loadings: $w_T = \frac{w_T}{norm\_w_T}$
    Calculate consensus score: $t_T = T \cdot w_T$
  end
  Deflation: $X_b = X_b - t_T \cdot w_T$
end

Figure 4.3: Model Code for CPCA (adapted from [50])

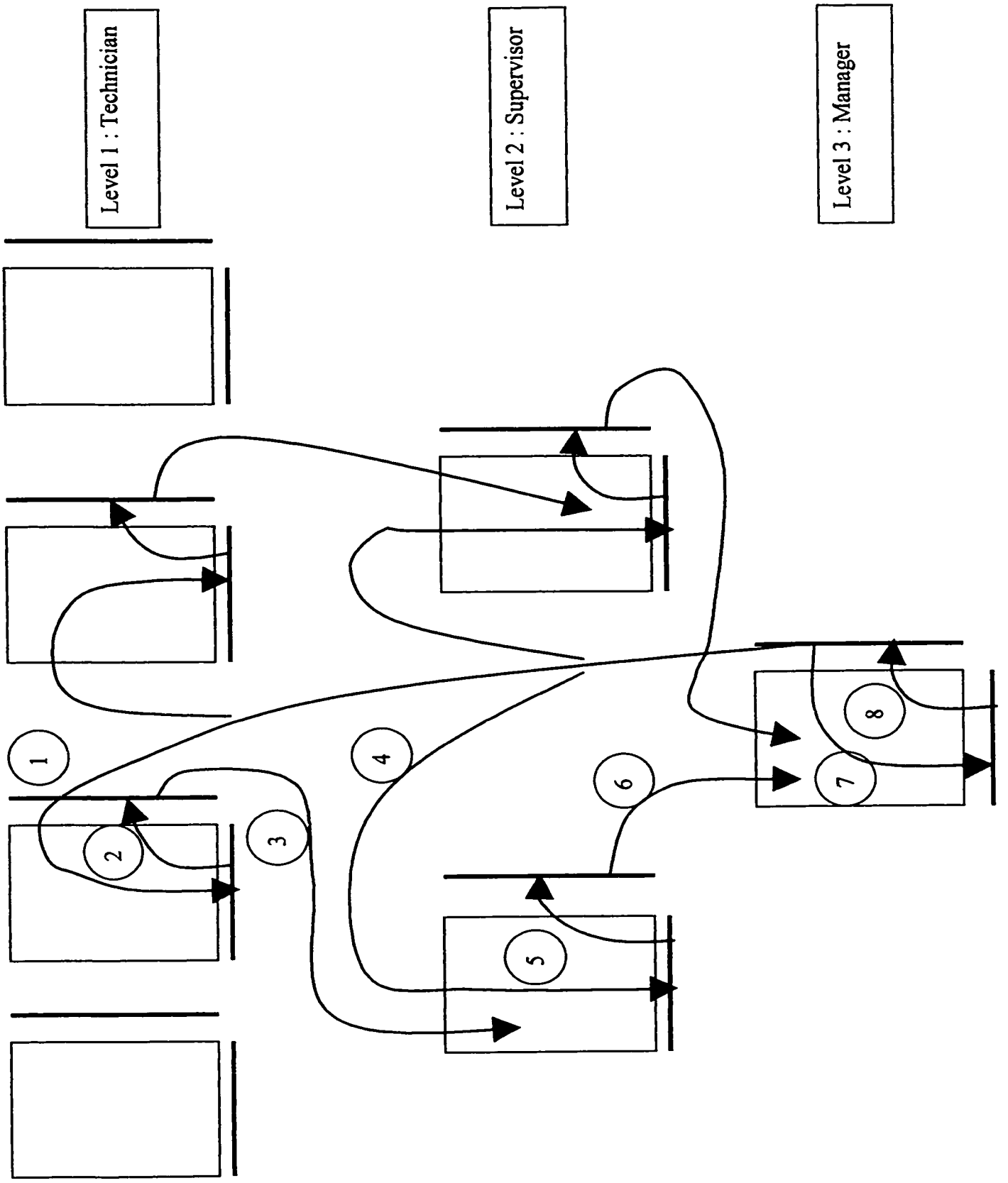


Figure 4.4 : Detailed Muti-Block, Multi-Level Model

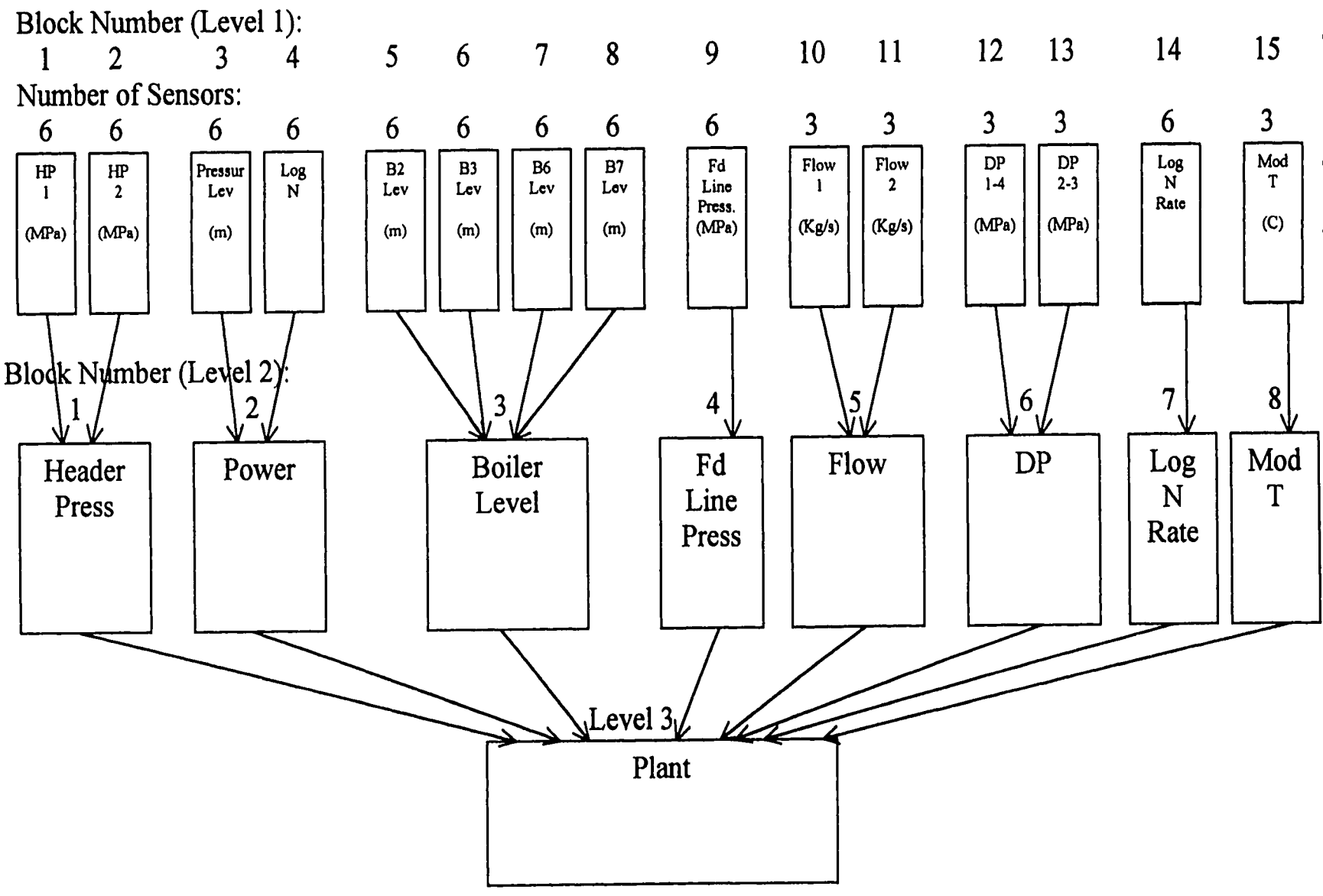


Figure 4.5: Blocking Strategy for NPP Data

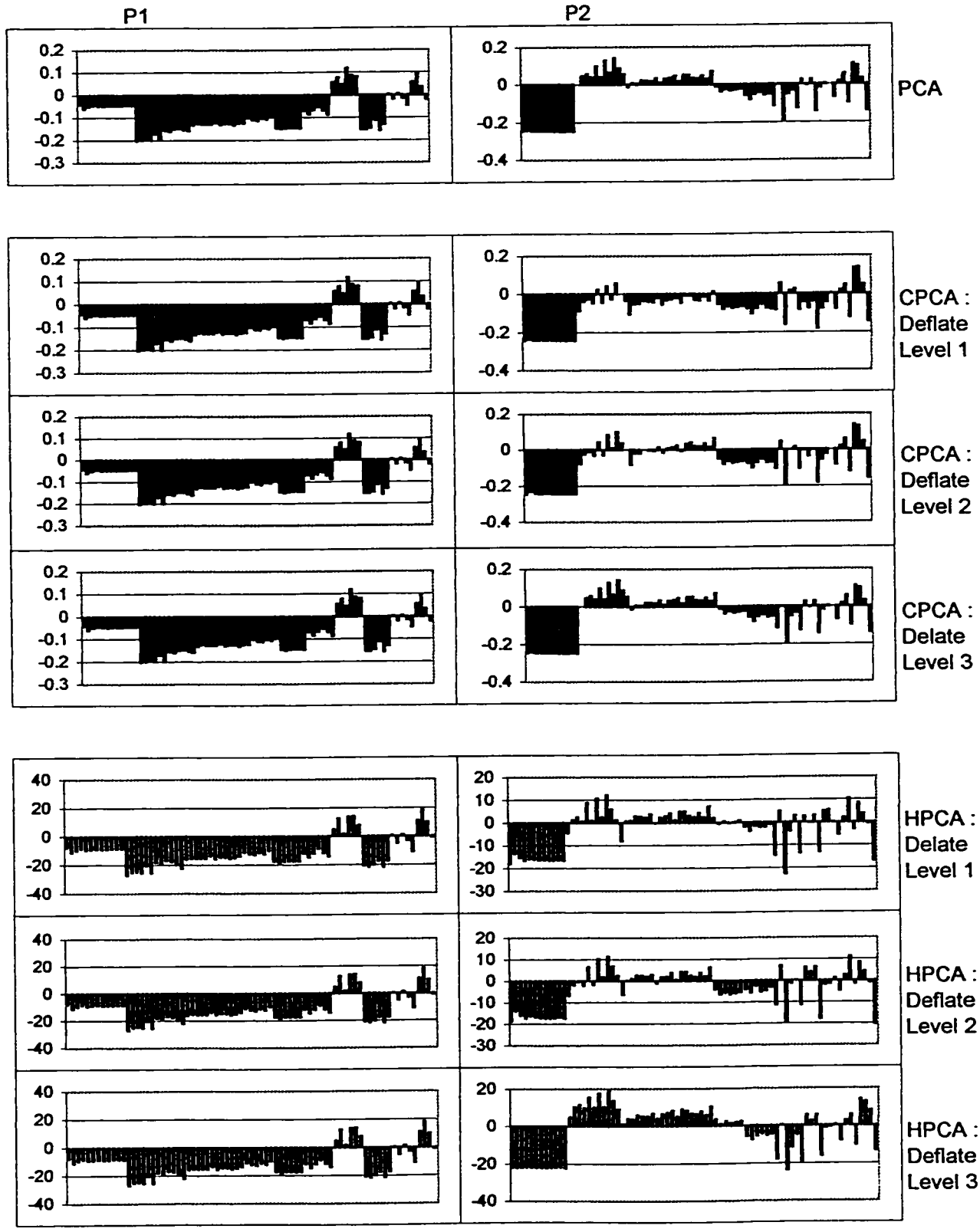


Figure 4.6 : Loadings for PCA, CPCA and HPCA Models

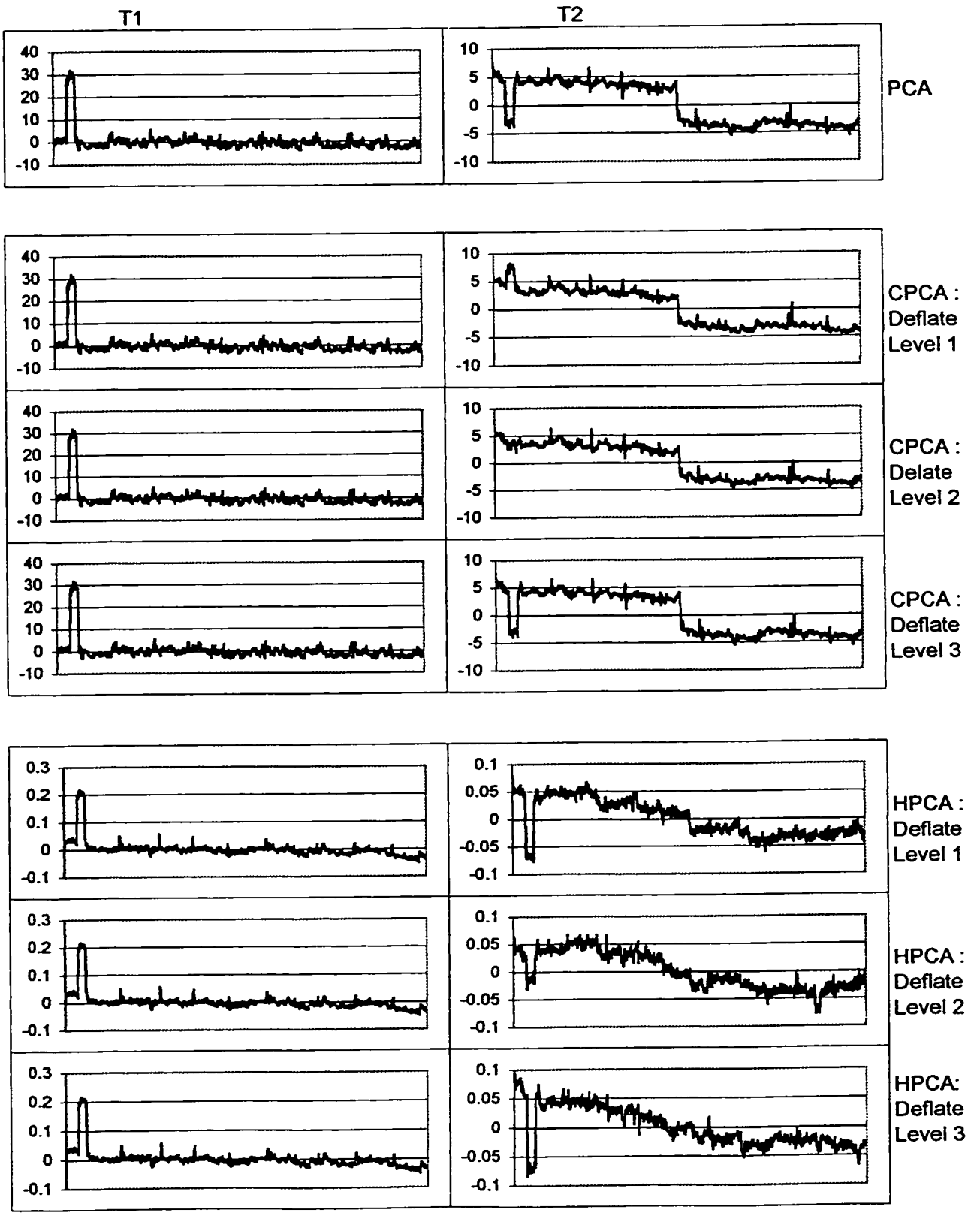


Figure 4.7: Scores for PCA, CPCA and HPCA Models

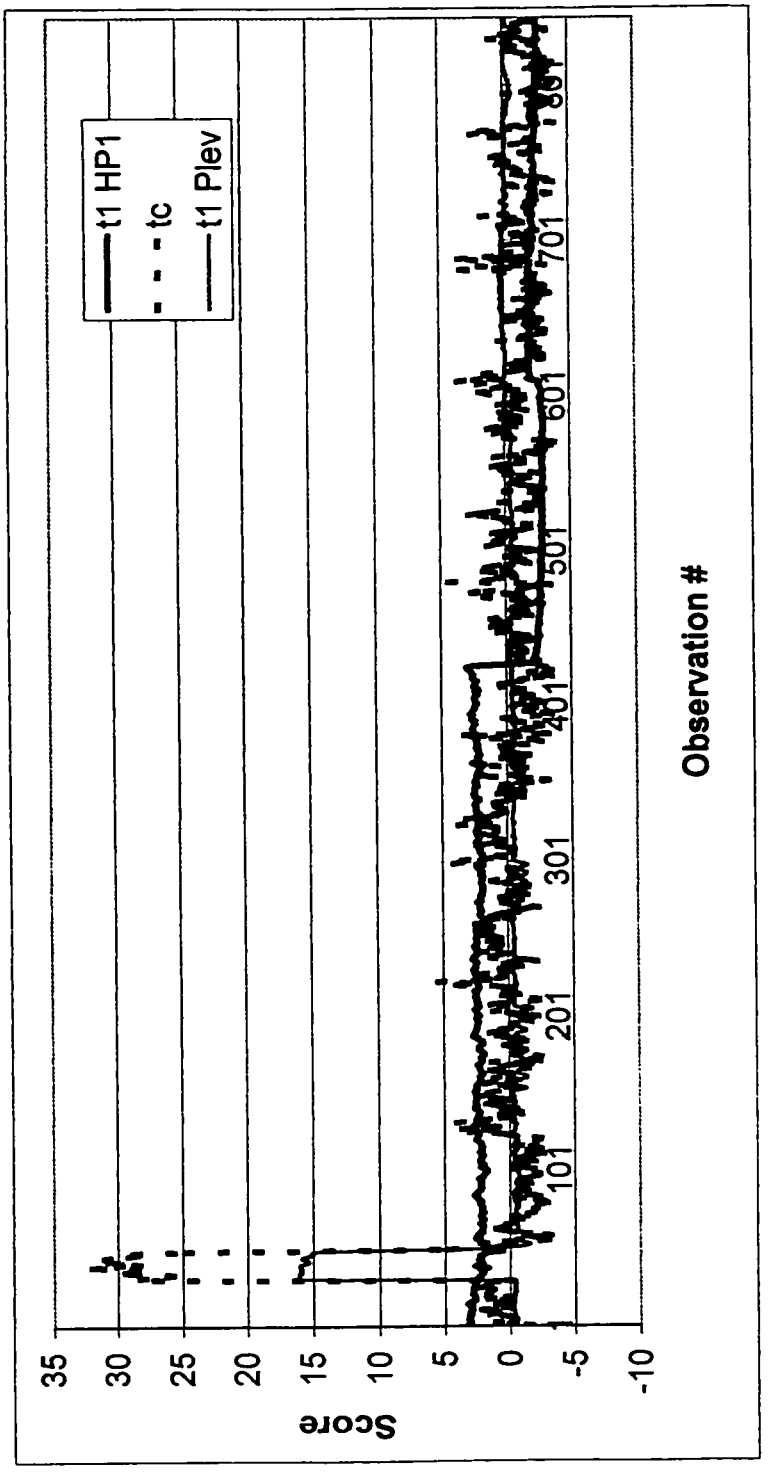


Figure 4.8 : CPCA 1st PC, Level 3 Score and Level 1 Score for Header Pressure 1 Block and Pressurizer Level Block

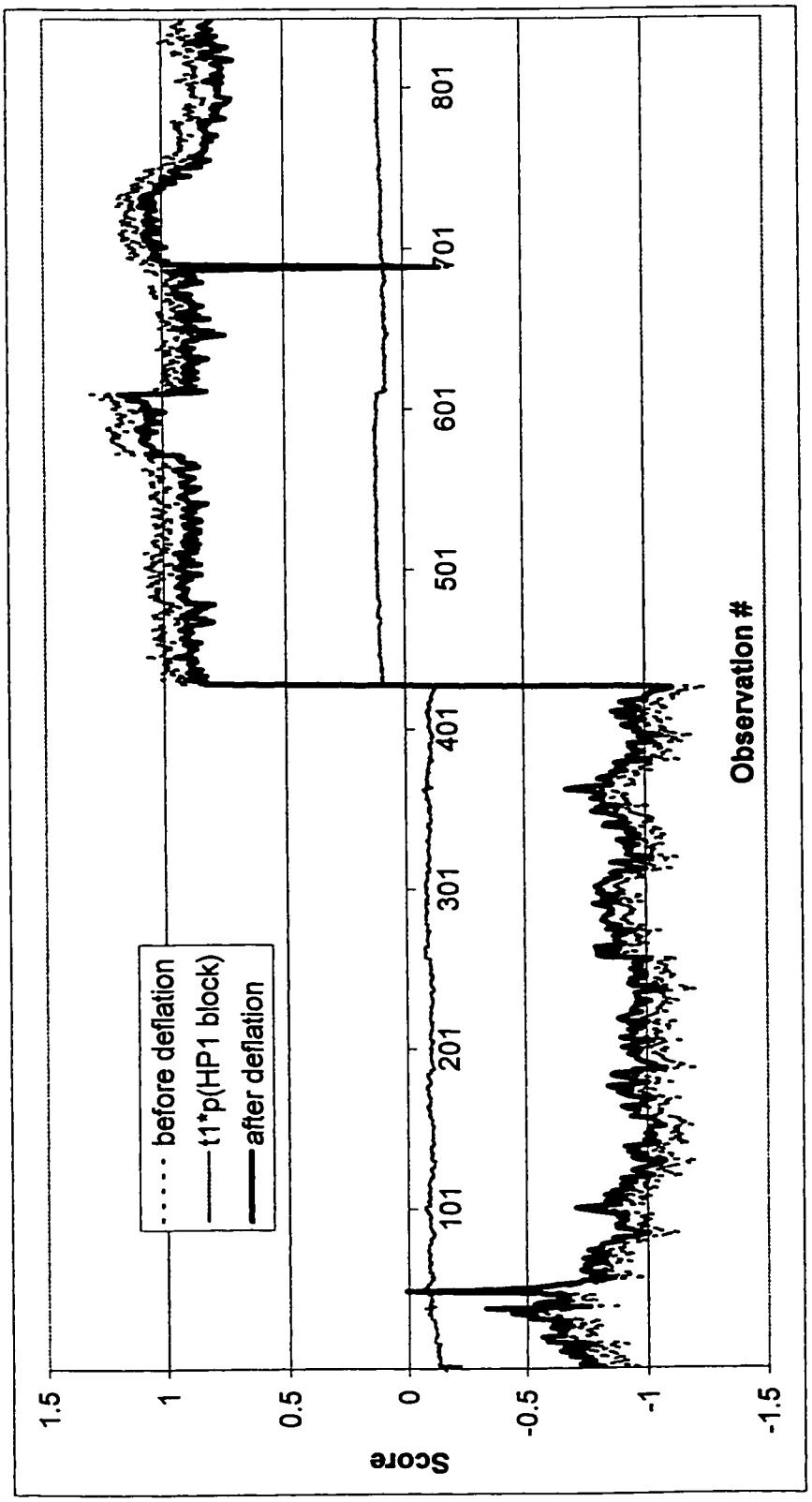


Figure 4.9 : Deflation Sequence for HP1, Channel D Using Level 1 Score

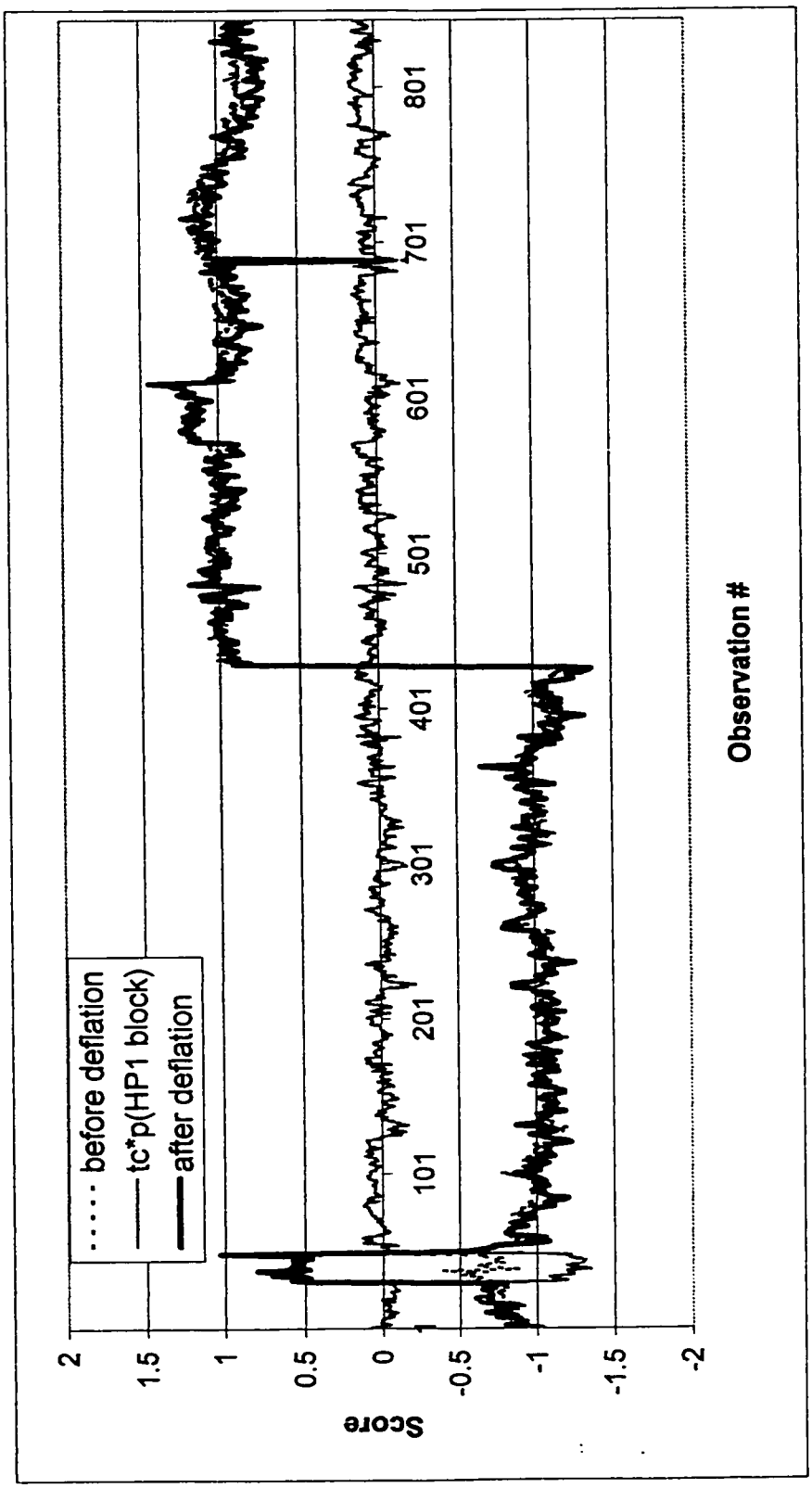


Figure 4.10 : Deflation Sequence for HP1, Channel D Using Level 3 Score

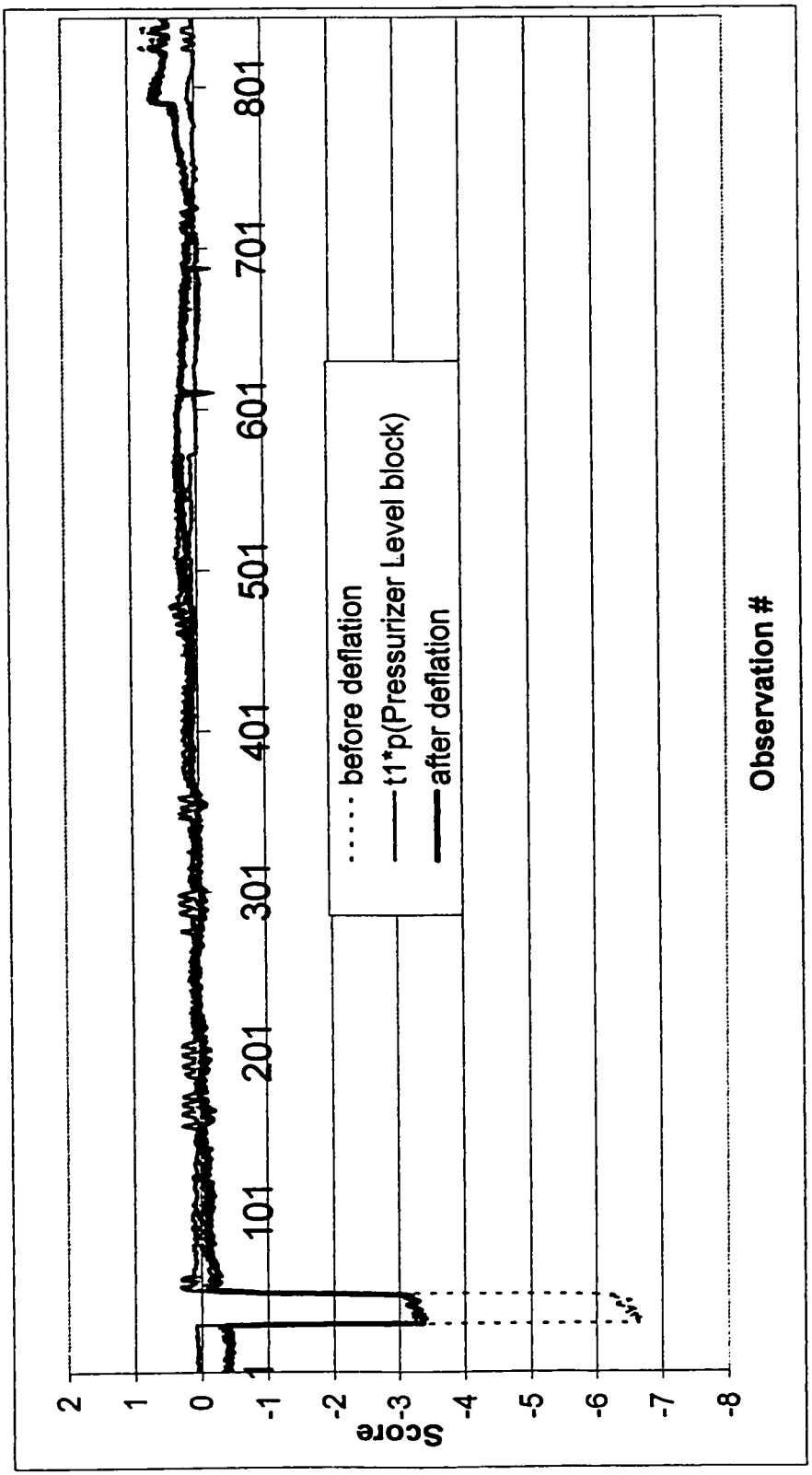


Figure 4.11 : Deflation Sequence for Pressurizer Level, Channel D Using Level 1 Score

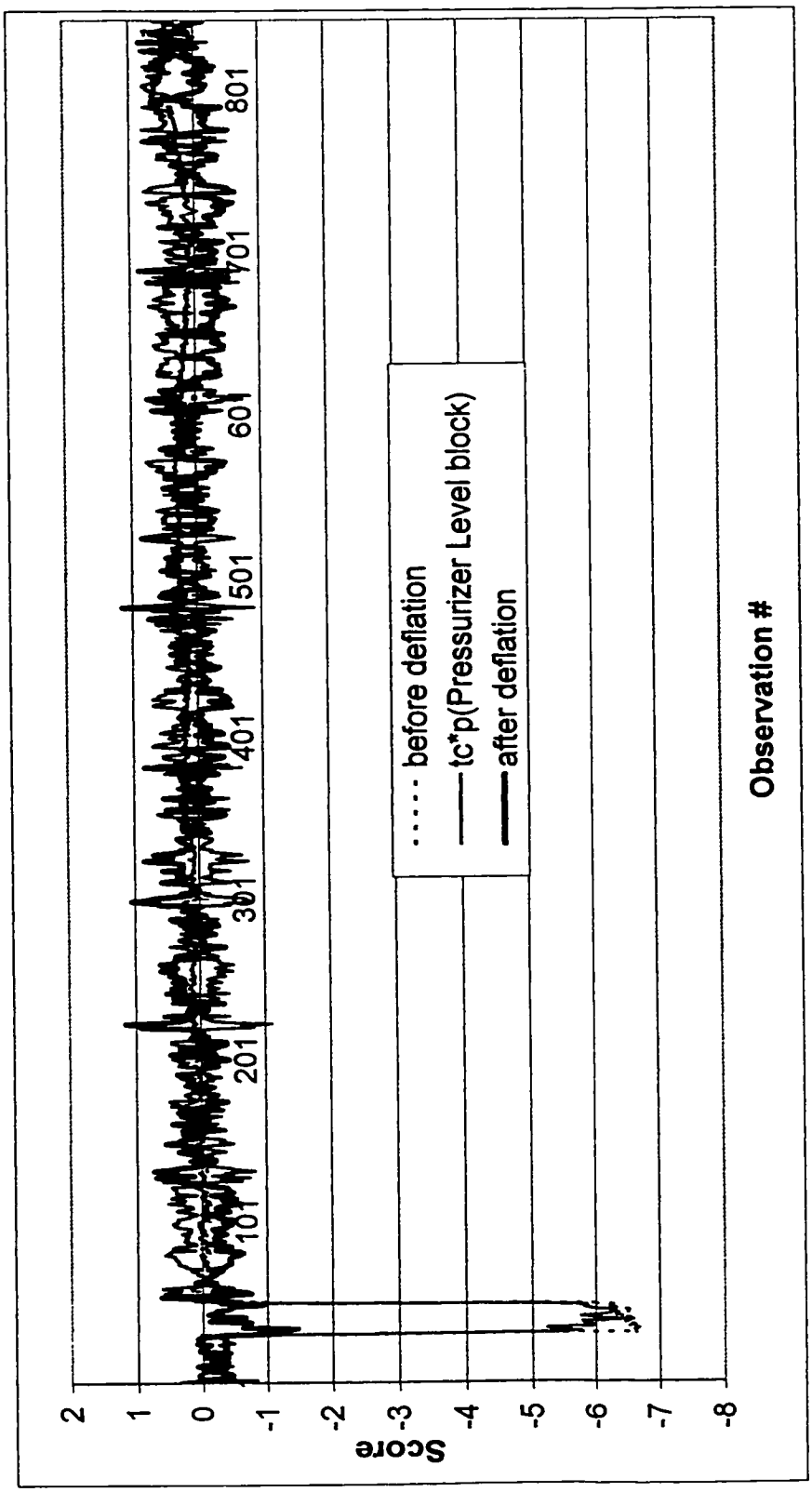


Figure 4.12 : Deflation Sequence for Presurizer Level, Channel D Using Level 3 Score

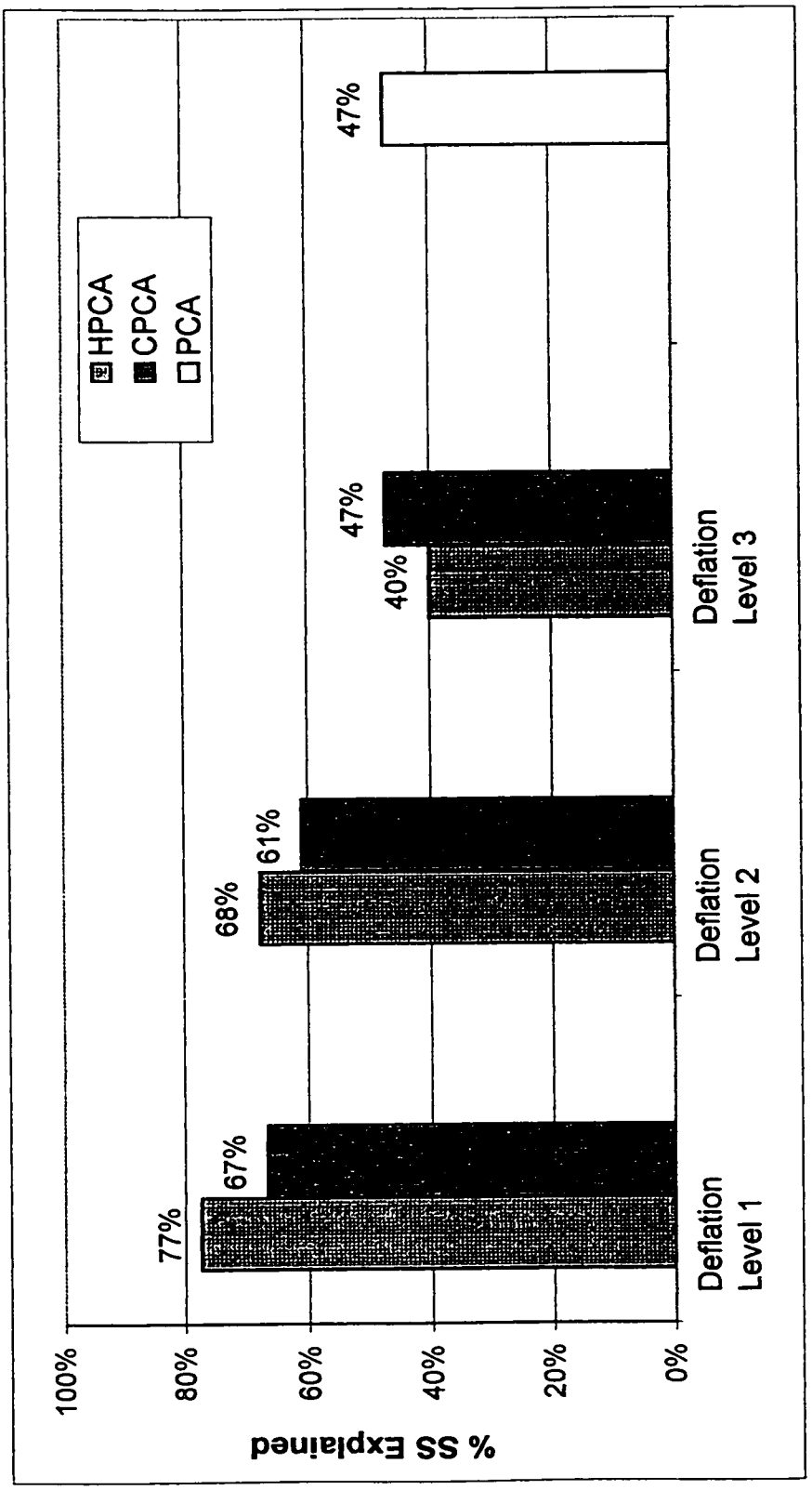
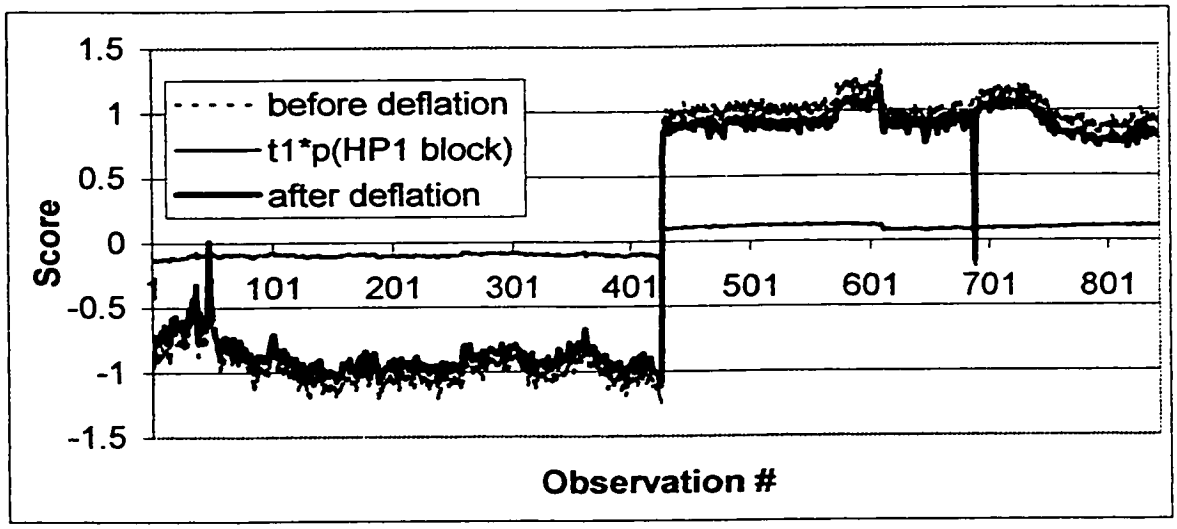
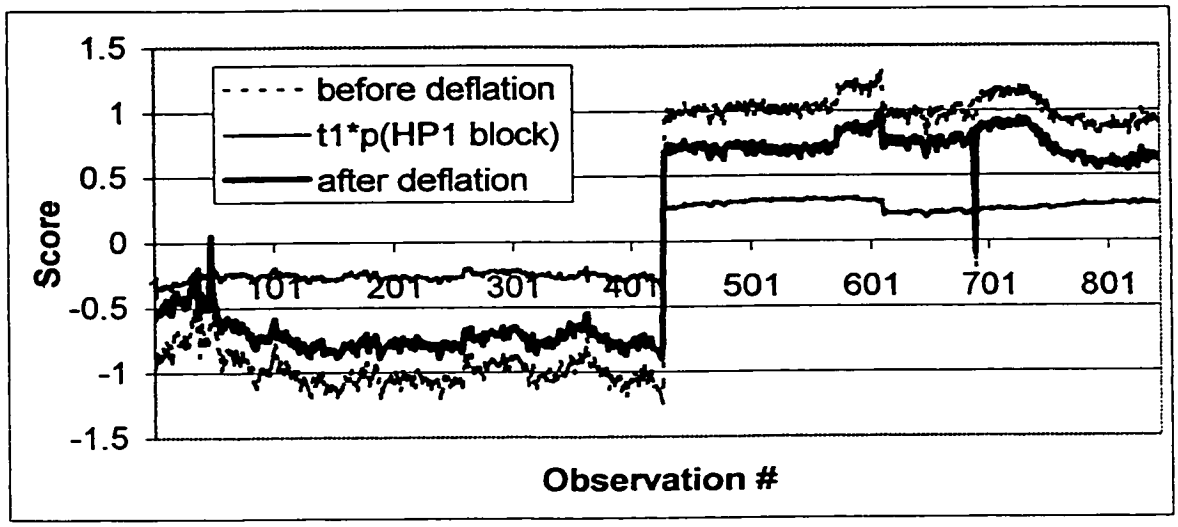


Figure 4.13 : Sum of Squares Explained for September Data Set (4 PC models)



CPCA Deflation Sequence for Header Pressure 1, Channel D
Using Level 1 Score (same as Figure 4.9)



HPCA Deflation Sequence for Header Pressure 1, Channel D
Using Level 1 Score

Figure 4.14 : Comparison of Deflation Sequence for CPCA and HPCA

CHAPTER 5

NPP Analysis and Monitoring Using Various PCA Techniques

5.1 Introduction

This chapter will present the results of using standard PCA techniques and multi-block, multi-level PCA for the analysis and monitoring of a CANDU NPP. The chapter will be divided into three main sections. The first section will discuss standard PCA utilization for an initial analysis of historical data from a CANDU reactor. In this section standard PCA is used to identify obvious outliers in the data. The analysis will also be used to decide which data will be considered steady state and used to develop the reference model. This is always an important step for developing monitoring methods using SPC, as was discussed previously. The second section will deal with using standard PCA techniques for process analysis. The goal of this research was to investigate the use of PCA to gain a deeper understanding and insight into the process. Finally, the feasibility of process monitoring using multi-block, multi-level PCA will be discussed in the third section.

5.2 Initial SPC Analysis Using Standard PCA

As discussed earlier, the first step in developing a SPC scheme for a specific process is to identify a historical dataset which can be used to develop the monitoring model. As was stressed previously, this dataset must contain only natural or inherent variability. So,

while on the surface, this may look like a straight-forward step, much care must be taken in identifying the correct dataset. First, the dataset must be in a useful format. This alone can be a formidable task and is the subject of Appendix A for this project. Once the data is in a useful form for analysis, it needs to be examined for outliers which are not considered inherent variability. For this project, three historical datasets were available; two weeks of data from each of Nov./95, March/96 and Sept./96. It was decided to use one of the datasets to develop the monitoring model and the other two as testing data. Although all datasets were supposed to represent normal, steady state data, each one was analysed to determine if one set would be a better choice. The first 10 days of each dataset was used for this initial analysis. Using 15 minute averages, this resulted in 960 observations per dataset. The first obvious step was to plot the raw data, in the form of run charts described in Chapter 1. These plots are given in Appendix E. As observed from these plots, there are some obvious outliers in each of the datasets. However, their cause is not immediately known and outliers should not be removed unless there is a reasonably good explanation for their cause. In order to help determine the cause of the outliers, a simple PCA was performed on each dataset. Each PCA used 2 principal components and the associated models were compared in various ways. It is noted that for this and all subsequent models, the data were mean-centered and auto-scaled, as discussed in section 2.2.1.

Figure 5.1 shows the eigenvalues for each of the three models. Recall from Chapter 2 that the eigenvalues can be used as a measure of the variability explained by each principal

component. If there are strong correlations among certain variables in the data, the associated principal components will have relatively large and well spaced eigenvalues compared to the remaining eigenvalues. At the other end of the spectrum, if all of the variables were completely uncorrelated, all eigenvalues would have approximately the same value and if the data were mean centered and auto-scaled, the values would be approximately 1.0. For November, there were three relatively large eigenvalues associated with the first three principal components. This indicates that the first three principal components explain a significant portion of the variability in the dataset. Table 5.1 shows that the first two principal components explain 32% of the sum of the squares. Said another way, this indicates that there are three relatively strong directions in the dataset. For March, only the first eigenvalue is relatively large while for September, the first two eigenvalues are large and they are approximately the same value. This indicates that they explain approximately the same amount of variability, which is also shown in Table 5.1. Finally, it is noted that after the first ten eigenvalues, all three models follow the same general trend.

Month	PC1	PC2	Cumulative
Nov.	20.1%	12.2%	32.3%
March	16.3%	7.8%	24.1%
Sept.	10.3%	10.2	20.5%

Table 5.1: Sum of Squares Explained for Each Model

Next, the models were compared in terms of their loadings. The loadings for the first two principal components for the three models are shown in Figures 5.2 and 5.3 respectively.

As observed in Figures 5.2 and 5.3, the loadings for PC1 and PC2 do not look the same for the three datasets. The differences for each principal component are shown in Table 5.2 which highlights the fact different variables contribute significantly to the loadings for each month.

Principal Component	Variables which Contribute Significantly		
	November	March	September
PC1	Pressurizer levels Log N All boiler levels	Boiler 7 levels Feedline pressures	Boiler 2 levels Boiler 6 levels Feedline pressures
PC2	Header pressures	All boiler levels	Log N Moderator Temps

Table 5.2: Significantly Contributing Variables for PC1 and PC2

This was a first indication that the three datasets may not represent the same steady state operating point and therefore a model based on one month may not be valid for other months.

Next, the first two scores for each model were plotted against each other. These plots are shown for the November, March and September data in Figures 5.4, 5.5 and 5.6 respectively. The November model has three distinct clusters whereas the March and September data seem to be better behaved. One could argue that the September data also have some clustering however, it is not as defined as in the November model.

Finally, the SPE for each model were compared, as shown in Figures 5.7, 5.8 and 5.9. From Figures 5.7 – 5.9, it was decided that the September dataset was the best behaved. This was based on the observation that there were 10 groups of outliers for September vs. 14 groups for March and approximately 20 groups for November. Based on the above

analysis of the loadings, scores and SPE, it was decided to use the September dataset as the reference dataset. This decision was based mainly on the SPE results and to a lesser degree on the score comparison.

As stated previously, there were two main goals of this PCA analysis; identify the reference dataset and identify any outliers which should be eliminated from the reference dataset. With September identified as the reference dataset, the task now turned to outlier identification. To do this, the contributions to the SPE outliers shown in the 10 groups in Figure 5.9 were analysed. The contributions to these outliers were easily related to process trip tests. A process trip test is a test were an instrument is valved out of service and its input is raised above its set point or limit to ensure that the process trip logic is correct and the proper actions take place. The groups of outliers were associated with various process trip tests by the way the contributions to the SPE cycled through specific groups of variables. For example, the main contributions to the SPE for observations 125 to 129 (group 2) are shown in Figure 5.10 and are summarized in Table 5.3.

Observation Number	Main Contributors to SPE
125	Differential Pressure 2-3, Ch G
126	Differential Pressure 2-3, Ch G Differential Pressure 1-4, Ch G
127	Differential Pressure 2-3, Ch. G Differential Pressure 1-4, Ch. G Pressurizer Level, Ch. G
128	Pressurizer Level, Ch. G Boiler 2 Level, Ch G Boiler 3 Level, Ch G Boiler 6 Level, Ch G
129	Pressurizer Level, Ch. G Boiler 6 Level, Ch G Boiler 7 Level, Ch G

Table 5.3: Main Contributors to SPE for Observations 125-129

The main contributors cycle through the differential pressures, pressurizer level and boiler levels for Channel G. This indicates process trip tests for the sensors of the associated variables in Channel G. Similarly, the contributions to observations 682 to 684 (group 7) are shown in Figure 5.11 and Table 5.4.

Observation Number	Main Contributors to SPE
682	Log N Rate, Ch. G
683	Header Pressure 1, Ch G
684	Header Pressure 2, Ch. G Feedline Pressure, Ch. G

Table 5.4: Main Contributors to SPE for Observations 682-684

Again, these results would indicate process trip tests for the sensors associated with Log N Rate, header pressures and feedline pressure for Channel G. Similar results were obtained for 7 of the other 8 outlier groups shown in Figure 5.9. The final group which should be discussed is Observation 488 (group 6). The contributions to the SPE for this observation are shown in Figure 5.12. As seen in Figure 5.12, only one variable contributes significantly to the error. That variable is Header Pressure 1, Channel F. This indicates that this outlier is probably not a result of a process trip and may have been caused by an erratic error or fault in the sensor. From this analysis, it is easy to rationalize removing the outliers which can be associated with process trip tests. These tests do not represent normal inherent process variability because the plant is not operating under normal conditions. It was also decided to remove observation 488 as again, it does not appear to represent normal inherent variability. This reduced the number of observations from 960 to 905.

Finally, the outliers associated with the scores, as shown in Figure 5.6 were analysed by looking at the contributions to the shift in scores for points outside the control limits. The contributions to the shift in t1 for Observation 34 from the center of the data, Observations 181-182 and Observations 373-374 are shown in Figures 5.13 to 5.15 respectively. As shown in these figures, the main variables contributing to the shift in t1 for these observations are Boiler 2 and 6 levels and the feedline pressure. This is consistent with the earlier analysis which showed that the variables which had the most significant weights in the first loading for the September data were the levels from Boilers 2 and 6 and the feedline pressures. The analysis for shifts in t2 is not as clear. Figure 5.16 shows that the contributions to the shift in t2 from the center of the data to observation 817 are from various header pressure, Log N and differential pressure sensors. This uncertainty in the cause of the shift may be due to the fact that observation 817 is just marginally outside the control limits. However, the number of sensors which could be considered to have significant weights in the second loading should be looked at in more detail. The whole issue of interrupting the shifts in scores will be investigated in detail in Section 5.3.2. For the purposes of defining a reference dataset, it was decided to keep all the observations outside the control limits in Figure 5.6. This was done because the causes of these outliers were not as clear as for the SPE analysis. As stated above, observation 817 was very close to the limit and its cause was not clear. For the outliers associated with shifts in t1, the contributing variables were clear. However, it was debatable as to whether these outliers were caused by larger than normal variations in the correlations associated with first principal but were still acceptable or if they truly

represented process faults. It was decided that although they were outside the control limits, they would still be used in the reference dataset because the plant personnel considered all the data normal steady state data, with the exception of the process trip tests.

One final step in the initial data analysis was completed. The September dataset was checked for autocorrelation with the dynamic PCA method described by Ku[30]. The results found that there was no autocorrelation in the data and hence the data did not need to be lagged. This result was expected as the time constants for the reactor would vary from seconds for the reactor physics and local boiler dynamics up to a minute for the overall heat transport system. The overall heat transport lags are large due to the large water inventory in the boilers. Therefore, the 15 minute averages have eliminated any autocorrelation that may have been present in the raw data.

In summary, using PCA for the initial analysis of a historical dataset was very useful. Process trip tests were easily identified and eliminated and some issues associated with identifying a reference dataset were highlighted.

5.3 Process Analysis

As stated in the introduction, the goal of this section was to investigate the usefulness of PCA to gain a deeper understanding and insight into the process. This analysis was completed in two steps. First individual PCA models were calculated for individual

variables. Secondly, a PCA was completed on all of the data available at once. Each of these investigations will be discussed in detail below.

5.3.1 Individual PCA Models for Specific Variables

This work was reported on at the 1997 CNS conference [51]. The November data were used as these were the only data available at the time of the analysis. This individual analysis was completed while the multi-block, multi-level PCA algorithm was being tested. Individual PCA models were developed for the 6 following variables:

1. Header Pressure (12 transmitters)
2. Pressurizer Level (6 transmitters)
3. Boiler Level (23 transmitters)
4. Boiler Feedline Pressure (6 transmitters)
5. Differential Header Pressure (5 transmitters)
6. HTS Flow (6 transmitters)

It should be noted from the above list that one boiler level and one differential pressure were deleted from the dataset. The boiler level signal was deleted because it was recalibrated during the 10 day period. The differential pressure signal was deleted due to what appeared to be excessive noise. However, the signal would still be capable of producing a reactor trip. Also, the averaging time was reduced from 15 minutes to 3 minutes. This was done at AECL's request to decrease the time required to detect the pressure transmitter faults. A 3 minute average was found to still reduce the noise to an

acceptable amount [52]. Table 5.5 summarizes the PCA models for each of the 6 variables listed above.

PC#	Head Pres		Pres Lev		Boil Lev		Fdli Pres		Diff Pres		HTS Flo	
	%SS	Cum	%SS	Cum	%SS	Cum	%SS	Cum	%SS	Cum	%SS	Cum
1	99.3	99.3	96.8	96.8	41.2	41.2	83.4	83.4	81.6	81.6	47.7	47.7
2	---	---	---	---	20.7	61.9	13.6	97.0	13.7	95.3	26.4	74.1
3	---	---	---	---	20.3	82.2	2.4	99.4	---	---	---	---
4	---	---	---	---	17.2	99.4	---	---	---	---	---	---

Table 5.5: PCA Model Summaries

In all cases except the flowrates, the first principal component represented an average of the transmitters. This was determined from the fact that all the weights in each of the first loading vectors were approximately the same. This was expected as all redundant sensors were highly correlated about their mean. For the header pressures and pressurizer levels, the first principal component represented over 95% of the variability or sum of squares in the dataset. Therefore, for these variables, one principal component was used in the model even though some of the significance tests indicated that more than one principal component should be used. For the other variables, additional principal components were required. For the boiler levels and differential pressures, the additional principal components described the variability associated with correlations between groups within the variables. For example, the loadings for the four principal components for the boiler levels are shown in Figure 5.17. As observed in Figure 5.17, the loadings for the second principal component consist of large negative and positive values for boilers 2 and 3 respectively and smaller negative and positive values for boilers 6 and 7 respectively. The same general trend , only with the larger negative and positive values for boilers 6 and 7

is observed in the fourth principal component. The variability explained by these principal components can be interpreted as the variability caused by levels of boilers 2 and 3 moving in the opposite directions to each other and the levels in boilers 6 and 7 moving in opposite directions to each other. By the same analysis, the variability explained by the third principal is the variability caused by the levels in boilers 2 and 3 moving in opposite directions to boilers 6 and 7. This would seem to make physical sense as boilers 2 and 3 are fed off one reactor outlet header and boilers 6 and 7 are fed off the other reactor outlet header located on the opposite side of the reactor.

The loadings for the HTS flowrates are shown in Figure 5.18. Flows 1, 3, 4, and 6 are highly weighted in the first PC while flows 2 and 5 are highly weighed in the second. This appears to be a result of the location of the channels where the flow is being measured. Flows 1, 3, 4, and 6 are measured in channels labeled Q (Q17, Q9, Q8, and Q16 respectively) while Flows 2 and 5 are measured in channels labeled J (J12 and J13 respectively). Finally, the additional principal components for the feedline pressures explained some of autocorrelation in the signals. The autocorrelation was found by lagging all the signals by one time step or 3 minutes. It should be noted that the other variables were also checked but only the boiler feedline pressure signals were autocorrelated. Again, the result is not unexpected as the time constant for the overall heat transport system could be up to a minute, as stated in Section 5.2, and the averaging was reduced from 15 to 3 minutes.

In summary, this analysis was able to provide some key insight into relationships among groups of sensors for specific variables. Specifically, interactions between boilers and the locations of the flow measurements were highlighted. These interactions would have been difficult to observe from basic run charts alone. Finally, it was found that the first score represented the mean of the sensors, as was expected.

5.3.2 PCA Model for All Data

The goal of this section was to determine what insight could be gained into the process by analyzing all of the available data at once using PCA. In order to do this, the process trip tests identified in Figures 5.7 and 5.8 for the November and March data were eliminated, as was done for the September data. This resulted in 843 observations for the November dataset and 864 observations for the March dataset for a total of 2612 observations. A 2 principal component PCA model was calculated for this entire dataset. The first PC explained 35.4% of the SS while the second PC explained 25.0%. As shown in Figure 5.19, the first two eigenvalues are large and distinct. The model had four distinct clusters in the t1 vs. t2 space, as shown in Figure 5.20. In order to determine the cause of the clustering, loading, score and SPE analysis were completed. The loadings for the first and second principal components are shown in Figure 5.21. From these loading plots, two distinct cases were observed. In the first case, all the sensors from the same variable were given a large weight. For the first PC, all the sensors associated with the pressurizer level, Log N, feedline pressure, and moderator temperature have large weights. An examination of the raw data shows that the average of these variables changed for the three different

datasets. An example of this trend is shown in Figure 5.22 for the pressurizer levels. This behaviour would indicate a process change, perhaps the plant was moving from one operating point to another. In the second case, one or two sensors from the same variable were given large weights. Again, for the first PC, there were two large weights for header pressure 1. They were for channels H and J. For header pressure 2 there were also two large weights, in channels D and J. From the raw data for header pressures 1 and 2, shown in Figures 5.23 and 5.24, it is observed that there is a change in the individual channels listed above. Figure 5.23 indicates Header Pressure 1, Channel J changes significantly from November to March while the other channels stay relatively the same. Channel H increases from March to September while all the other channels decrease. Similarly for Header Pressure 2, shown in Figure 5.24, Channel J changes significantly during the month of March while Channel D changes significantly from November to March and to a lesser degree from March to September. A similar analysis can also be done for boilers 2 and 3. For Boiler 2, the largest weight is associated with Channel G. Boiler 3 level has a large weight for sensor J. Again, the raw data, given in Figures 5.25-5.26 shows that these individual sensors are changing significantly. These individual sensor changes can be attributed to calibrations. Therefore, the loadings for the first PC represent both process changes which affect all sensors for a given variable and calibrations which affect individual sensors. An analysis of the loadings for the second PC shows the same general trends. All sensors for both header pressures as well as the pressurizer levels have large weights indicating process changes. Also, boiler 2 level, Channel E and boiler 6 level, Channel G have large individual weights indicating

calibrations. Finally, it should be noted that this analysis has not been exhaustive. There are other individual sensors with large weights in both principal components.

Given this loading analysis, where the weights in the first two PC's can be related to both process changes and calibrations, interpretation of the contributions to shifts in the scores is not an easy task. This difficulty was first encountered in Section 5.2. Figure 5.27 shows the contributions to the shift in t1 from the March to September data. A close analysis of this plot shows that there are significant contributions from Boiler 2 Level Ch. E and G, Boiler 3 level Ch. J, Flow 1 Ch. E, Flow 2 Ch. D and E, Log N Rate Ch. F and to a lesser degree from most of the sensors associated with the pressurizer levels, Log N and Feedline pressures. However, as a whole, the analysis of this plot is not straightforward and somewhat cumbersome.

A plot of the SPE for this model is shown in Figure 5.28. There are basically four groups of outliers or observations which lie above the SPE limit. Again, in order to investigate the cause of these outliers, the contributions from the individual variables were examined. Contribution plots for observations 34 and 45 which are in the first group of outliers are shown in Figure 5.29. The main variables contributing to these outliers are pressurizer levels, Log N values and all boiler levels. An analysis of the raw data for November shows that the pressurizer levels and Log N values changes suddenly for Observations 34-50. The boiler levels also changed to a lesser degree. It was determined that this behaviour was associated with a power change in the plant. The next outlier, Observation

539, had only one significantly contributing variable, the feedline pressure from Channel F. This is shown in Figure 5.30. It is suspected that this observation is a true outlier with respect to this one sensor. The final two groups of outliers, Observations 1573-1574 and 1873 are very interesting. They both have the same contributing variables, as shown in Figures 5.31 and 5.32. As seen in these figures, the main contributors are the feedline pressures, Boiler 3 levels and Boiler 7 levels to a lesser degree. The same trend was also seen in Observations 1484, 2060 and 2348, which are close to the SPE limit. These observations would indicate that there was some sort of process upset involving the feedline pressure and Boilers 3 and 7. Although no further information can be obtained from the analysis, this knowledge provides a very useful starting point for a more in-depth investigation.

In summary, the PCA on the three months of data provided some very useful information and insight into the process. The SPE analysis was very useful for detecting process upsets. It was found that individual variables contributed to the loadings of the PC's in two general ways which could be related to either process changes or sensor calibrations. Finally, for this type of data, the analysis of the contributions to shifts in scores is somewhat confusing due to the way the variables contribute to the loadings, as discussed above.

5.4 Process Monitoring

In this section, the feasibility of monitoring a NPP using CPCA will be investigated. Recall from Chapter 3 that the motivation for using CPCA was to deliver relevant information to different functional groups with the NPP. The assessment of NPP monitoring with CPCA was completed in two steps. First, a sensitivity analysis was performed using a CPCA model developed from the reference dataset. This was done to determine if the CPCA model could detect sensor errors on the lowest or technician level while filtering out the errors on the higher levels. Secondly, the reference model was tested with data from the other months to determine if long term monitoring would be feasible.

5.4.1 CPCA Model Development

The CPCA model was developed using 4 principal components. The loadings were normed and the deflation was done using the first level scores, as described in Section 4.4. The model was developed using the first 10 days of the September dataset, based on the results discussed in the Section 5.2. The remaining data was used as an initial test of the model. The SPE for the development and testing of the September model are shown in Figures 5.33 and 5.34 respectively. These graphs highlight the usefulness on the CPCA scheme. Any fault which might affect the overall SPE can be easily tracked back to specific groups of sensors. This is opposed to the standard PCA model which gives only one overall SPE and contribution plots need to be used immediately from this. The usefulness of the CPCA scheme will be demonstrated in the next two sections.

5.4.2 Sensitivity Analysis

The sensitivity analysis was performed by adding and subtracting the offsets to and from each variable listed in Table 3.2. This was done initially for the September data. Ideally, it was hoped that the errors would be detected in level 1 of the CPCA model but not level 2 or 3. The results are displayed in figures such as Figure 5.35, which shows the results on an expanded scale for adding the 50 kPa offset to each of the header pressures. The x axis shows the results for 13 different trials, the 12 offset trials and one trial where no offset was added. The y axis shows the results for the different levels in the CPCA model plus the results for the normal PCA. Finally, the z axis shows the percentage of data points above the SPE limit for each case. The results for all the variables are given in Figures 5.36-5.38 in this format. Some key observations from these graphs are as follows. The results are generally the same for the addition or subtraction of the offset errors, as shown in Figures 5.36-5.38. The models seem to be more sensitive to offset errors in some variables than in others. This is highlighted in Figure 5.39, which shows that the CPCA model is sensitive to errors in the header pressures but not as sensitive to errors in Boiler 2 Level. This observation led to the decision to check the consistency of the sensitivity analysis from month to month. Similar models were built and the same simulations were run for the Nov. and March datasets. The results of these tests are shown in Figures G1-G6, in Appendix G. As observed in these figures, the results are not consistent from month to month. This is highlighted in Figure 5.40, which shows the header pressure results for Sept. and Nov. This variable to variable and month to month inconsistencies were investigated by plotting the square root of the average SPE for each

trial against the ratio of the applied offset error to the standard deviation for each variable. Various combinations were plotted as shown in Figure 5.41. In all cases, the relationship was linear and approximately 1:1. Therefore, it was concluded that the sensitivity of the model for each variable was a function of the original scaling of the variable. In this case,

$$\overline{\mathrm{SPE}} \approx \left(\frac{\text{applied offset error}}{\text{standard deviation}} \right)^2 \qquad 5.1$$

This result is as expected. If there are no other faults present in the data, the squared prediction error should be equal to the artificially applied error, properly scaled. In this case, the applied offset error should be divided by the standard deviation of the sensor being examined. The issue of scaling and its affect on the analysis will be examined in more detail in the next section.

Finally, the ability of the multi-level CPCA algorithm to filter out the sensor faults in higher (supervisor, manager) levels was examined. Overall, the results again vary from variable to variable and month to month with generally marginal performance. The best example of the methodology working as hoped is shown in the results for the Boilers 2 and 7 levels for November. Here, the errors are detected very clearly in the first level while not at all in the second and third levels. The results for Boilers 2 and 7 levels are also reasonably good for the September and March data. At the other extreme is the results for the header pressures for September, shown in Figure 5.36. Here the errors are clearly detected in all three levels. The same results hold for the March header pressures.

For the November data, the errors are detected in the first two levels but not the third. The marginal performance could be related to the scaling and, as mentioned above, will be discussed in the next section.

5.4.3 Model Validity

An obvious requirement for process monitoring is that the results from the methodology used must remain valid and meaningful over the long term. However, the results from Sections 5.2 and 5.3 would indicate that a model calculated using the data from one month would not be valid for the other months. This was confirmed by analysing the March and Nov. data with the four PC CPCA model calculated from the Sept. data. The results for the SPE are shown in Figures 5.42 and 5.43. As observed, the overall SPE indicates that there is a very significant problem with both the Nov. and March data, when compared to the Sept. data. However, this is not the case, according to the source of the data, AECL. As a side note, Figures 5.42-5.43 show that the CPCA model was very useful for quickly identifying which groups of sensors or variables were causing the large SPE. Consider, for example, the March SPE data shown in Figure 5.43. From the second level, it is easily determined that all variables except the feedline pressure are contributing significantly to the overall SPE. From the first level, it is easily determined that both header pressures contribute significantly to the overall header pressure SPE while only Boilers 2 and 3 contribute significantly to the overall boiler lever SPE.

Originally, it was thought that main factors causing the differences between the September, March and November data were changes resulting from the process moving to different operating points and sensor calibrations. In order to confirm this hypothesis, the March data was investigated in detail. First the header pressures were examined. Figure 5.44 shows three graphs related to the SPE for Header Pressure 1 for the March dataset. Figure 5.44a shows the SPE, of which the average is 243 and the 99% control limit is 9. Again, this clearly indicates there is a problem. The next step in the detailed analysis was to examine the contributions to this SPE. The average contributions to the SPE from the six sensors for Header 1 are shown in Figure 5.44b. The main contributions to the SPE come from the sensors associated with Channels E and H. Using these average SPE's and Equation 5.1, the expected offset errors were estimated and are shown in Figure 5.44c. Finally, the actual differences between the average Header 1 pressures for March and September were calculated from the raw data and are also shown in Figure 5.44c. Two important points can be drawn from this graph. First, the estimated and actual offset errors agree reasonably well. It should be noted that for Channel H, the actual offset error is negative. This means that the average March value is actually less than the average September or reference value. This information is not available from the offset error estimated from the SPE. The second and more important point is that all of the estimated and actual offset errors are well below the 50 kPa. Recall from Table 3.2 that 50 kPa is the minimum error which would be considered significant. Therefore, although the actual offset errors are causing a large SPE, in the final analysis, they would not be considered significant and in all likely hood be treated as false alarms. The cause of this

elevated SPE can be traced back to the scaling methodology used in the model development. As was noted in Section 5.2, the data were mean-centered and auto-scaled before any models were developed. This meant that the values associated with each pressure sensor for Header 1 were divided by their standard deviations which are shown in Table 5.6.

Channel	Standard Deviation (kPa)
D	4.2
E	2.1
F	3.2
G	1.3
H	2.0
J	2.0

Table 5.6: Standard Deviations for Header 1 Pressures

As observed in Table 5.6, the standard deviations are one order of magnitude smaller than the error which would be considered significant. To put this into perspective, the SPE was calculated using Equation 5.1, a 50 kPa error and the standard deviations given in Table 5.6. These values are shown in Table5.7.

Channel	Standard Deviation (kPa)	SPE for 50 kPa error
D	4.2	142
E	2.1	567
F	3.2	244
G	1.3	1479
H	2.0	625
J	2.0	625

Table 5.7: SPE Given 50 kPa Offset Error

Given the 99% SPE control limit is 9, Table 5.7 clearly indicates that the this monitoring methodology is too sensitive. One obvious solution to this problem would be to simply

increase the SPE limit. However, as seen in Table 5.7, the SPE for a 50 kPa offset error varies from 142 to 1479. Therefore, one SPE limit could not be applied to all six sensors. The same analysis was completed for Header 2 Pressure. The results are shown in Figure 5.45. For Header 2, there are two distinct cases. Up to Observation 800 the average SPE is approximately 559 and after Observation 800 the average SPE is approximately 63. In both cases, the average SPE is well above the control limit of 5. The contribution plots related to the two cases are shown in Figure 5.45 along with the estimated and actual offset errors. In the first case, the main contribution comes from Channel J. The estimated offset error is 48 kPa while the actual error is 60 kPa. In this case, the error is significant and should be detected. In the second case, after Observation 800, the main contributions are from Channels D and F. However, the estimated and actual offset errors are well below the 50 kPa threshold. Overall, it is believed that this trend represents a re-calibration of Channel J. If plant records indicated that this is the case, the new mean for Channel J could be substituted in the model. However, the issue still remains that after the re-calibration, the SPE remains well above the limit, indicating a fault when, in reality, there is not one. The same problem was found in all of the variables to a greater or lesser degree. Figure 5.46 shows the ratio of the significant offset error to the standard deviation for all sensors. Figure 5.46 shows that for the header pressures, pressurizer levels and differential pressures, the significant offset errors are an order of magnitude or more greater than the standard deviations used for the scaling. The significant offset errors for the boiler levels and feedline pressures are approximately twice as large as the

standard deviations. This analysis clearly indicates that auto-scaling is not the correct scaling method for this type of data and monitoring objectives.

One possible solution to this scaling issue would be to scale the variables by the significant offset errors. This solution has merit on two points. First, if all the sensors for a given variable are the same, that is, the same type of sensor made by the same manufacturer, the variability in the sensors should be relatively constant. Secondly, this type of approach to scaling was discussed by Kresta et. al. [23]. In this paper, Kresta suggested using specification ranges for quality variables or sensor ranges for process variables as a natural alternative to auto-scaling. An analysis was completed using this new method of scaling. A model was developed using the September data and tested using the March data. The results for Headers 1 and 2 pressures are shown in Figures 5.47-5.48. Several interesting points can be noted from these graphs. First, regarding the limits, it is seen that the SPE is still much greater that the 99% limit. However, using constant scaling factors of the significant offset errors, a critical SPE limit of 1.0 can be defined by Equation 5.1. It is observed in Figure 5.47 that the average SPE is 0.6 which is below 1.0. This would indicate that there are no faults present, which is the desired response. Also, it is noted that the estimate of the offset error for the new scaling procedure is the same as the actual error, as opposed to the estimate from the auto-scaling procedure which was close but not exactly the same. In Figure 5.48, the average SPE is 1.5 up to Observation 800, indicating there is a problem. Again, this is the desired response as Channel H sensor is in need of a re-calibration. After Observation 800, the

average SPE drops to 0.14 which is once more the desired response. This drop was most likely due to a calibration of the sensor. The operational logs would need to be consulted to confirm this assumption. Finally, Figure 5.48 again shows that the estimated offset error for the new scaling procedure is better than the estimate using auto-scaling. A similar analysis was completed for the other variables. The results are given in Figures 5.49-5.52. For the boiler levels, Figure 5.49, Boiler 2 has a large average SPE of 21 and, indeed, there are two sensors, Channels E and G, which are well above the significant offset threshold of 5 cm. For Boilers 3 and 6, the average SPE is slightly above 1.0 and there are three sensors which are marginally above the 5 cm threshold. Finally, the average SPE for Boiler 7 is well below 1.0 and there are no significant offset errors. The same general trend is found in the flow analysis, shown in Figure 5.50. For Flow 1, the average SPE is slightly greater than 1.0 and there are two sensors slightly above the threshold. For Flow 2, the average SPE much larger than 1.0 and correspondingly, the reading from Channel D is significantly above the threshold limit. For the differential pressures, shown in Figure 5.51, the average SPE's are significantly below 1.0 and there are no significant offset errors. For the pressurizer levels, shown in Figure 5.52, the average SPE is 6.4 which is significantly greater than one. Figure 5.52 shows that the actual offset error for all six sensors is at or slightly above the 7 cm level. This would indicate that there was some process change that affected the pressurizer level and further investigation would be advisable.

To summarize, the process monitoring analysis has highlighted two important points. First, in developing a monitoring model, some consideration must be given to the magnitude of the faults which are desired to be detected and the magnitude of the scaling coefficients. In this instance, it appears that auto-scaling is not a feasible method. One alternative may be to scale the data by the minimum offset errors or faults should be detected. Areas for future work associated with this alternative will be discussed in the next chapter. The second important point is that, even with a better scaling procedure, process monitoring in this fashion may not be practical. As seen in Figures 5.47-5.52, even with scaling the data by the significant offset errors, some of the variables in the March data still appear to be out of control with respect to the September data. As discussed above, the large SPE's are occurring from either one sensor in a variable group, as is the case in Header 2 pressure and Boiler 2 level or from all the sensors in a variable group, as is the case for the Pressurizer level. However, the above analysis also found that the errors in the data are indeed larger than the significant offset errors defined in the original scope of the project. This would indicate that the monitoring procedure is functioning properly but having a monitoring method constantly indicating the process is out of control is not practical. This issue goes back to the fundamental questions of what are the objectives and goals of monitoring the process and what variability is considered normal for the process. This issue, along with other areas for future work, will also be discussed in the next chapter.

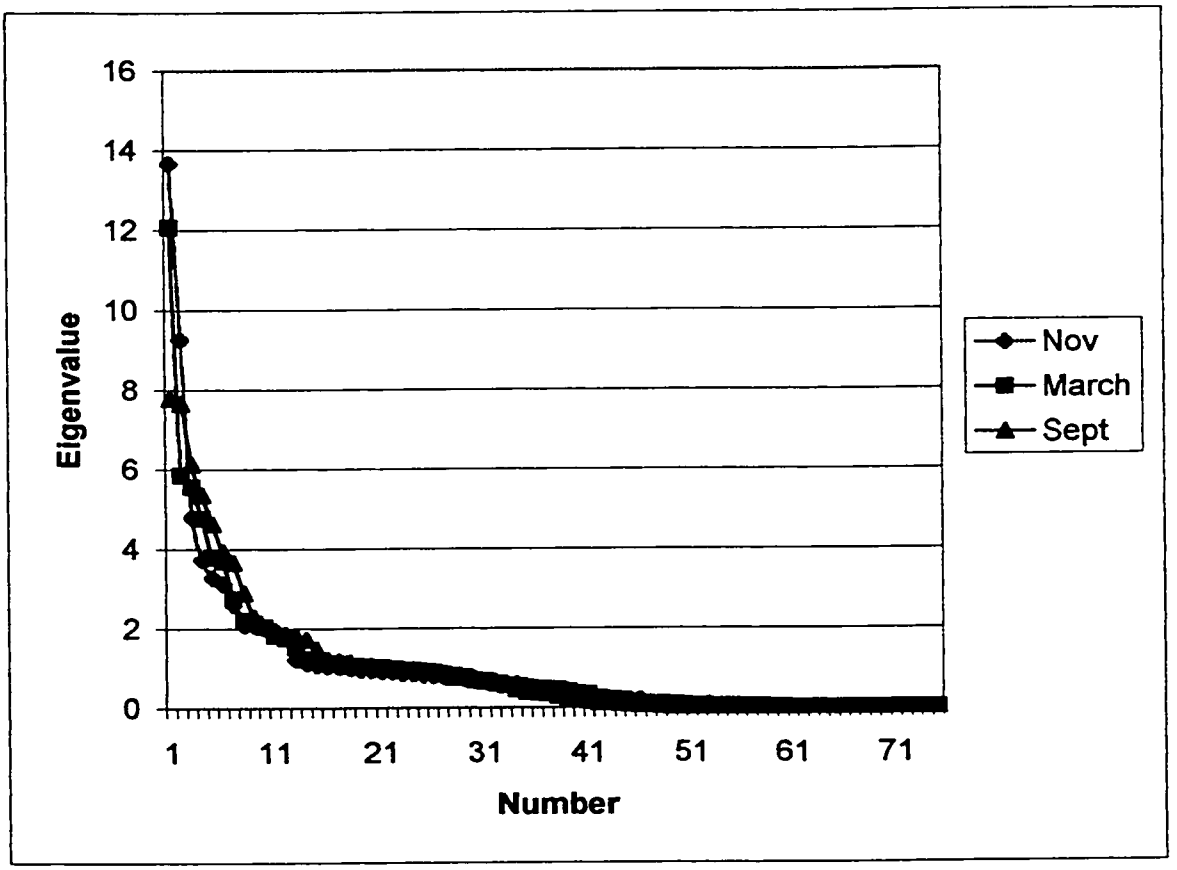


Figure 5.1: Eigenvalues for November, March and September Data, 2 PC's

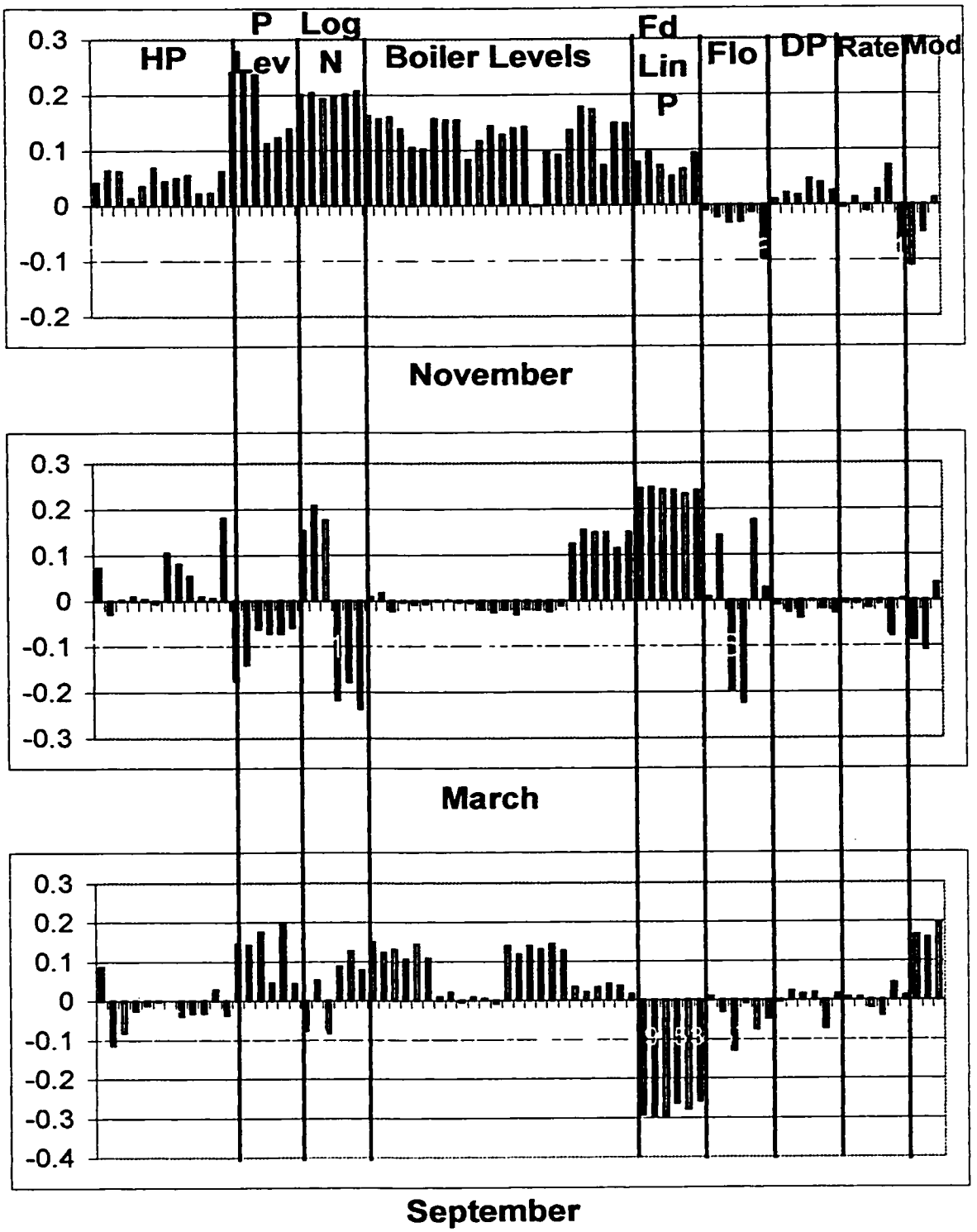


Figure 5.2: First PC Loadings for Individual Models

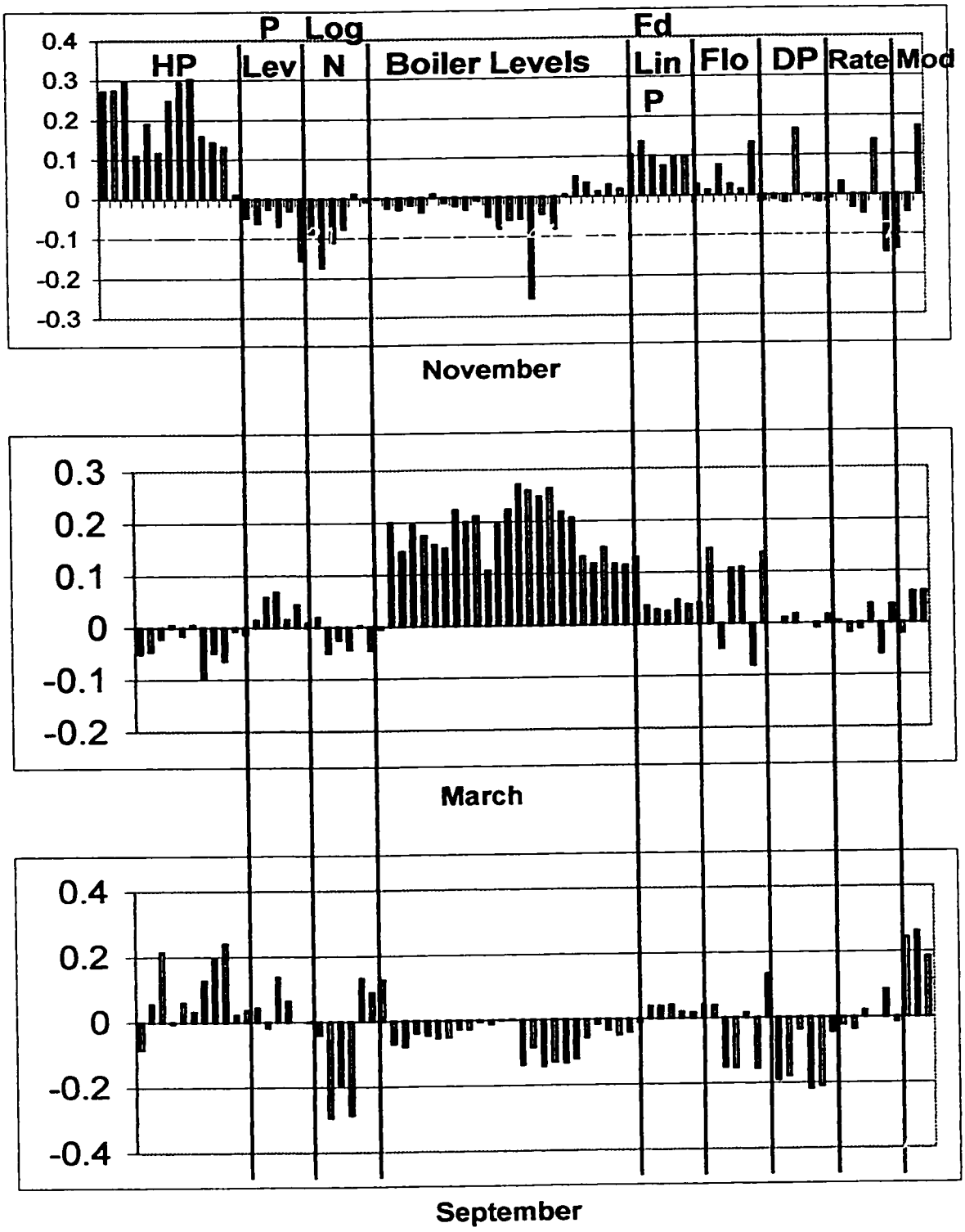


Figure 5.3: Second PC loadings for Individual Models

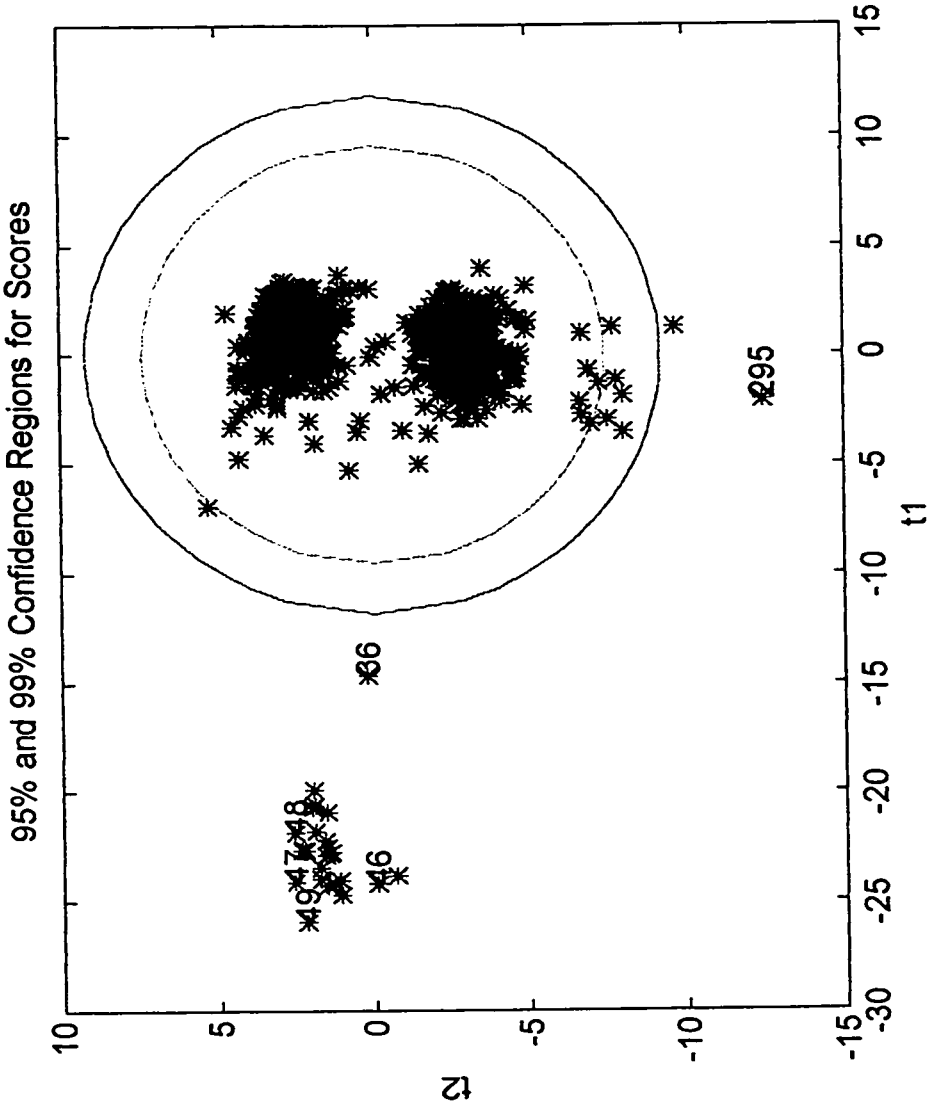


Figure 5.4: t1 vs t2 for November Data

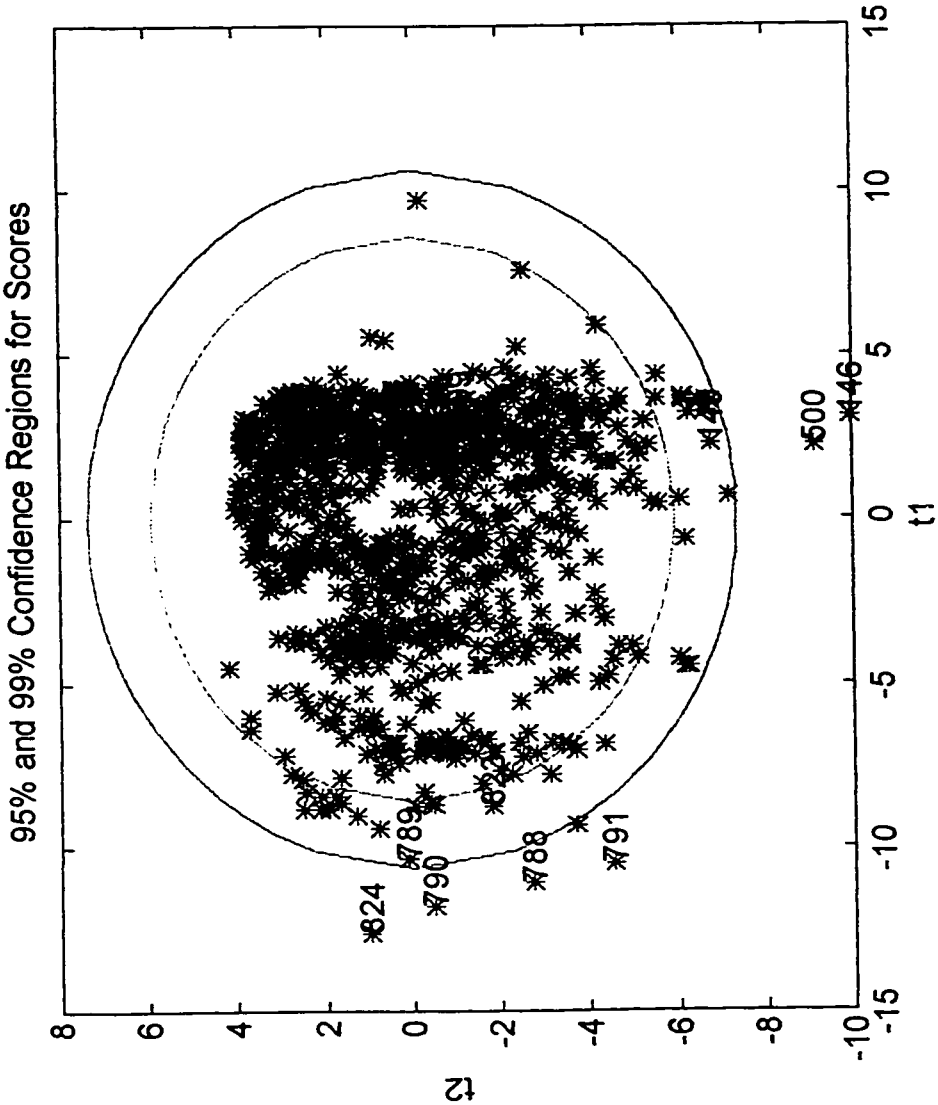


Figure 5.5: t1 vs t2 for March Data

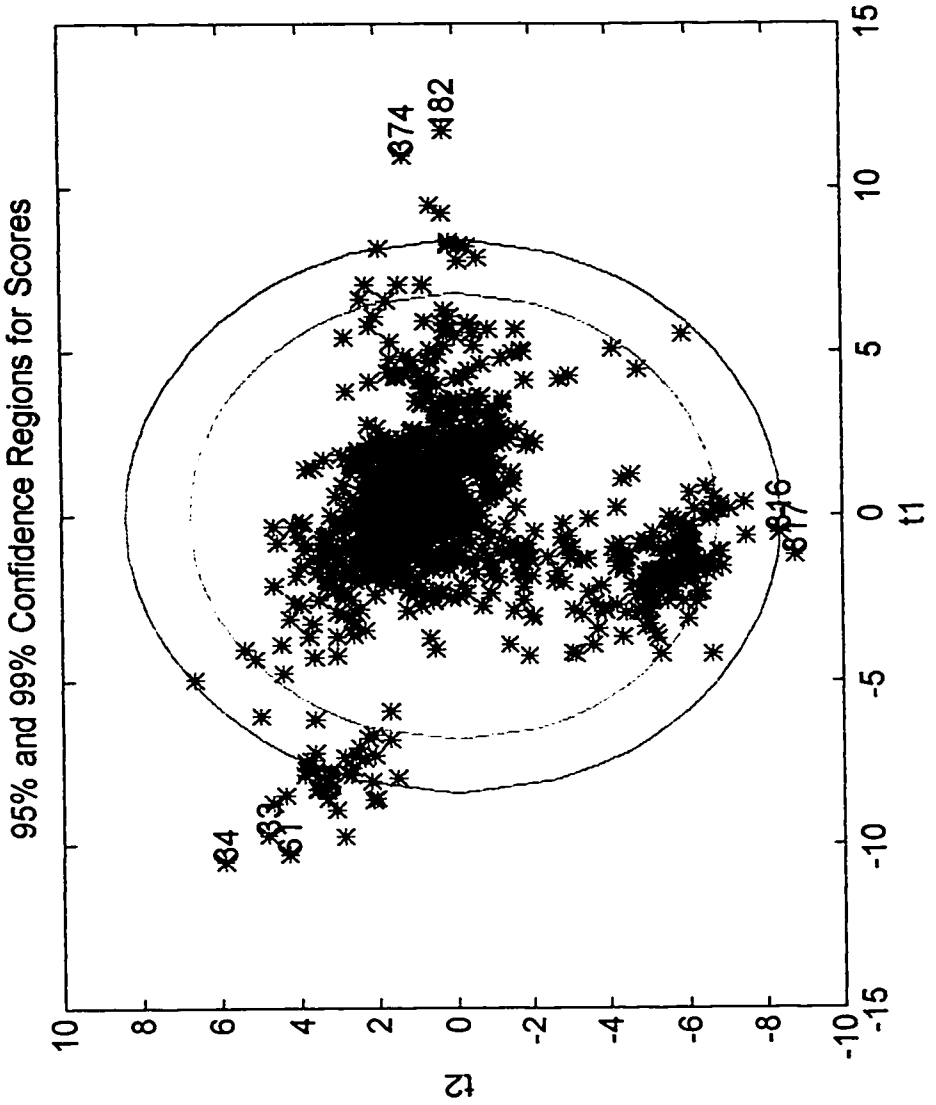


Figure 5.6: t1 vs t2 for September Data

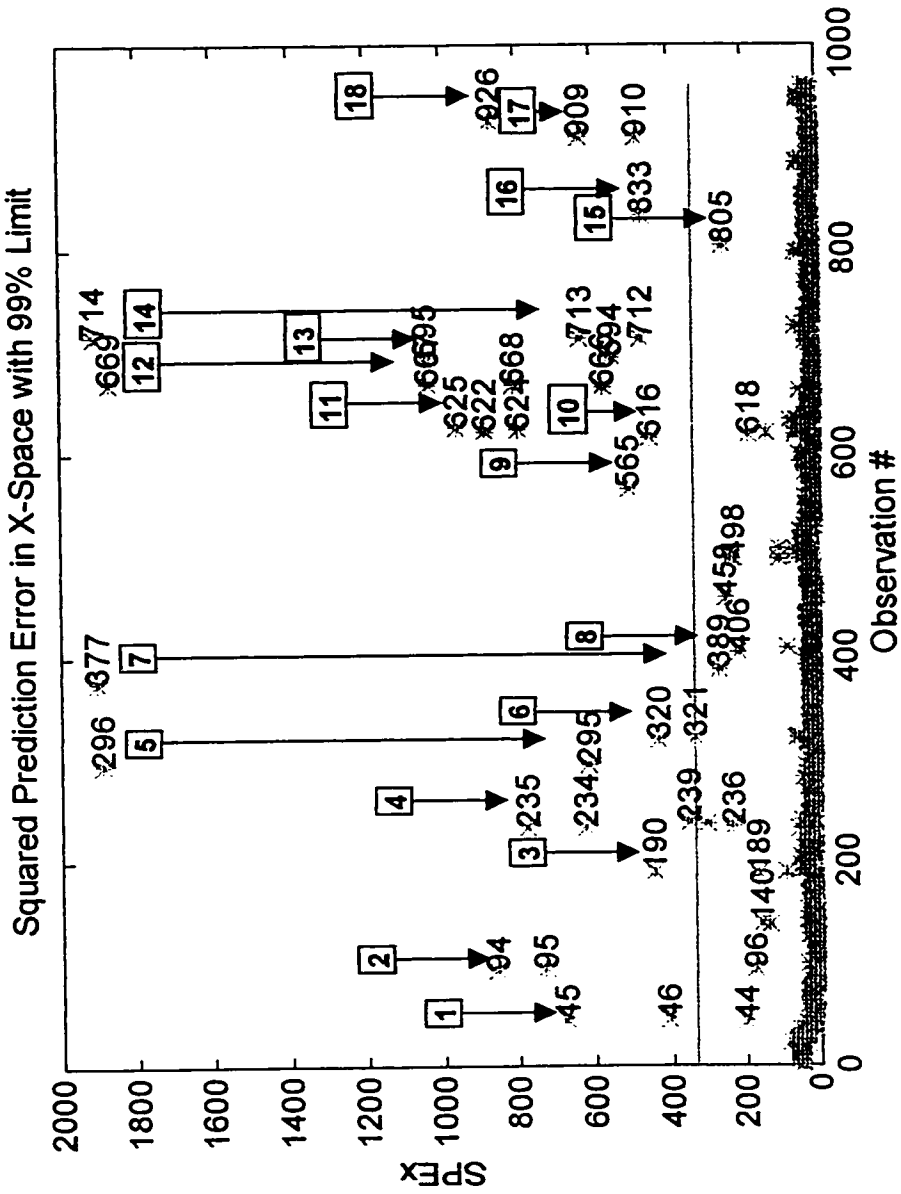


Figure 5.7: SPE for November Data

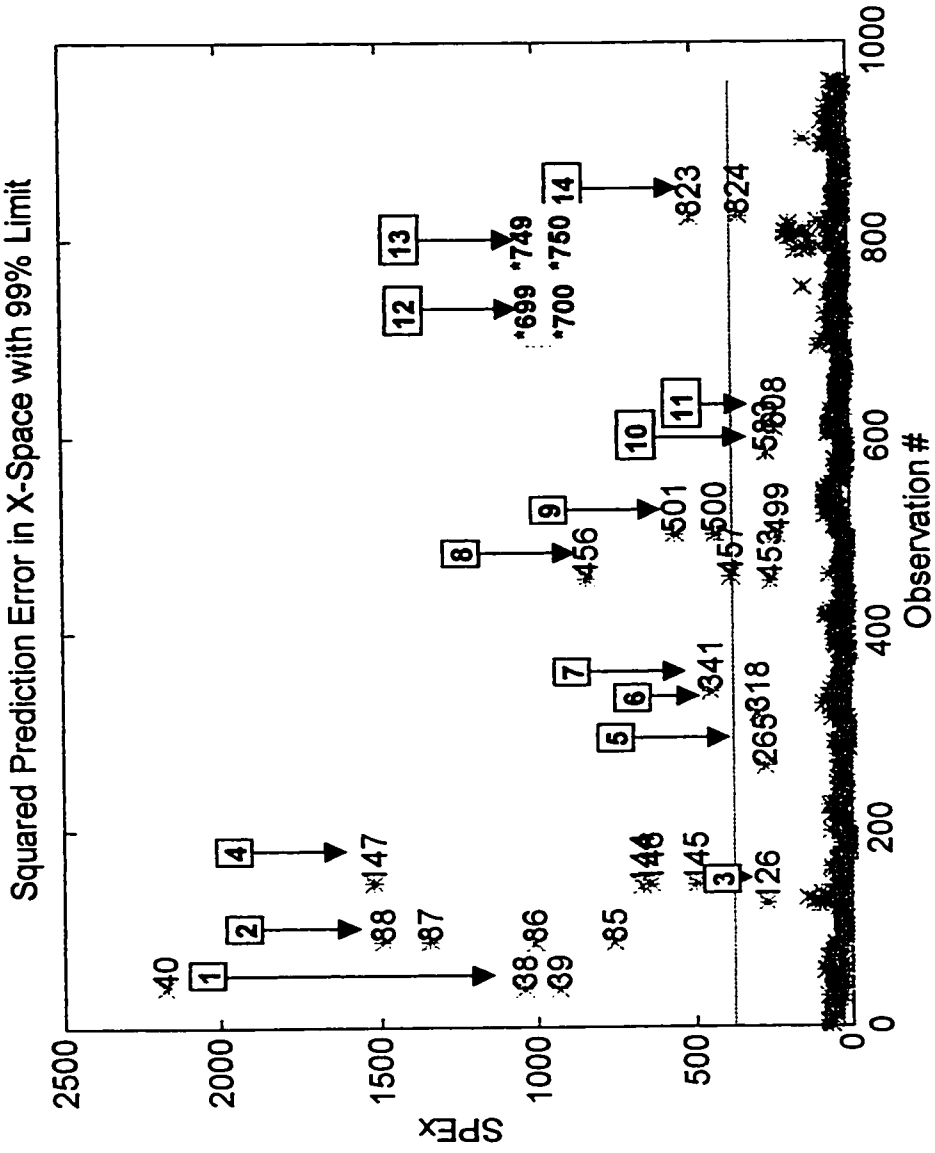


Figure 5.8: SPE for March Data

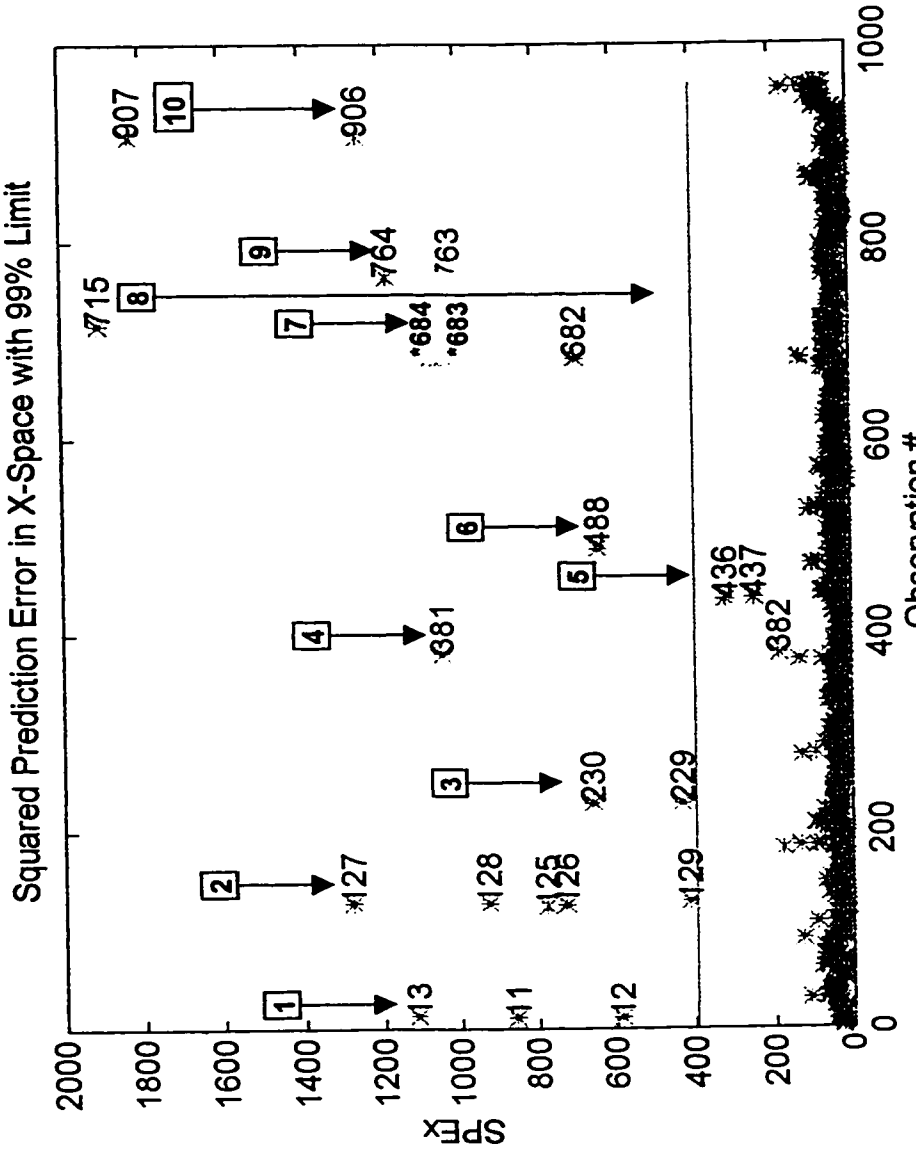


Figure 5.9: SPE for September Data

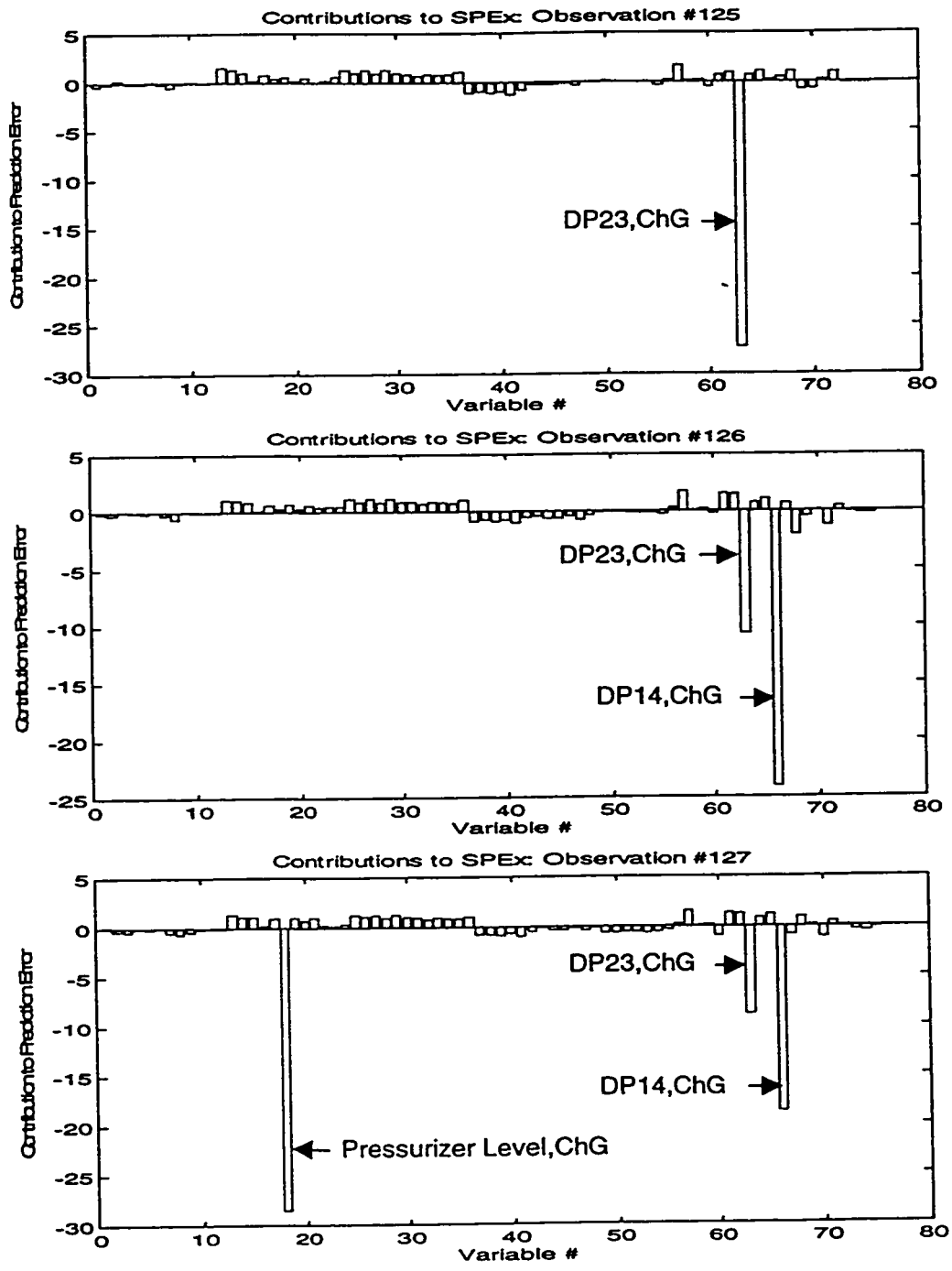


Figure 5.10: Contibutions to SPE for Observations 125 - 129

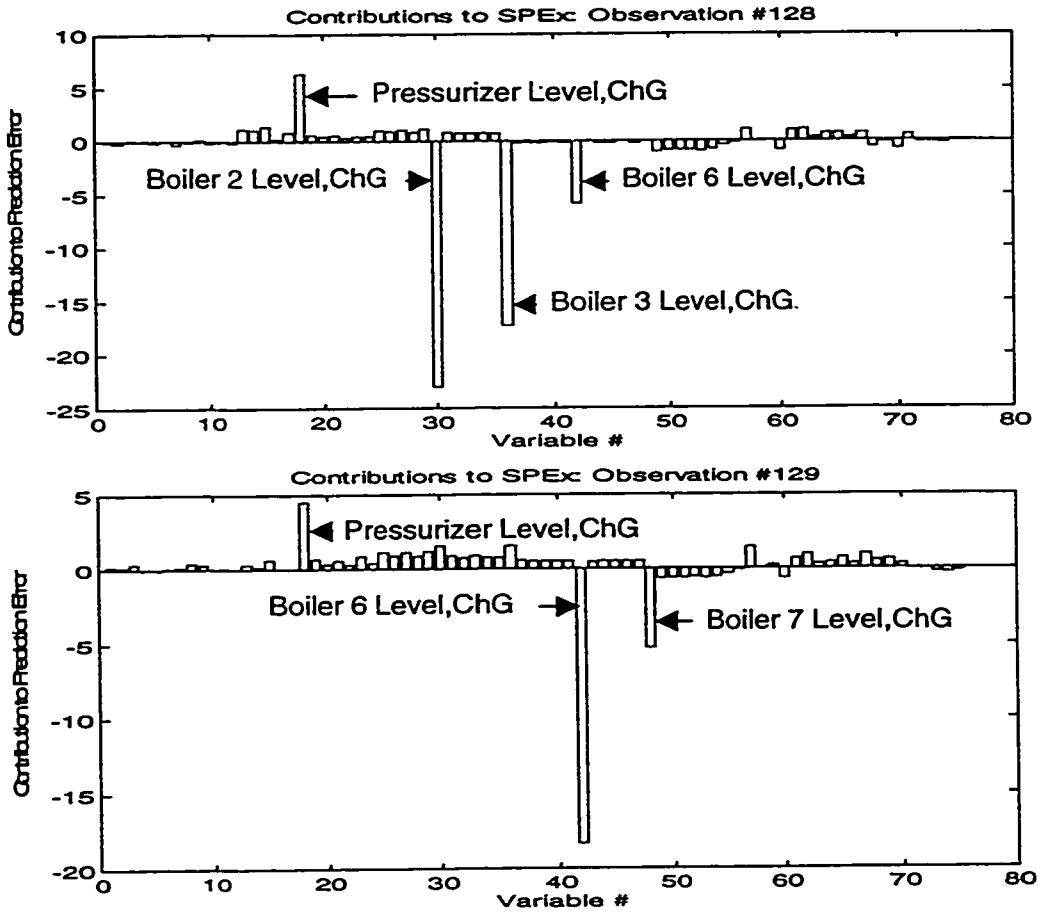


Figure 5.10 con't: Contributions to SPE for Observations 125 - 129

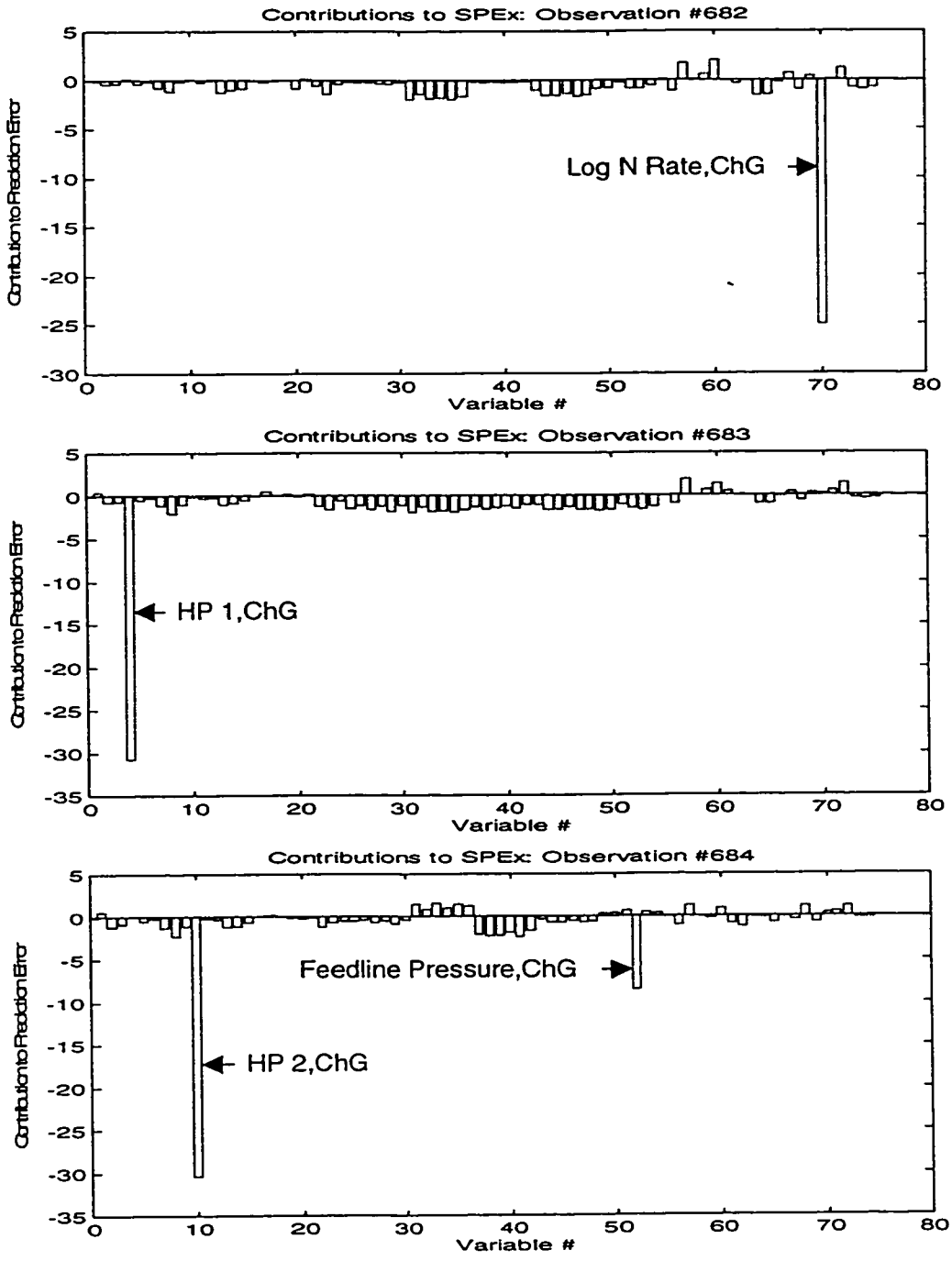


Figure 5.11: Contributions to SPE for Observations 682 - 684

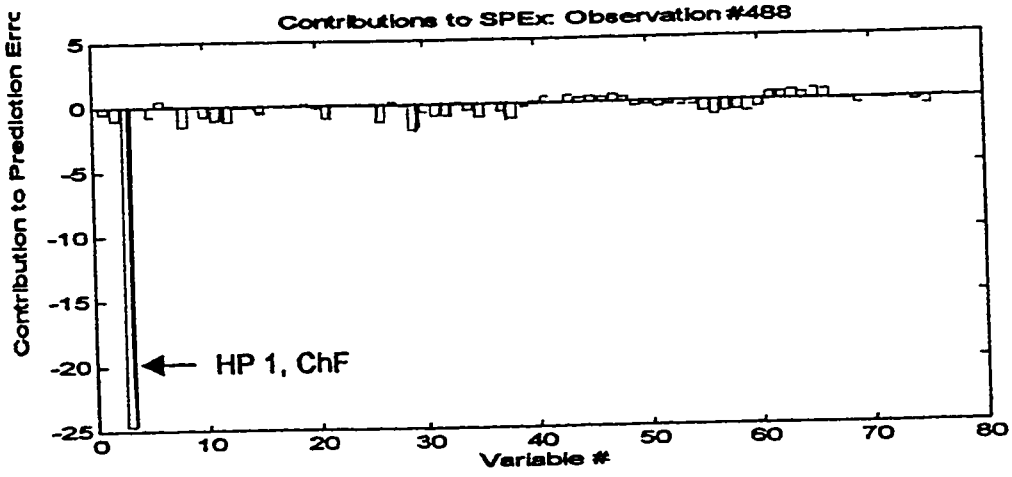


Figure 5.12: Contributions to SPE for Observation 488

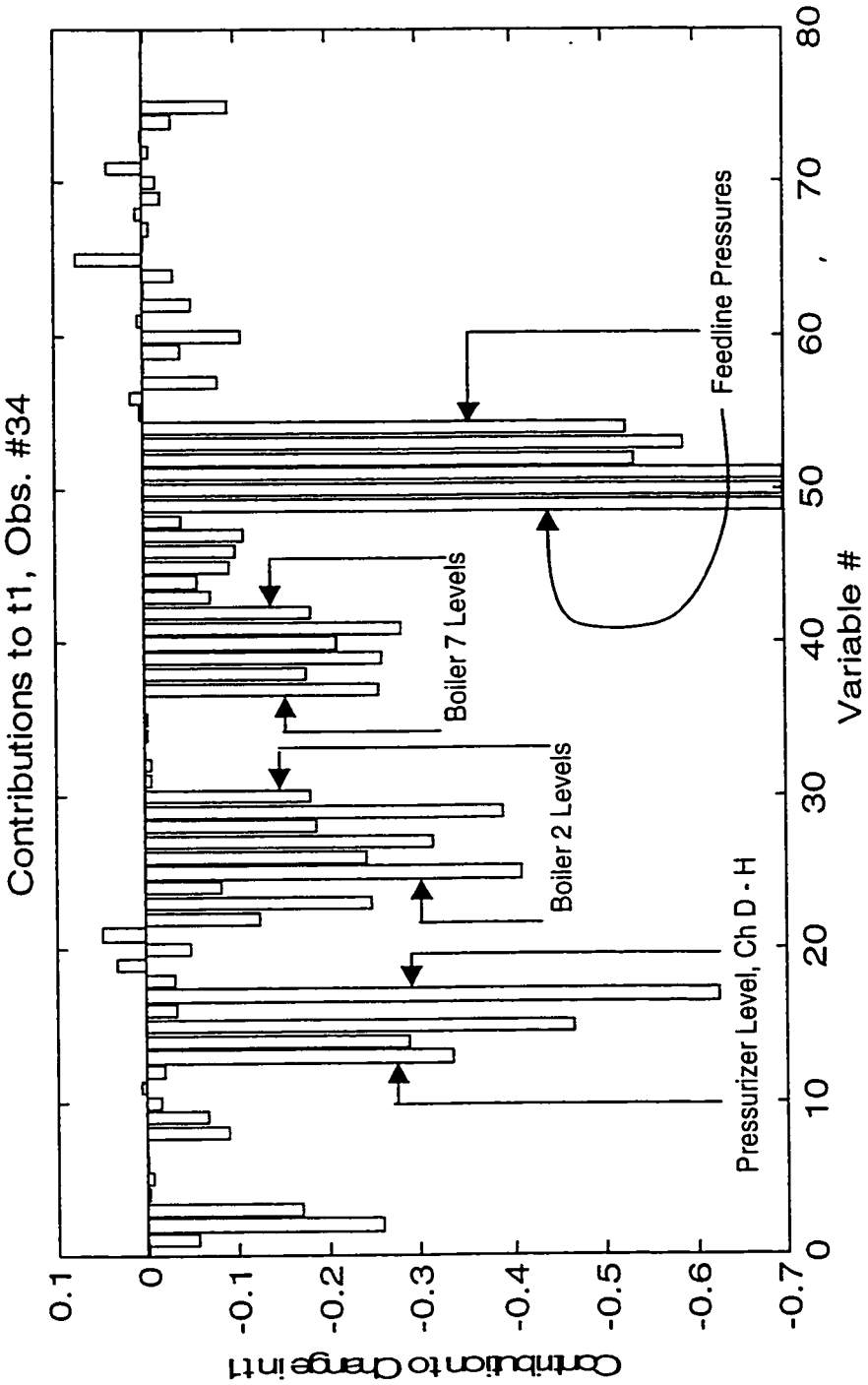


Figure 5.13: Contributions to Shift in t1 From Center Cluster to Observation 34

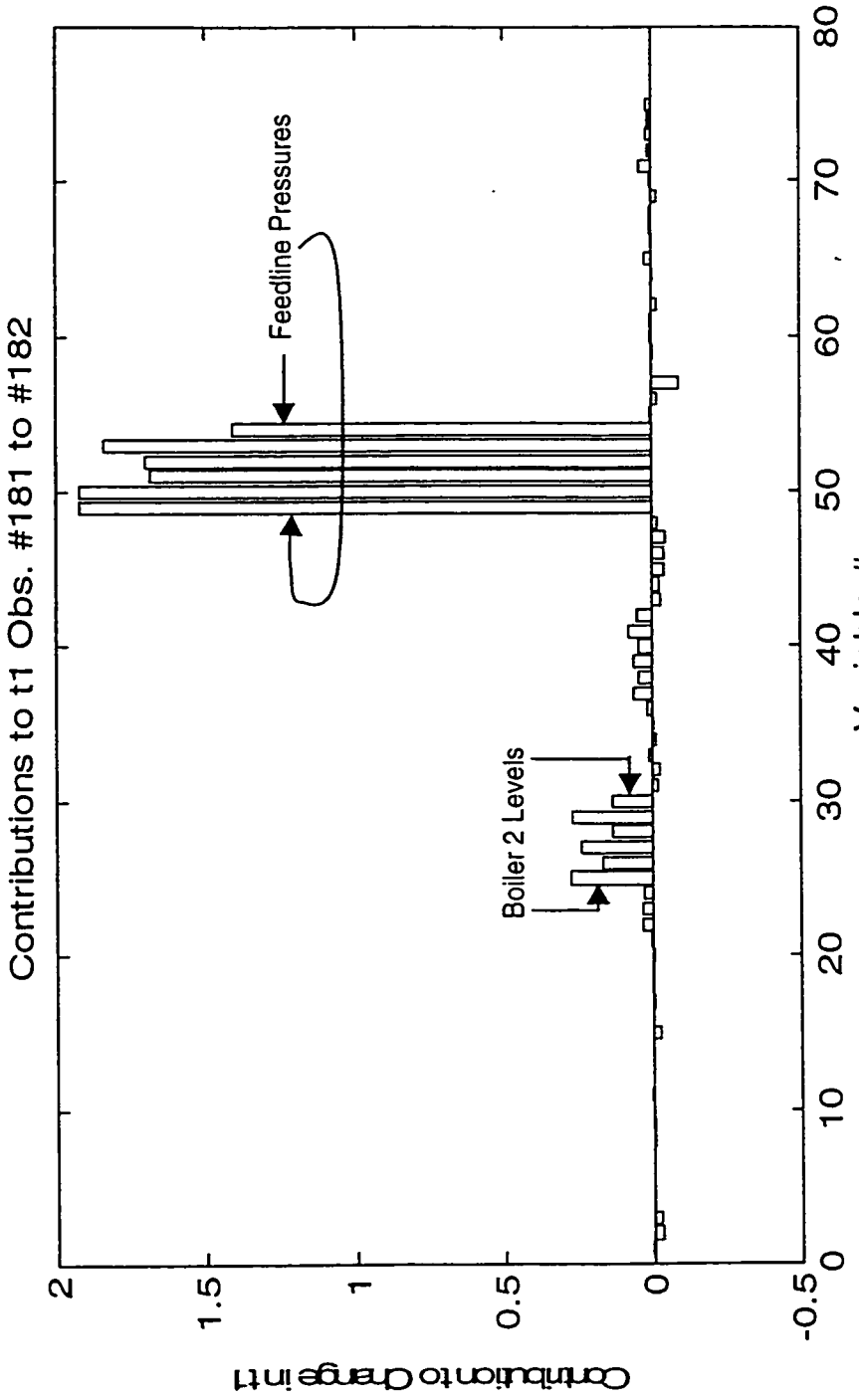


Figure 5.14: Contributions to Shift in t1 From Observations 181 to 182

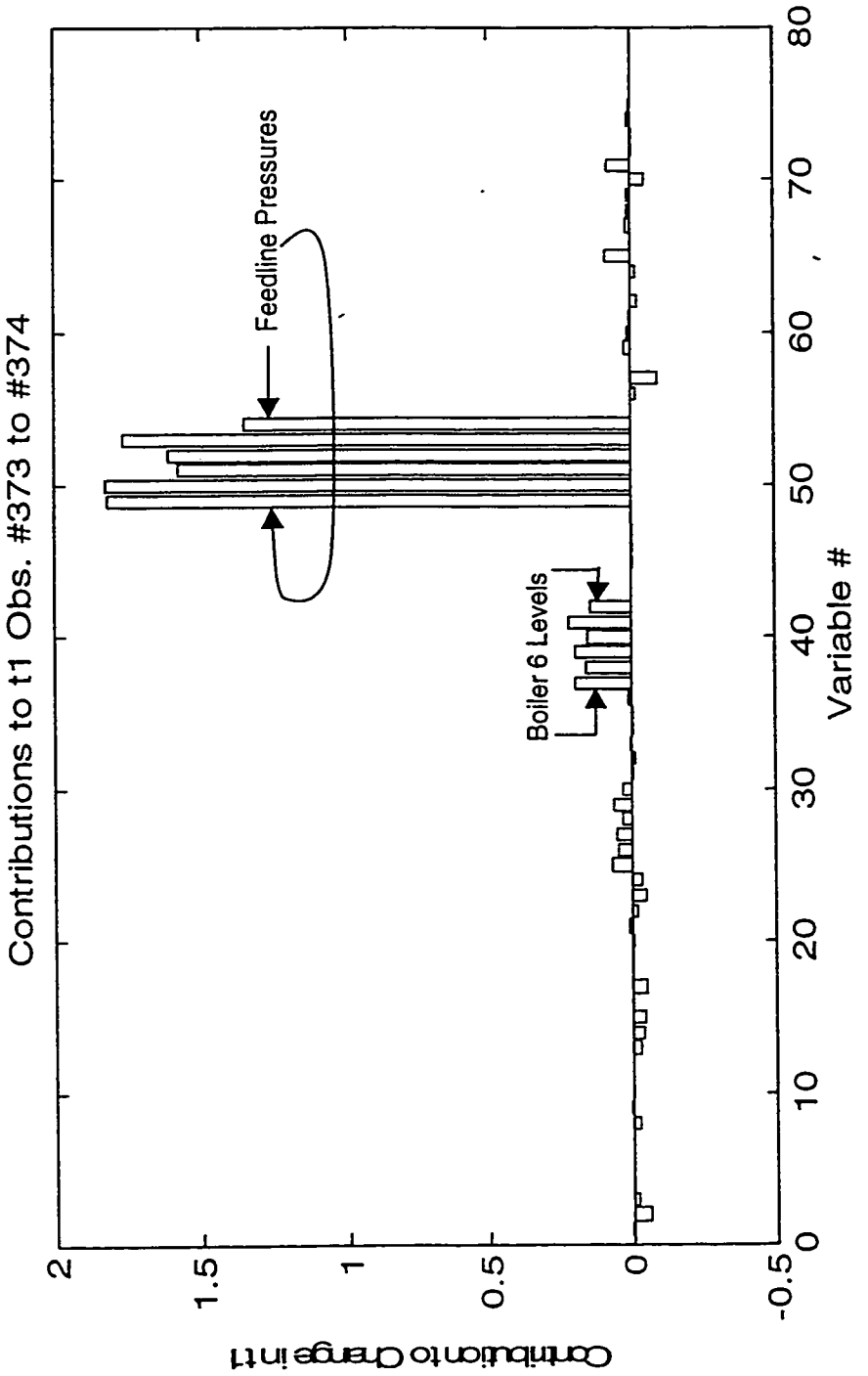


Figure 5.15: Contributions to Shift in t1 From Observations 373 to 374

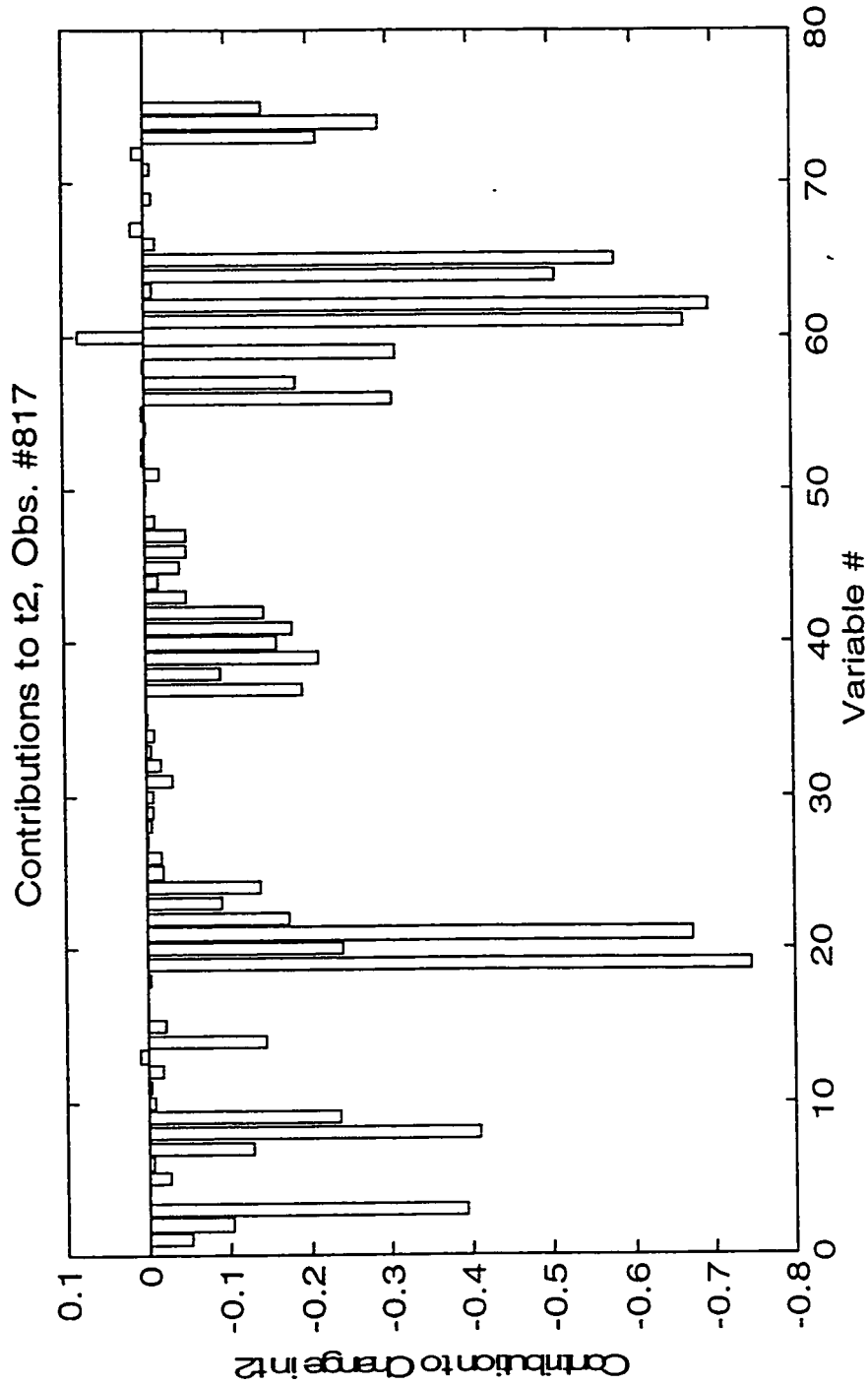


Figure 5.16: Contributions to Shift in t2 From The Center Cluster to Observation 817

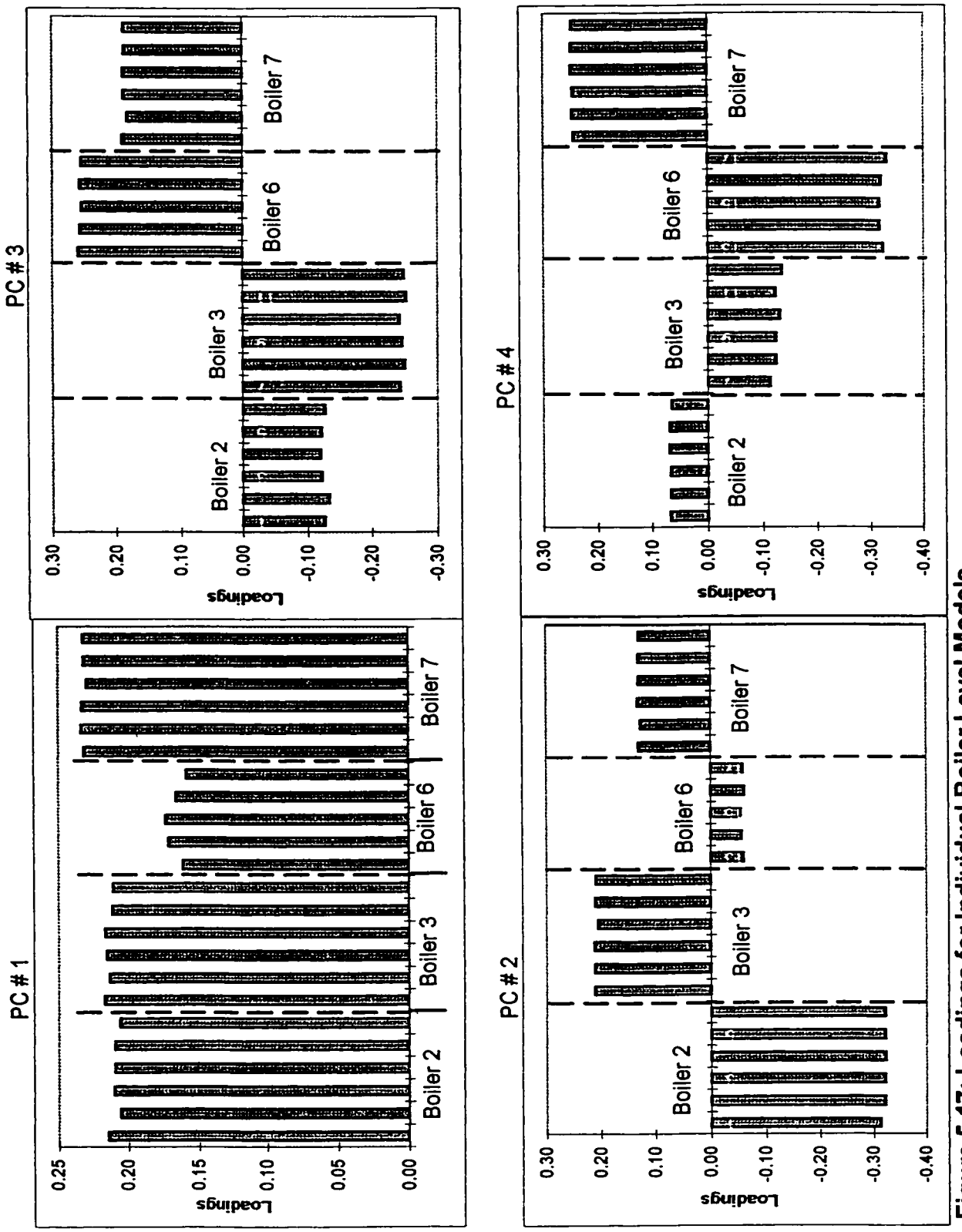


Figure 5.17: Loadings for Individual Boiler Level Models

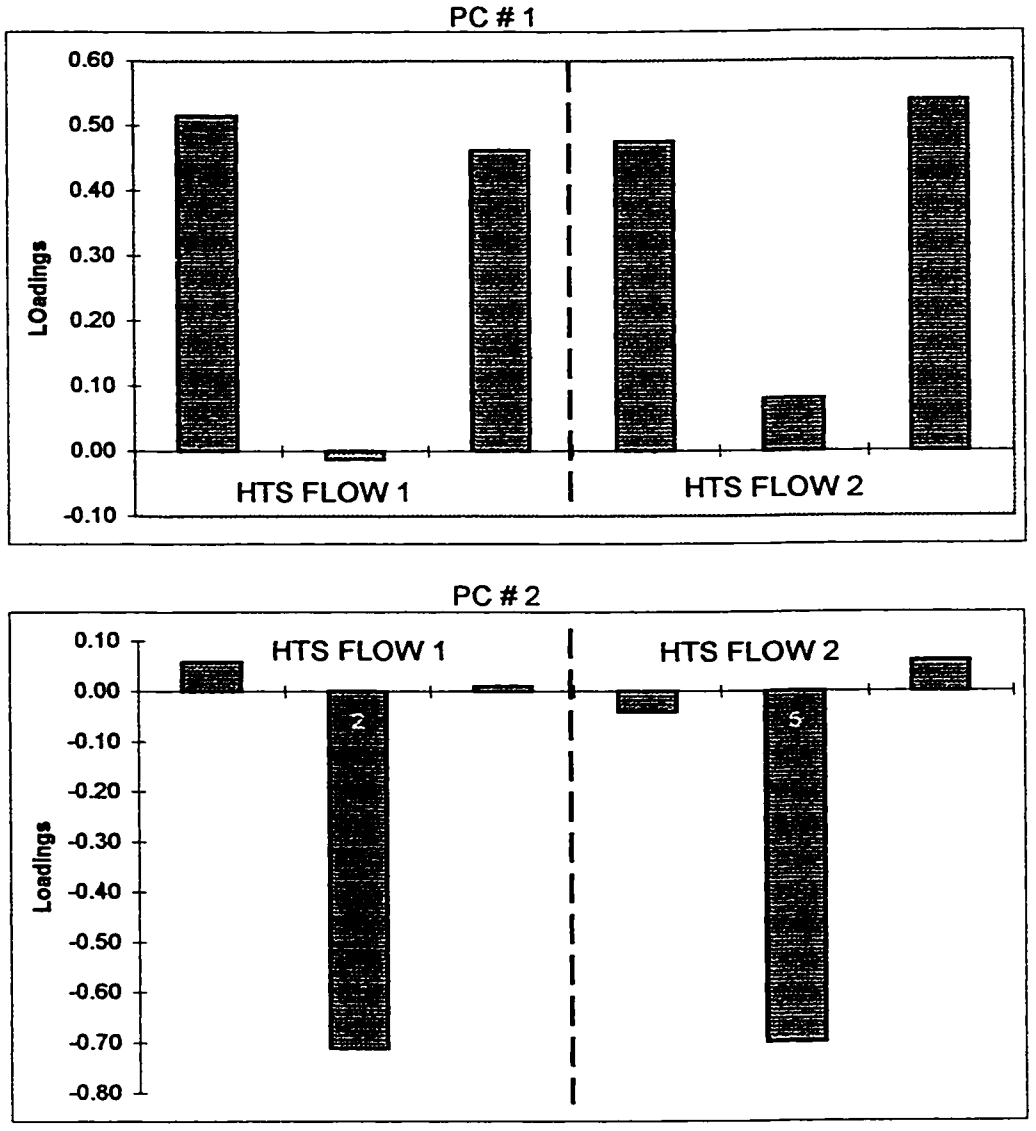


Figure 5.18: Loadings for Individual Flows

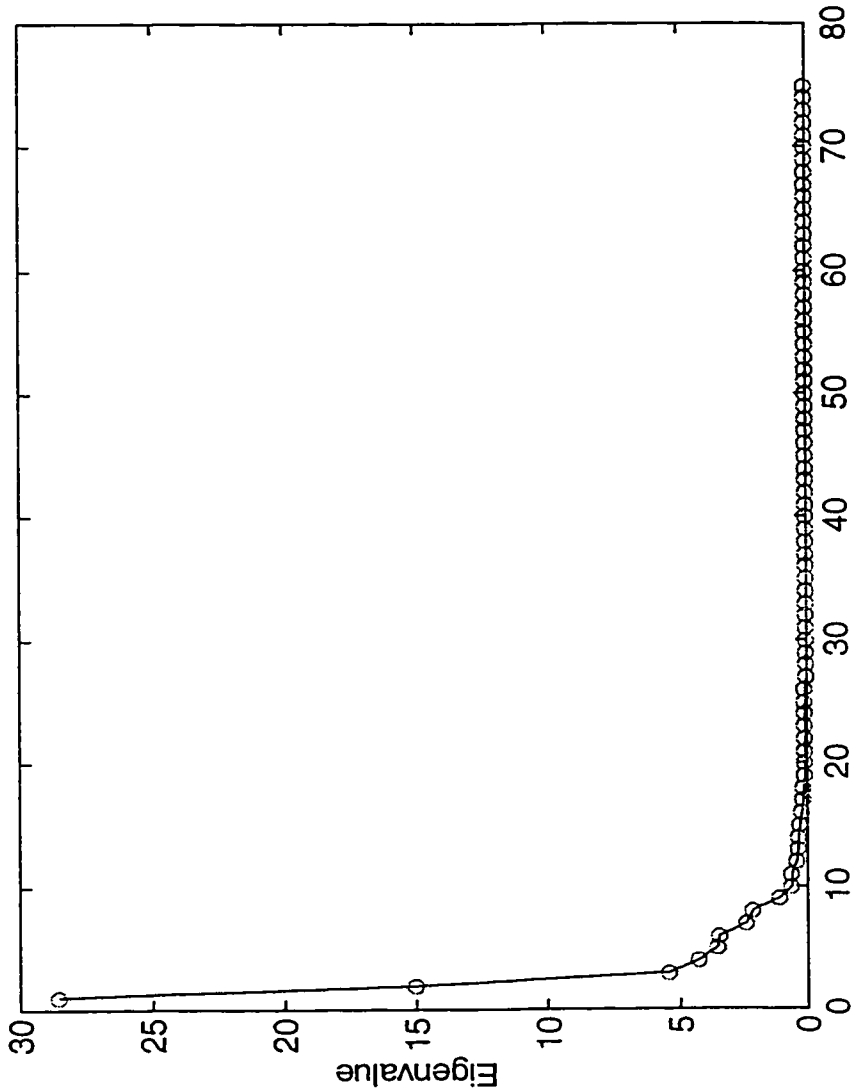


Figure 5.19: Eigenvalues for Entire Dataset

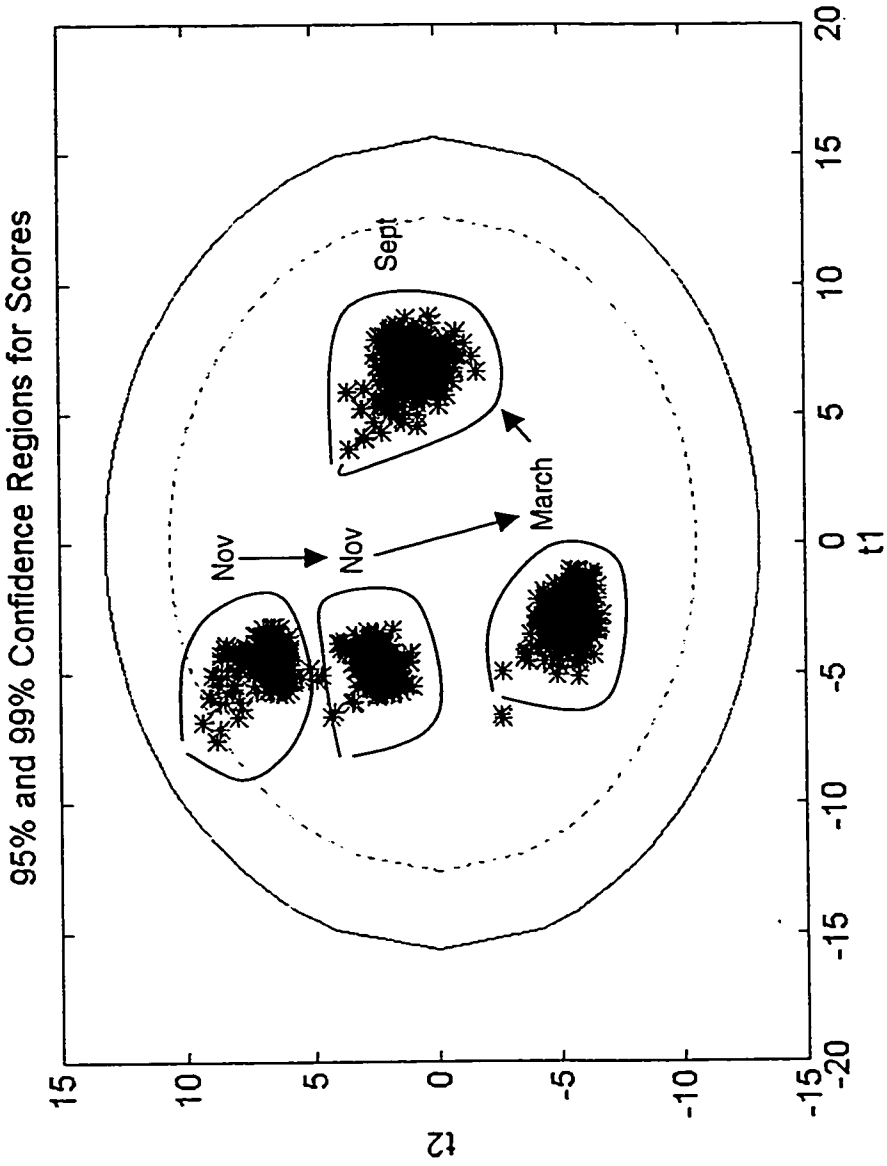


Figure 5.20: t1 vs t2 For Nov/March/Sept Data

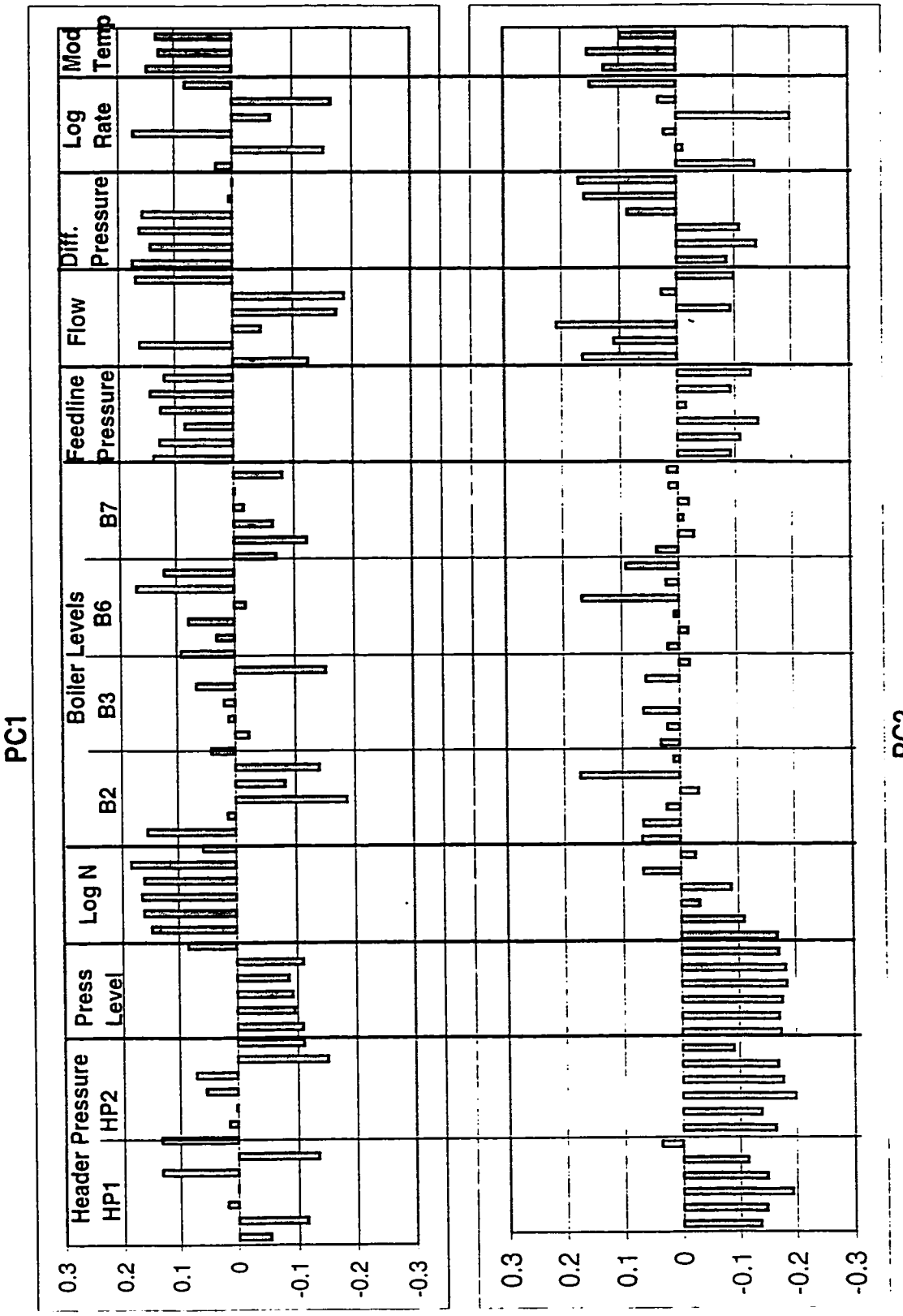


Figure 5.21: First and Second PC Loadings for Nov/March/Nov Data

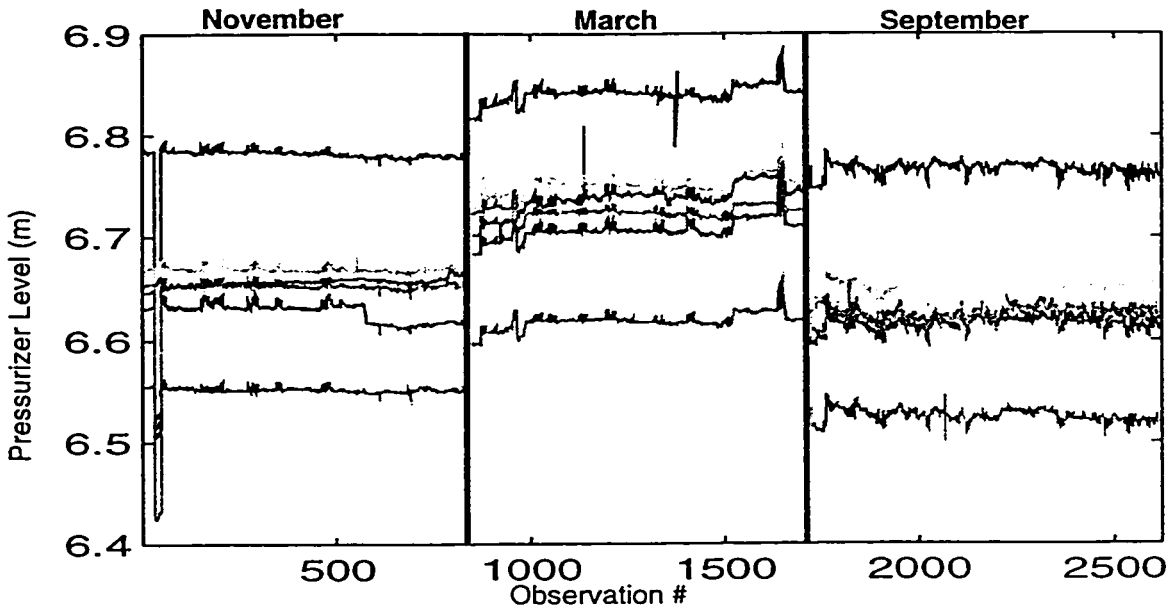


Figure 5.22: Pressurizer Level Raw Data for Nov/March/Sept

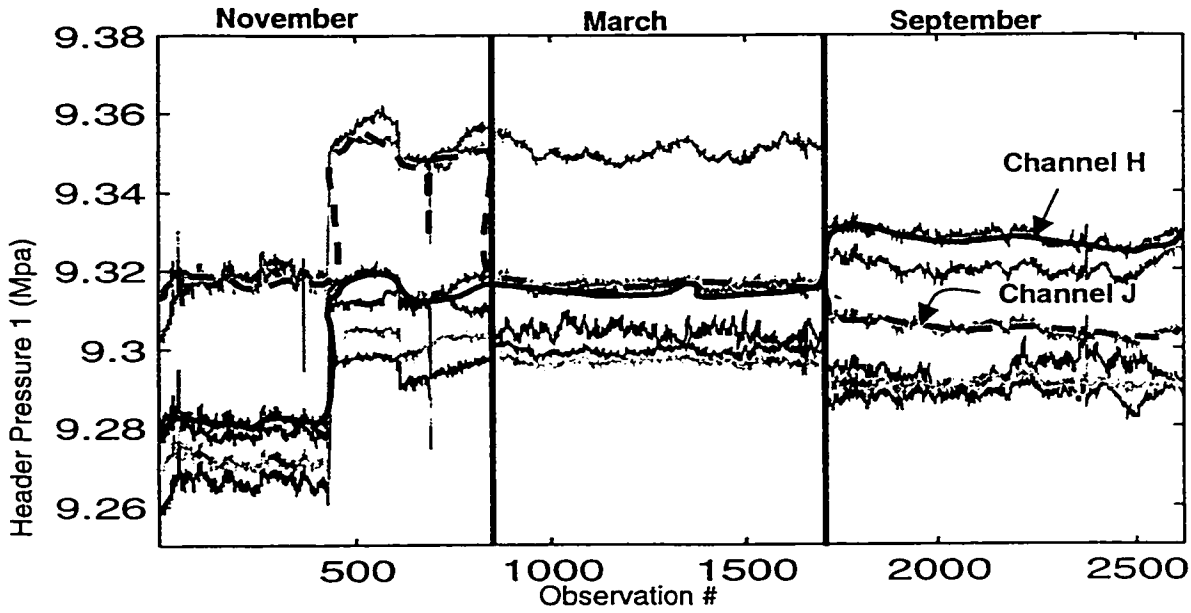


Figure 5.23: Header Pressure 1 Raw Data for Nov/March/Sept

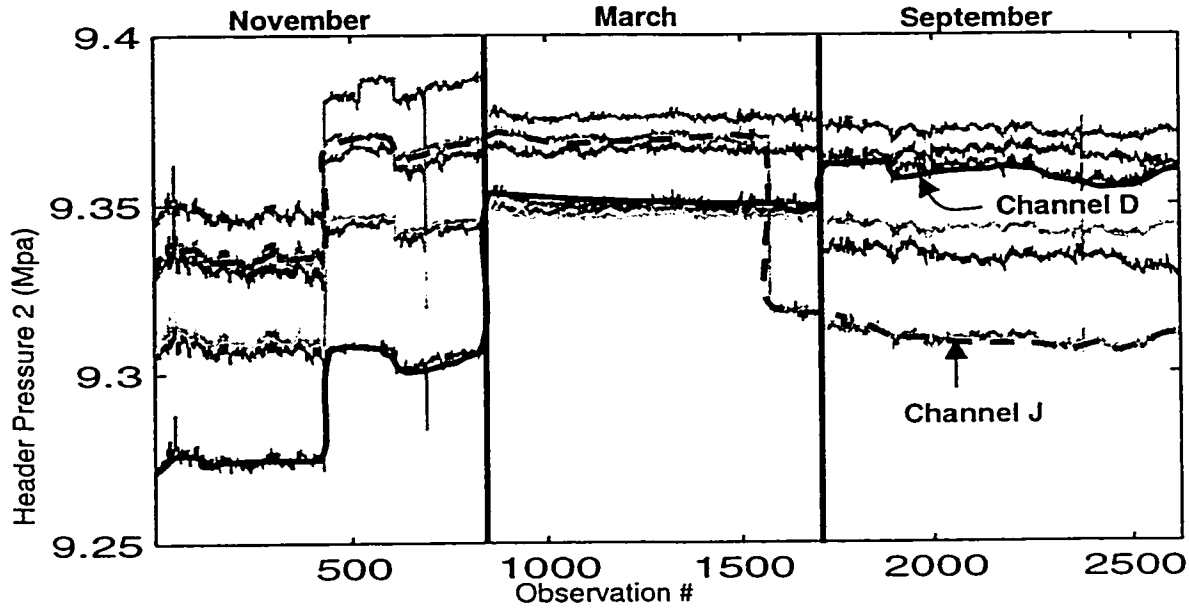


Figure 5.24: Header Pressure 2 Raw Data for Nov/March/Sept

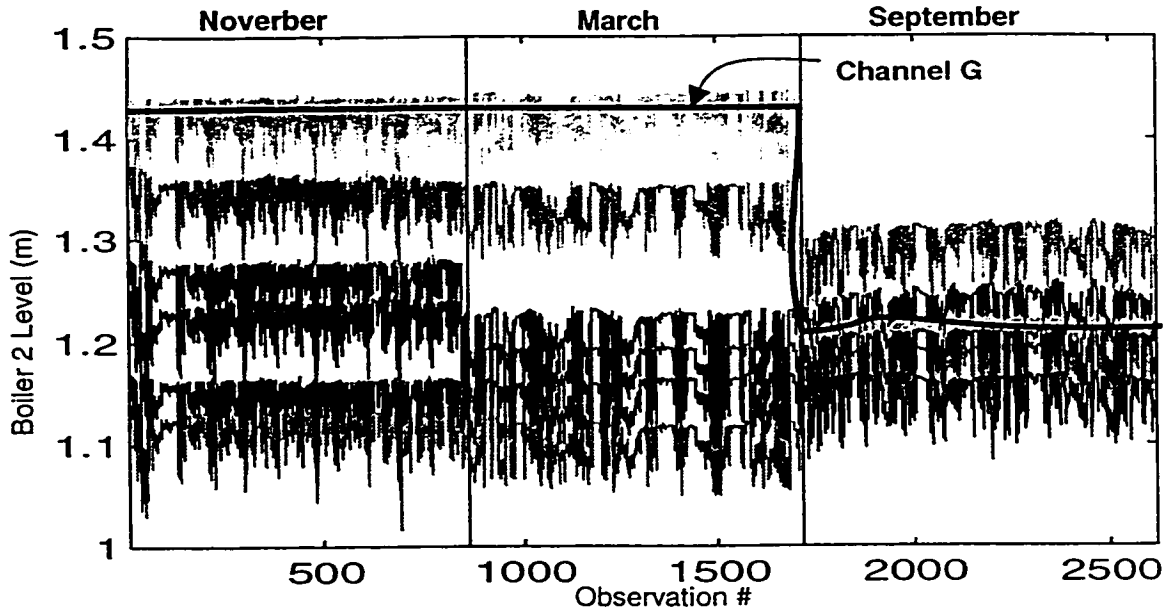


Figure 5.25: Boiler 2 Level Raw Data for Nov/March/Sept

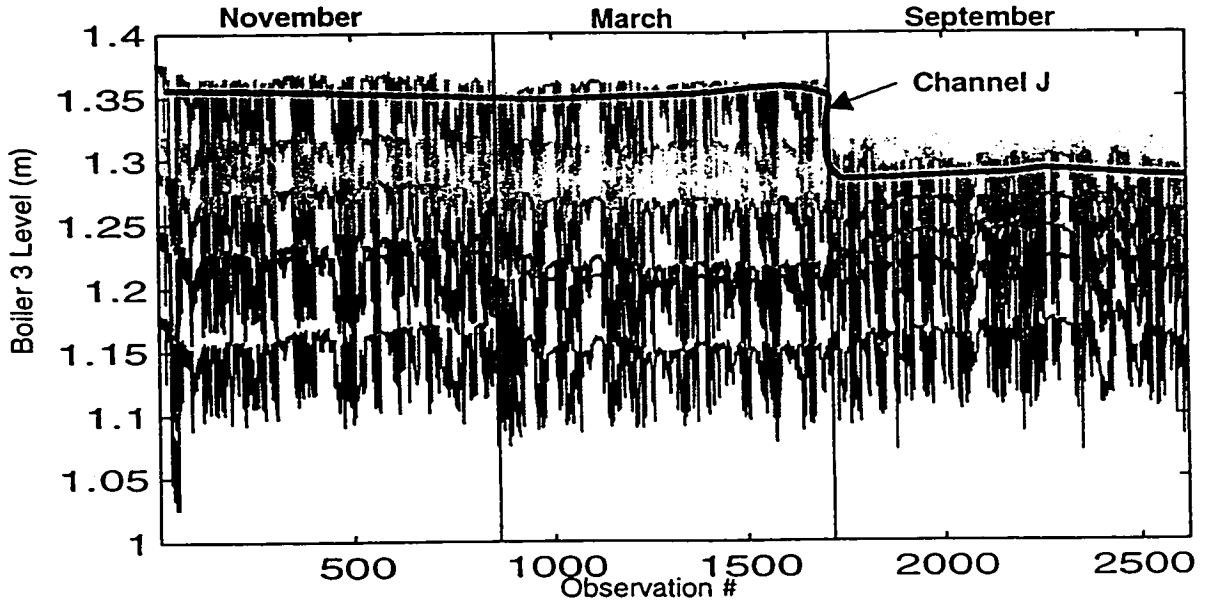


Figure 5.26: Boiler3 Level Raw Data for Nov/March/Sept

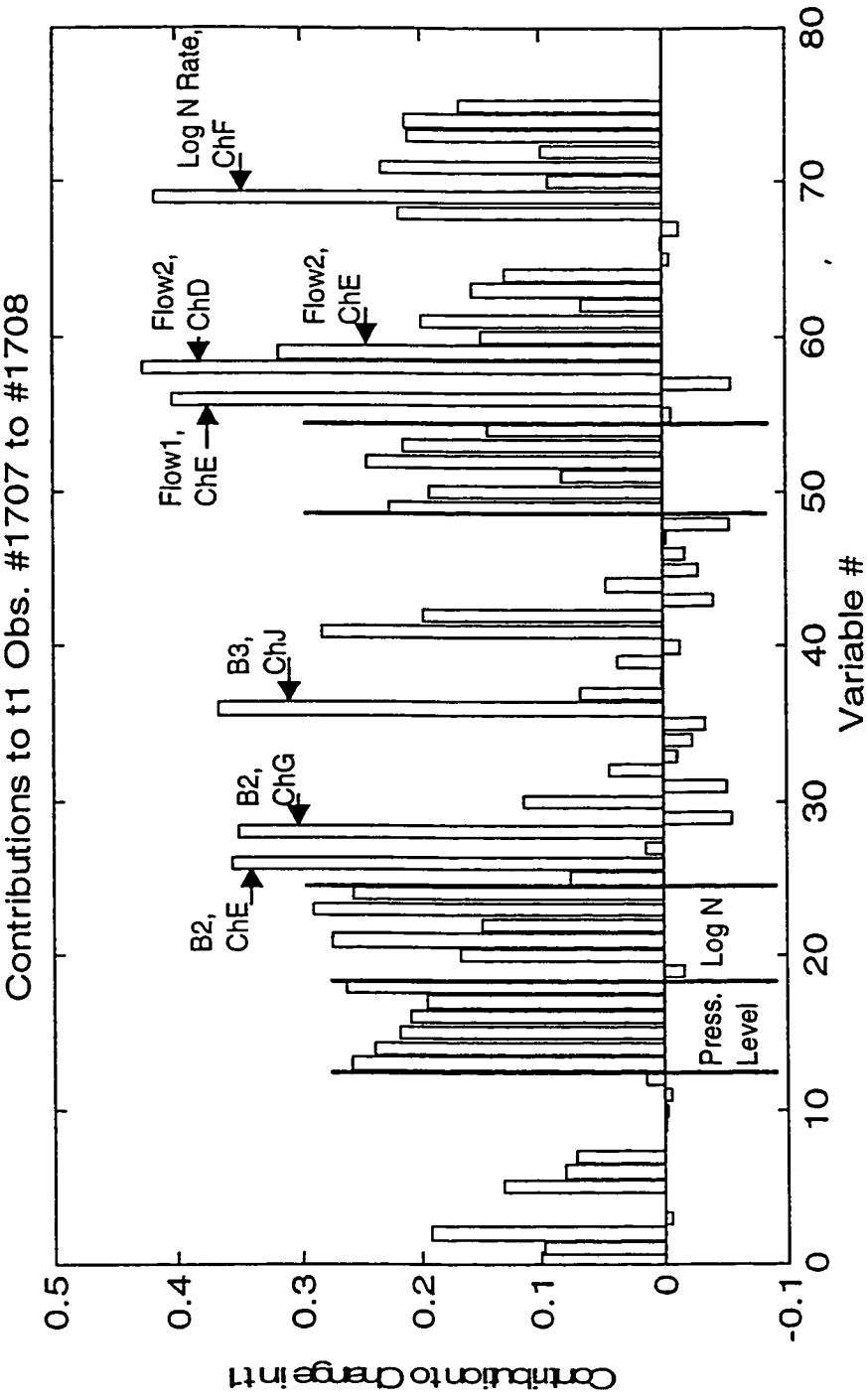


Figure 5.27: Contributions to Shift in t1 From March Data to Sept. Data

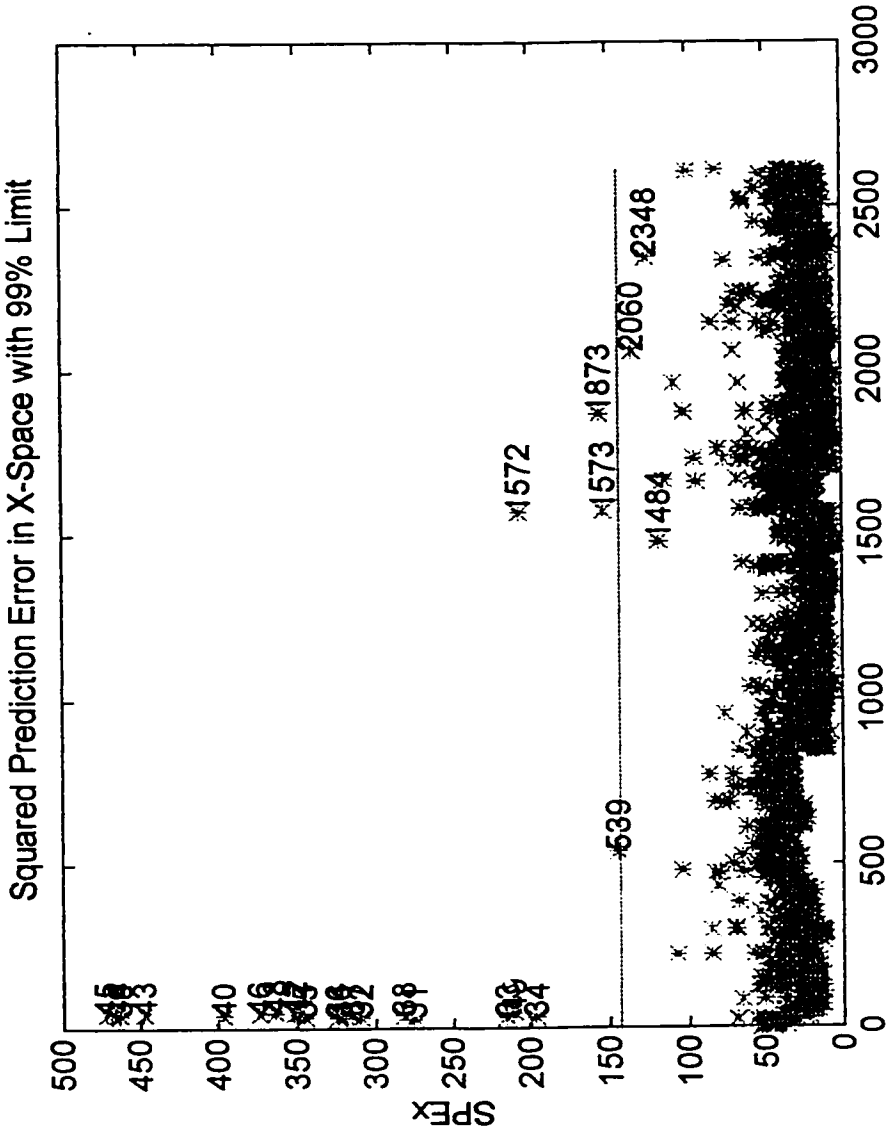


Figure 5.28: SPE for Nov/March/Sept Data

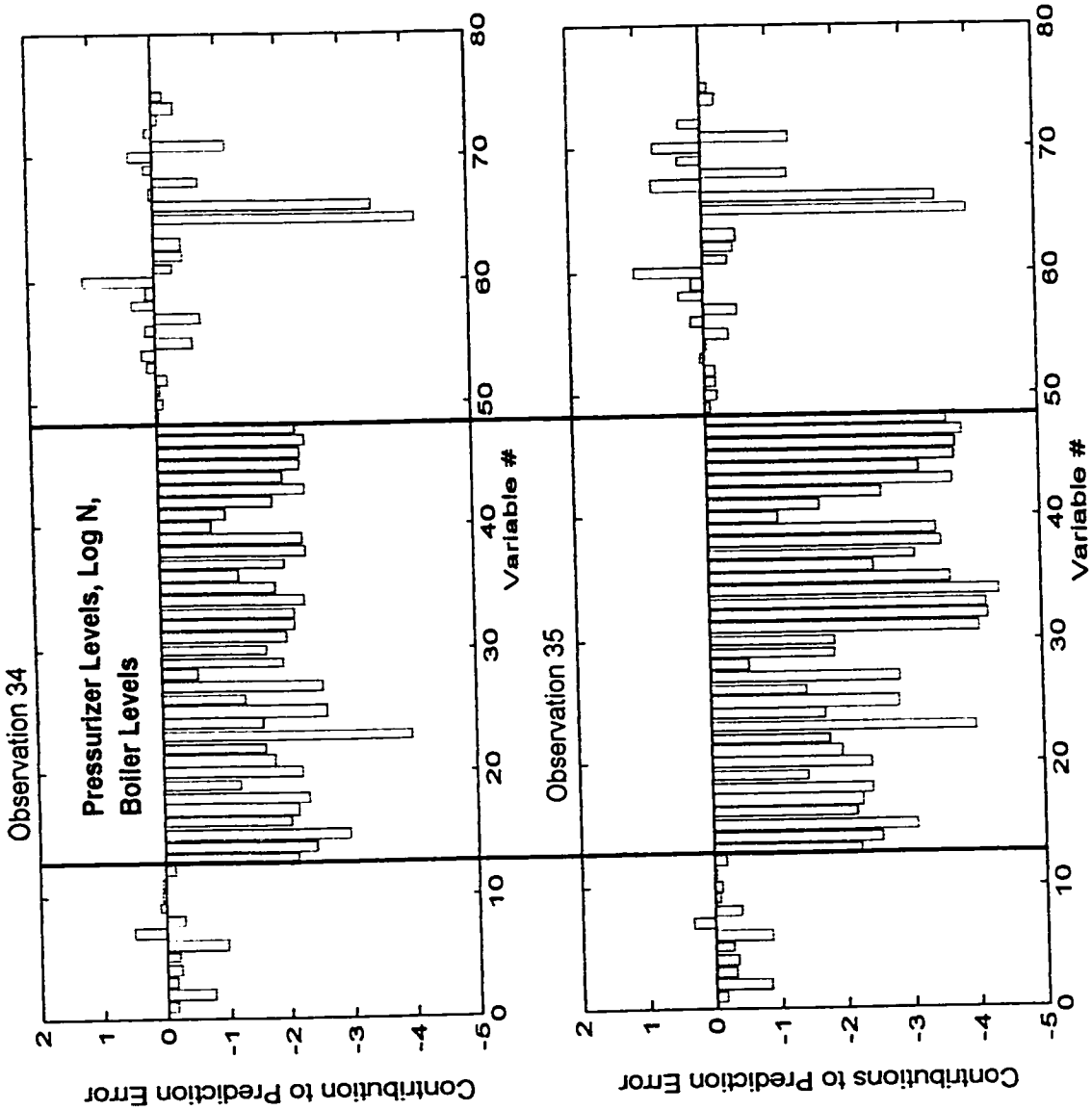


Figure 5.29: Contribition to SPE for Observations 34-35

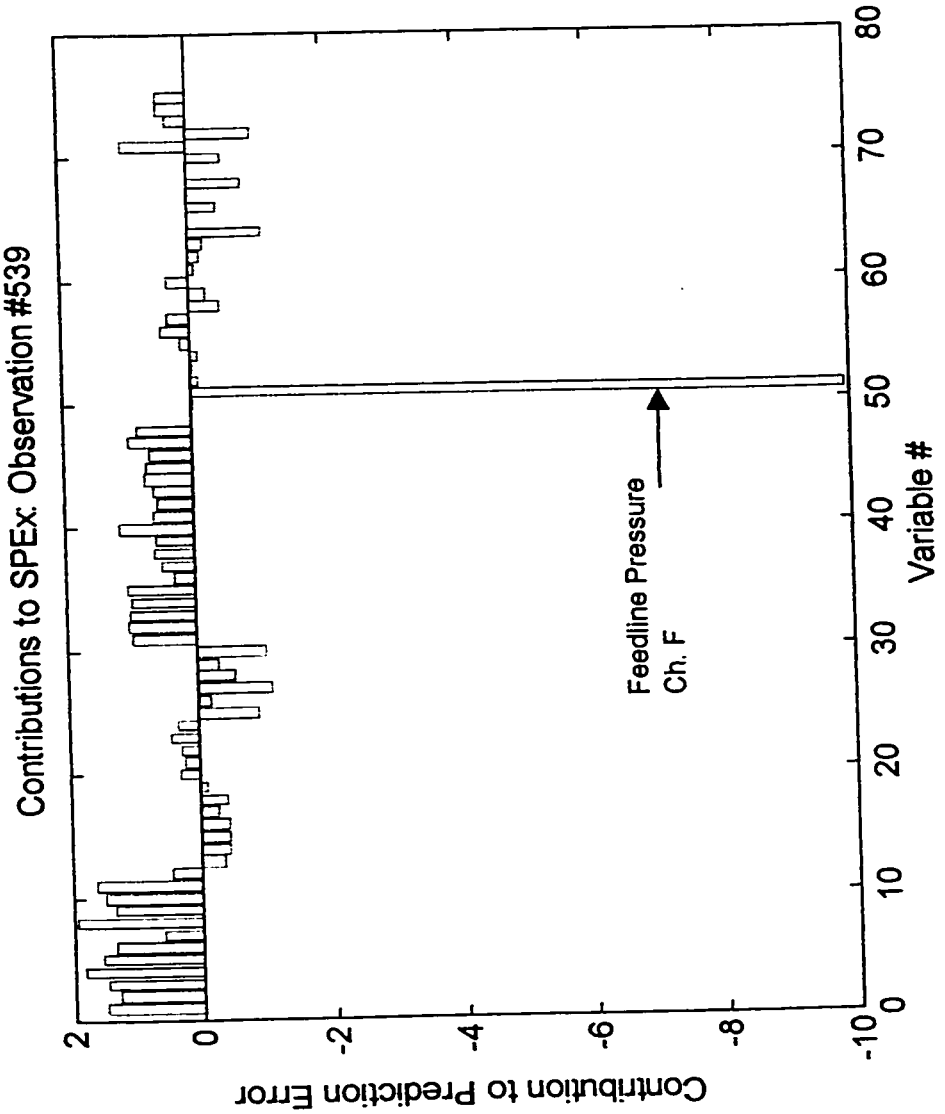


Figure 5.30: Contributions to SPE for Observation 539

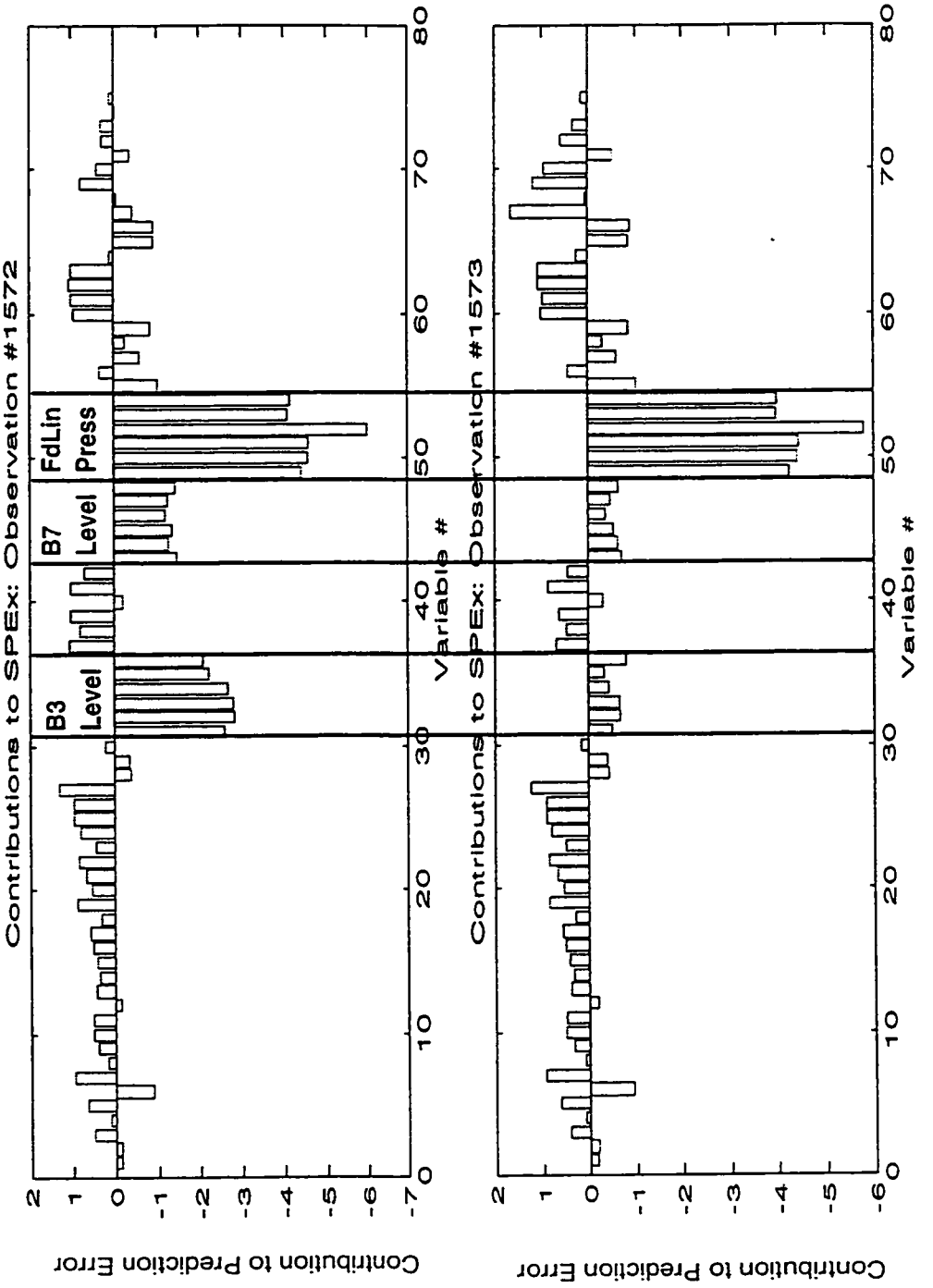


Figure 5.31: Contributions to SPE for Observations 1572 - 1573

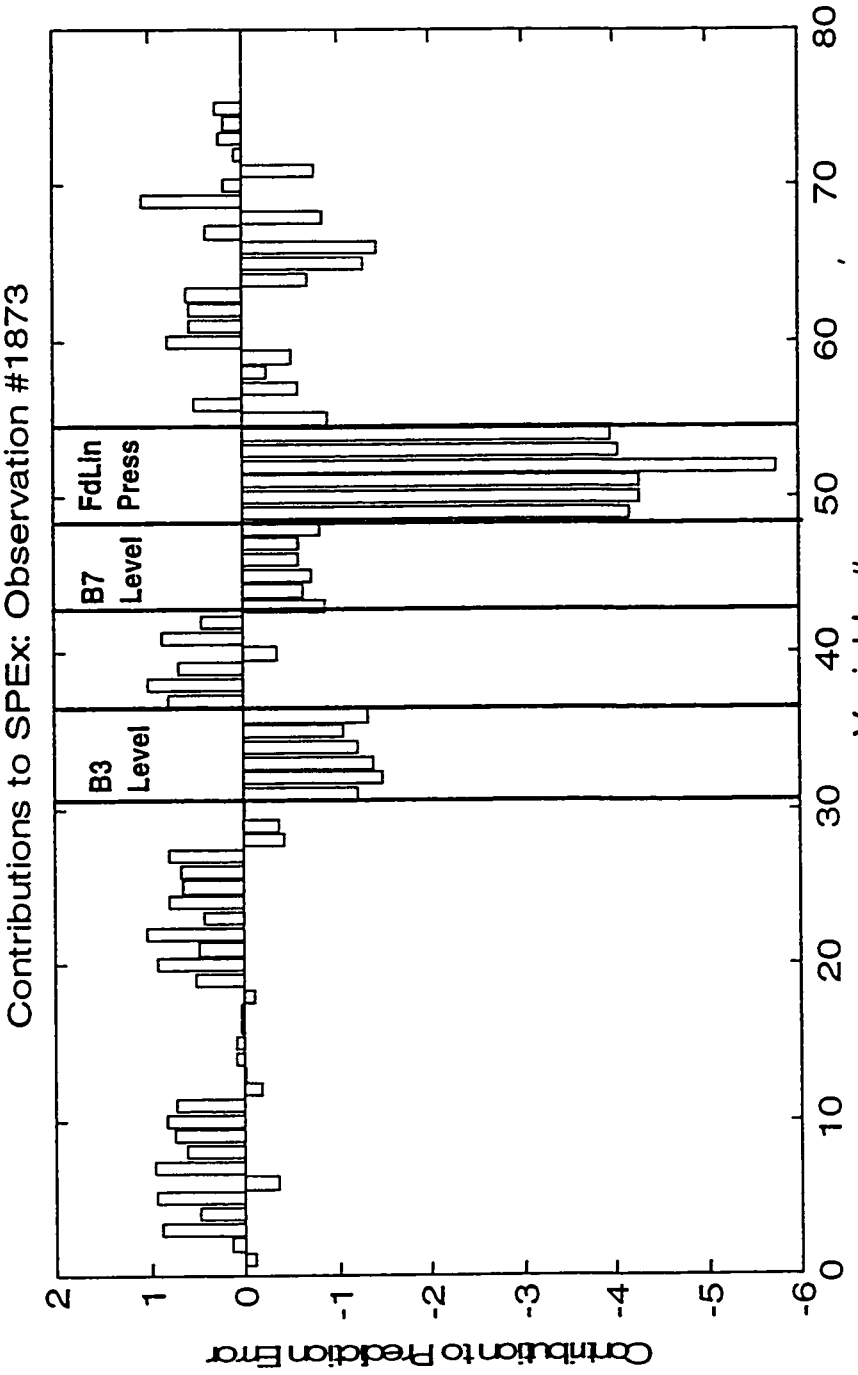


Figure 5.32: Contributions to SPE for Observation 1873

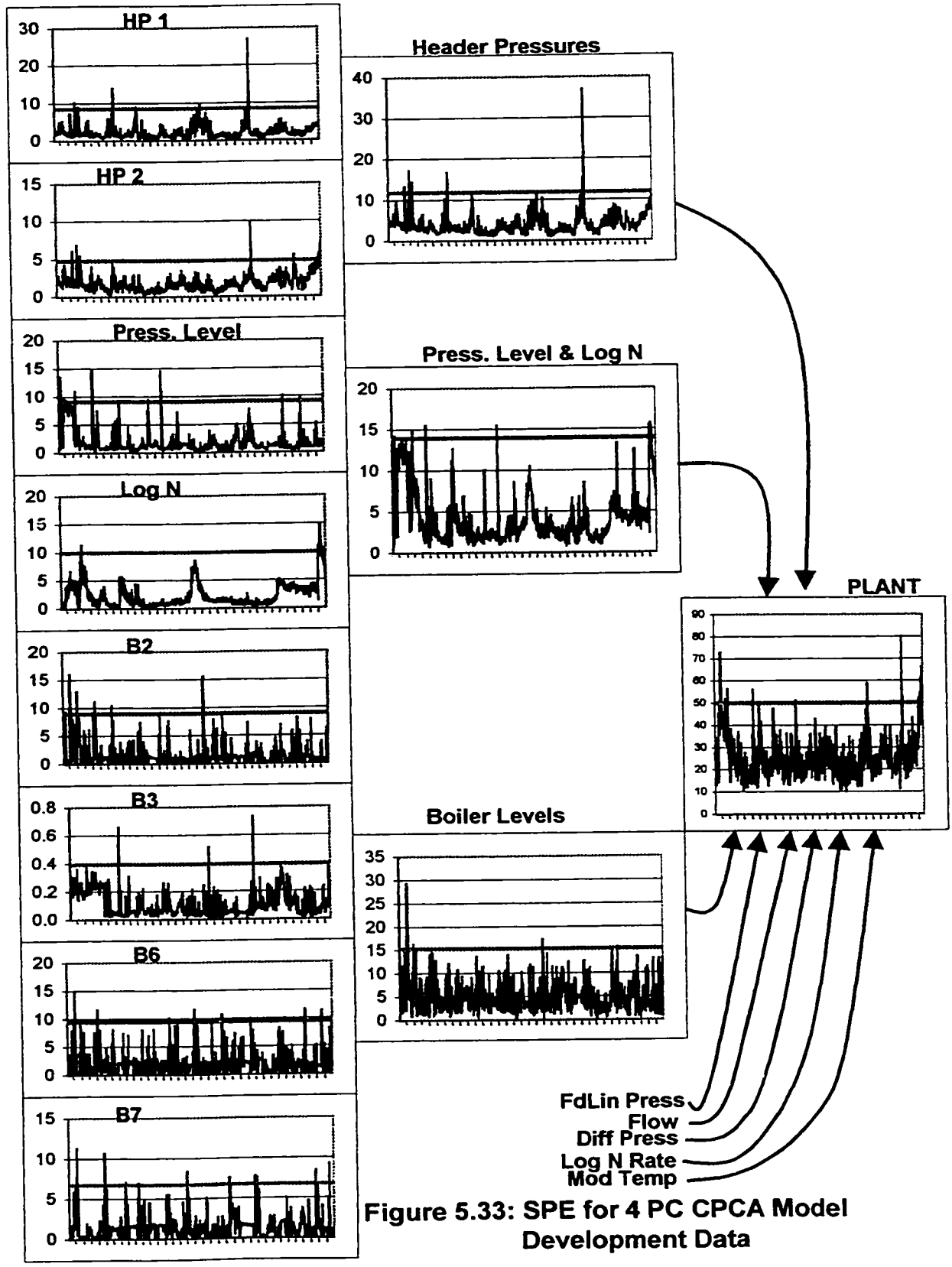


Figure 5.33: SPE for 4 PC CPCA Model Development Data

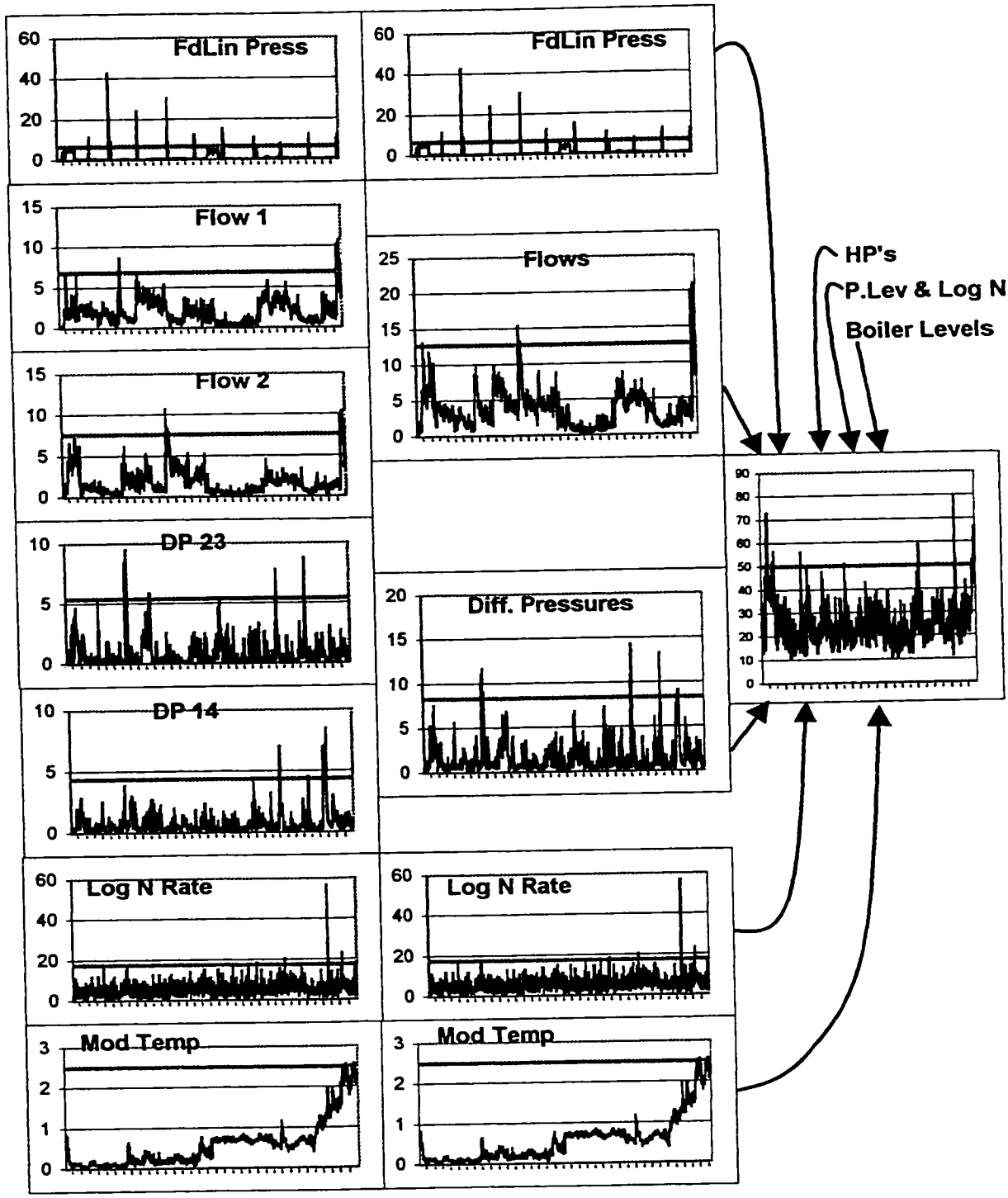


Figure 5.33 con't

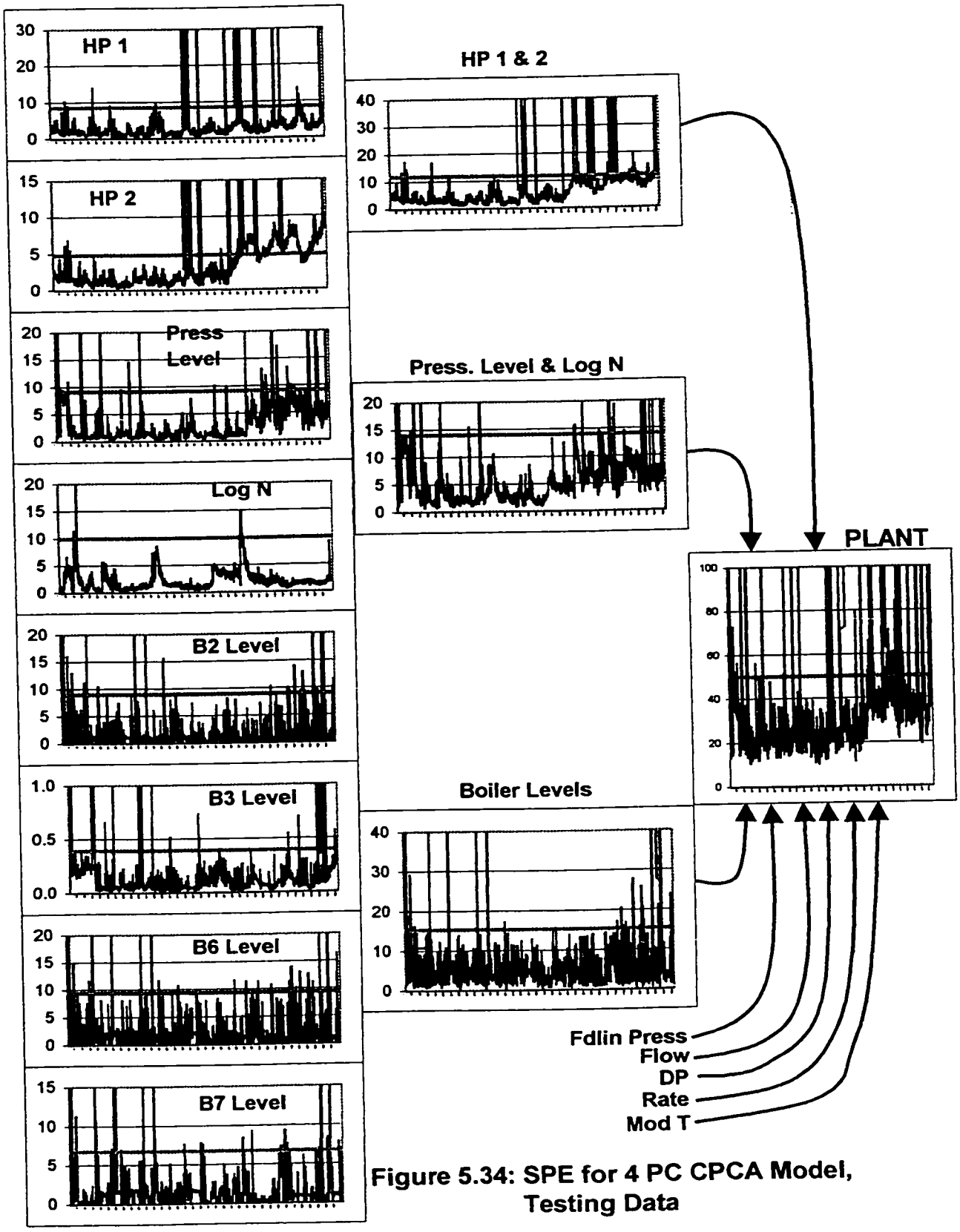


Figure 5.34: SPE for 4 PC CPCA Model, Testing Data

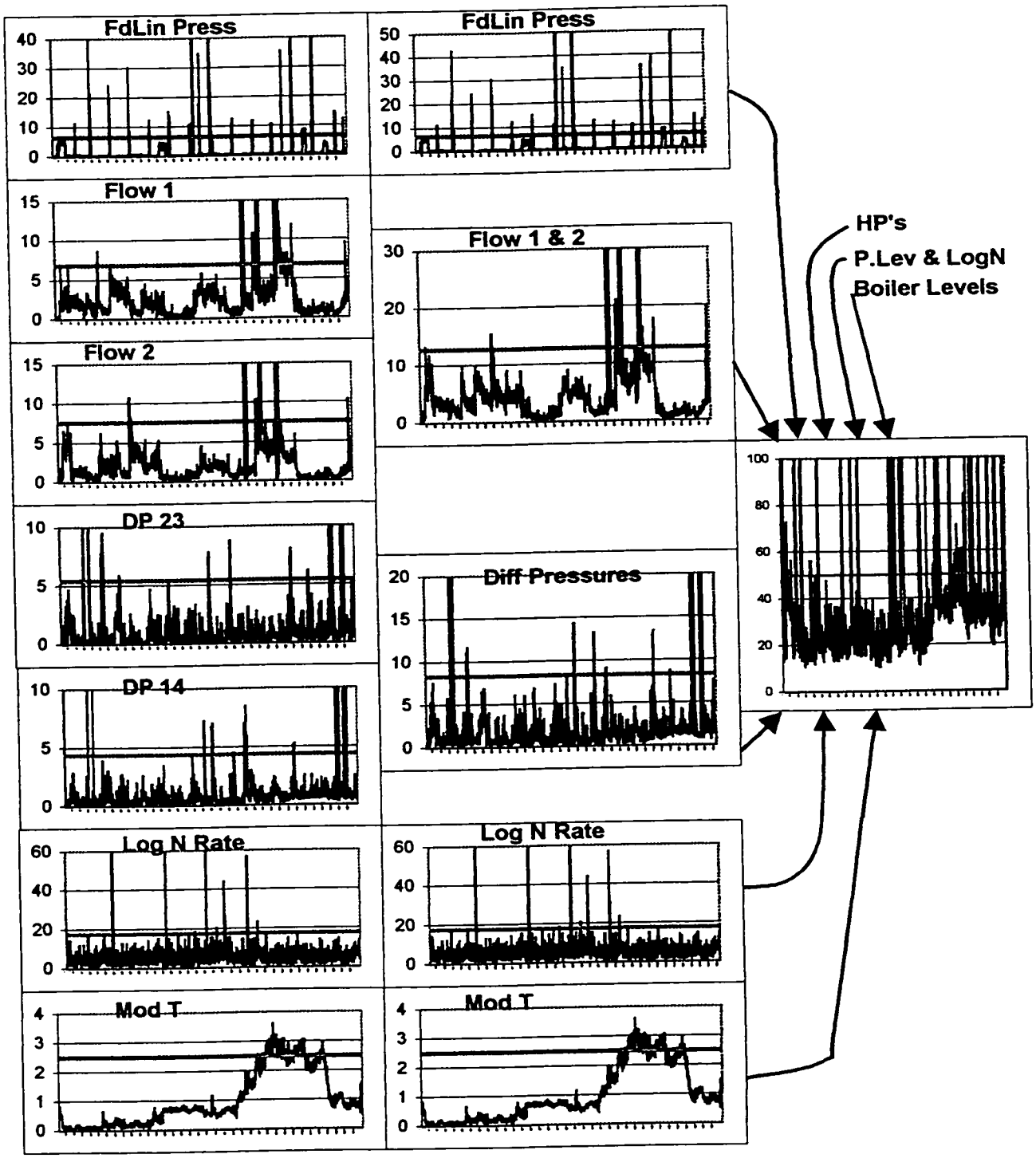


Figure 5.34 con't

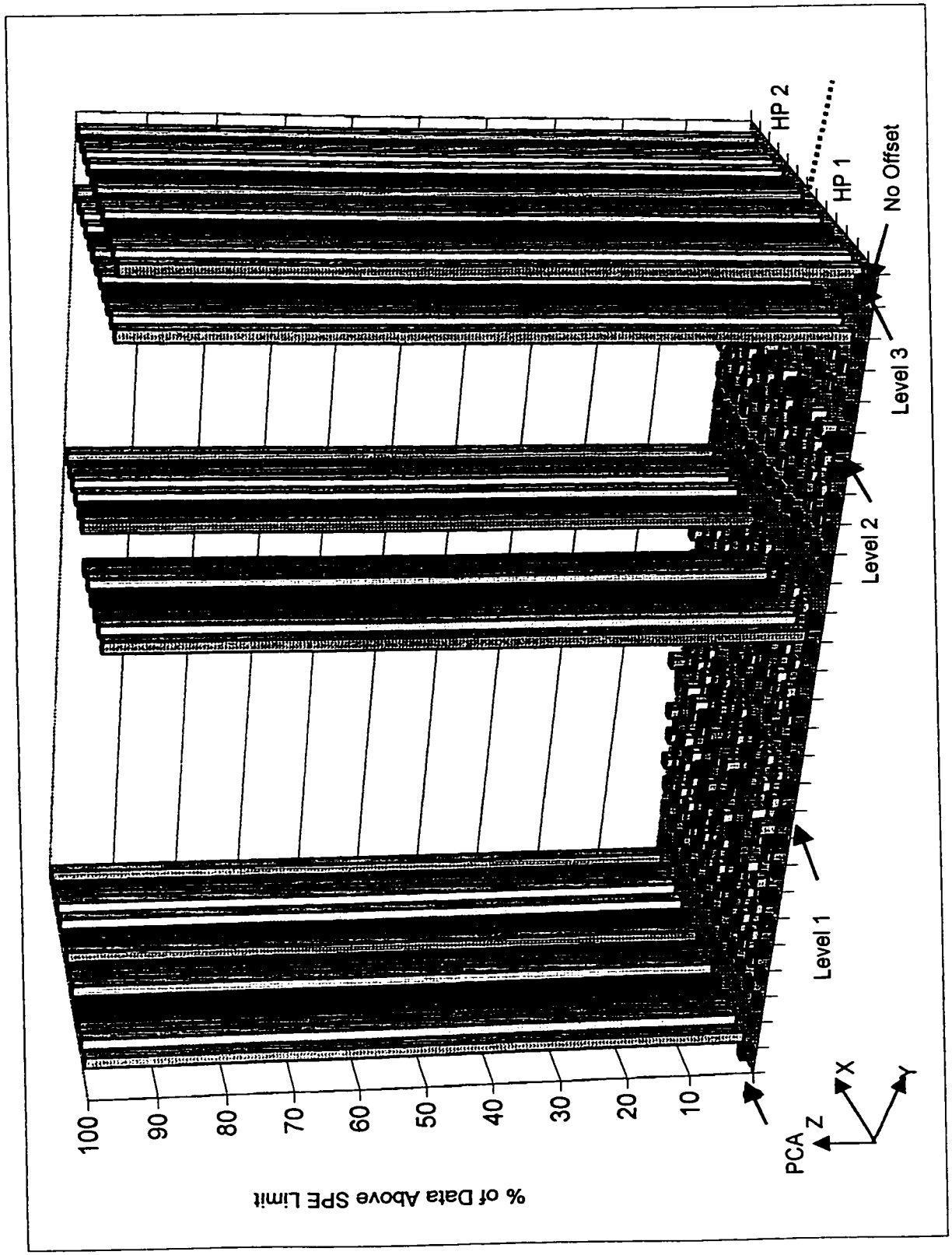


Figure 5.35: Expanded Sensitivity Graph, Header Pressure + 50 kPa Results

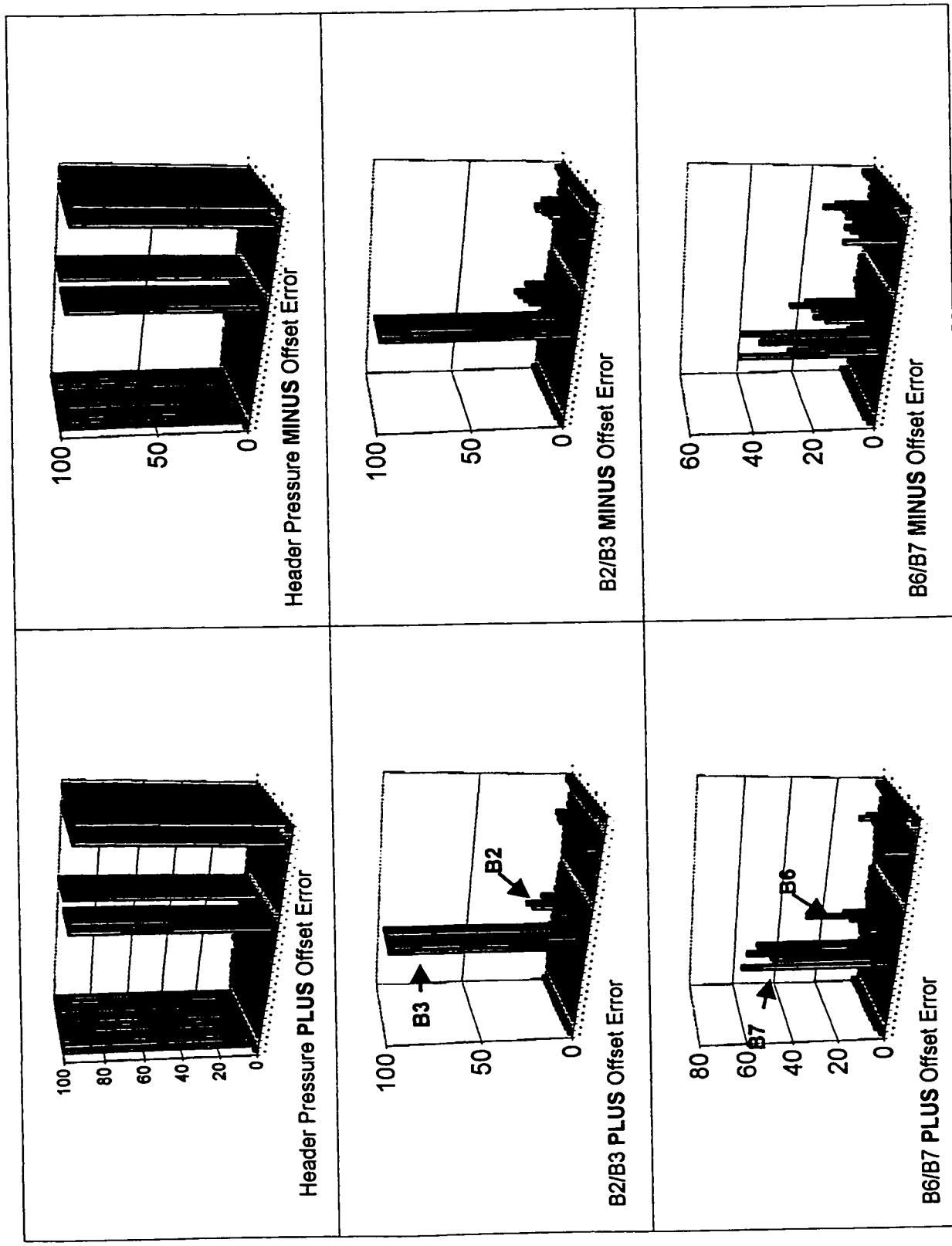


Figure 5.36: Sensitivity Results for Header Pressures and Boiler Levels

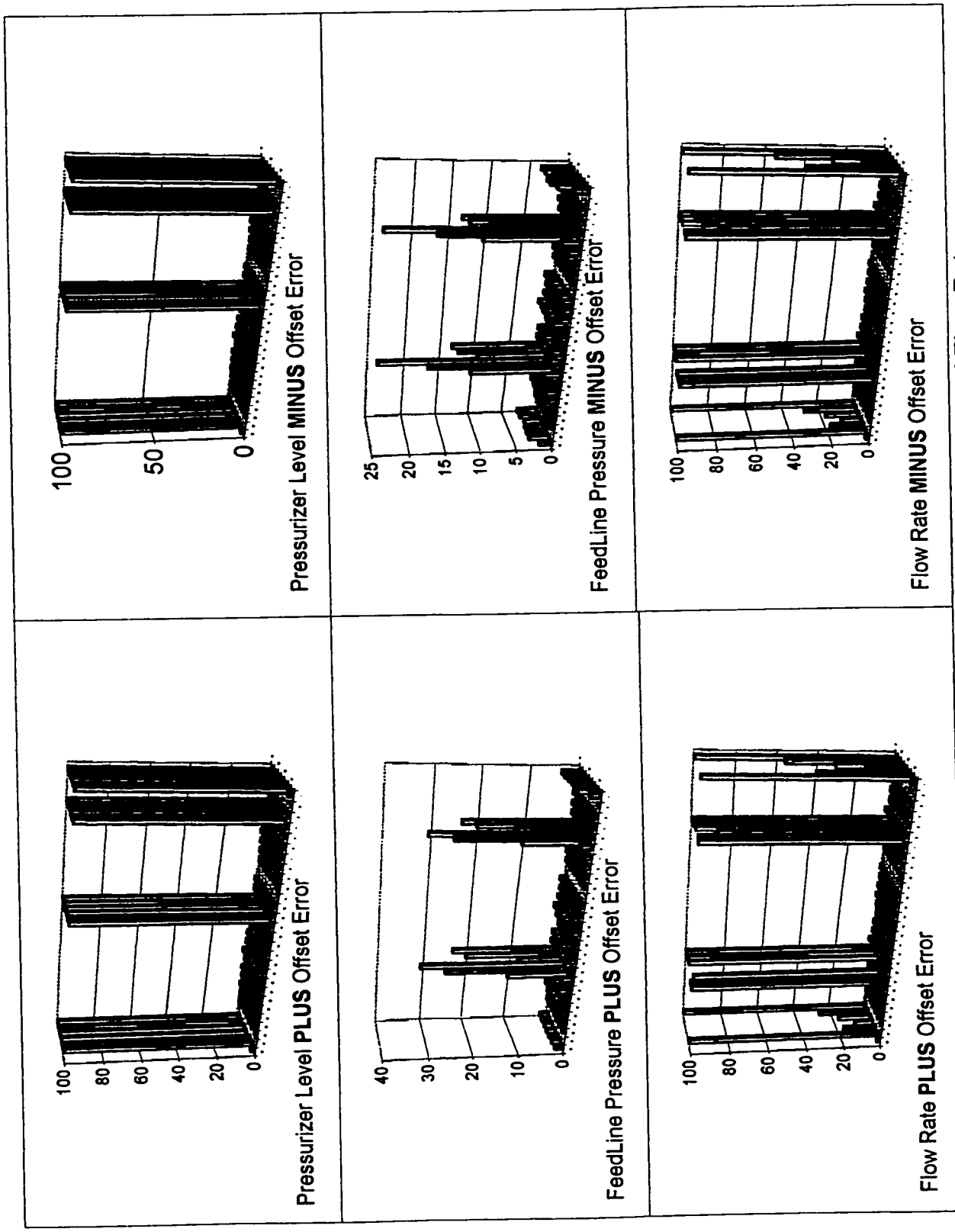


Figure 5.37: Sensitivity Results for Pressurizer Level, FdLin Pressure and Flow Rates

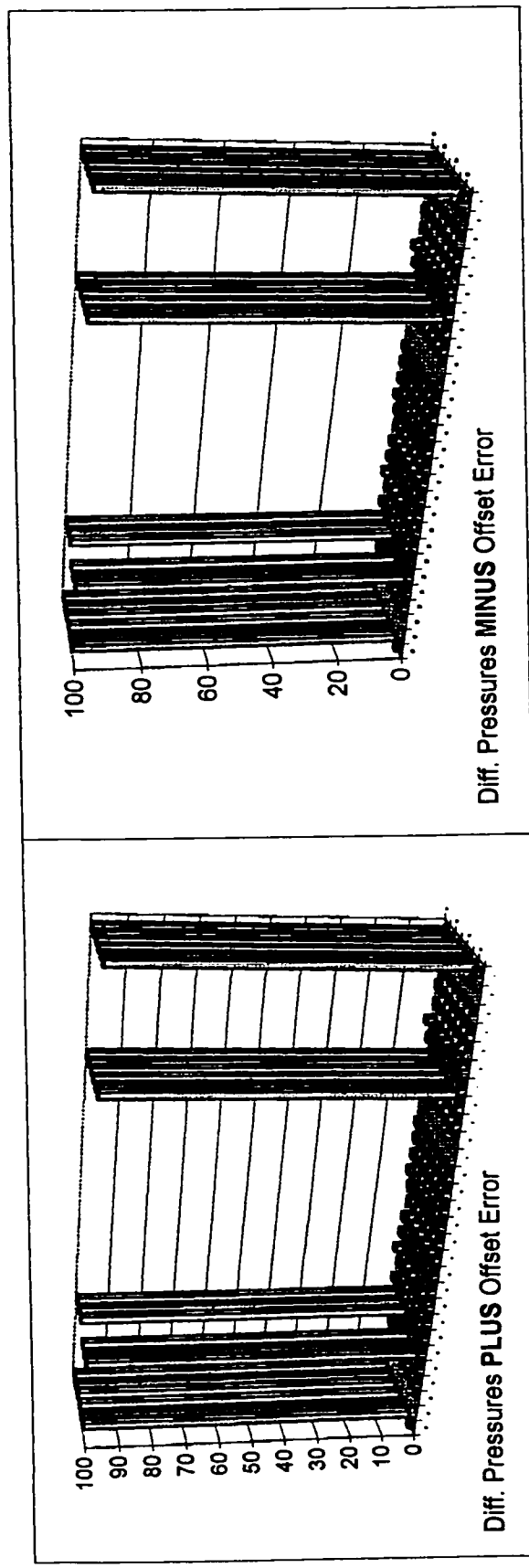


Figure 5.38: Sensitivity Results for Diff. Pressures

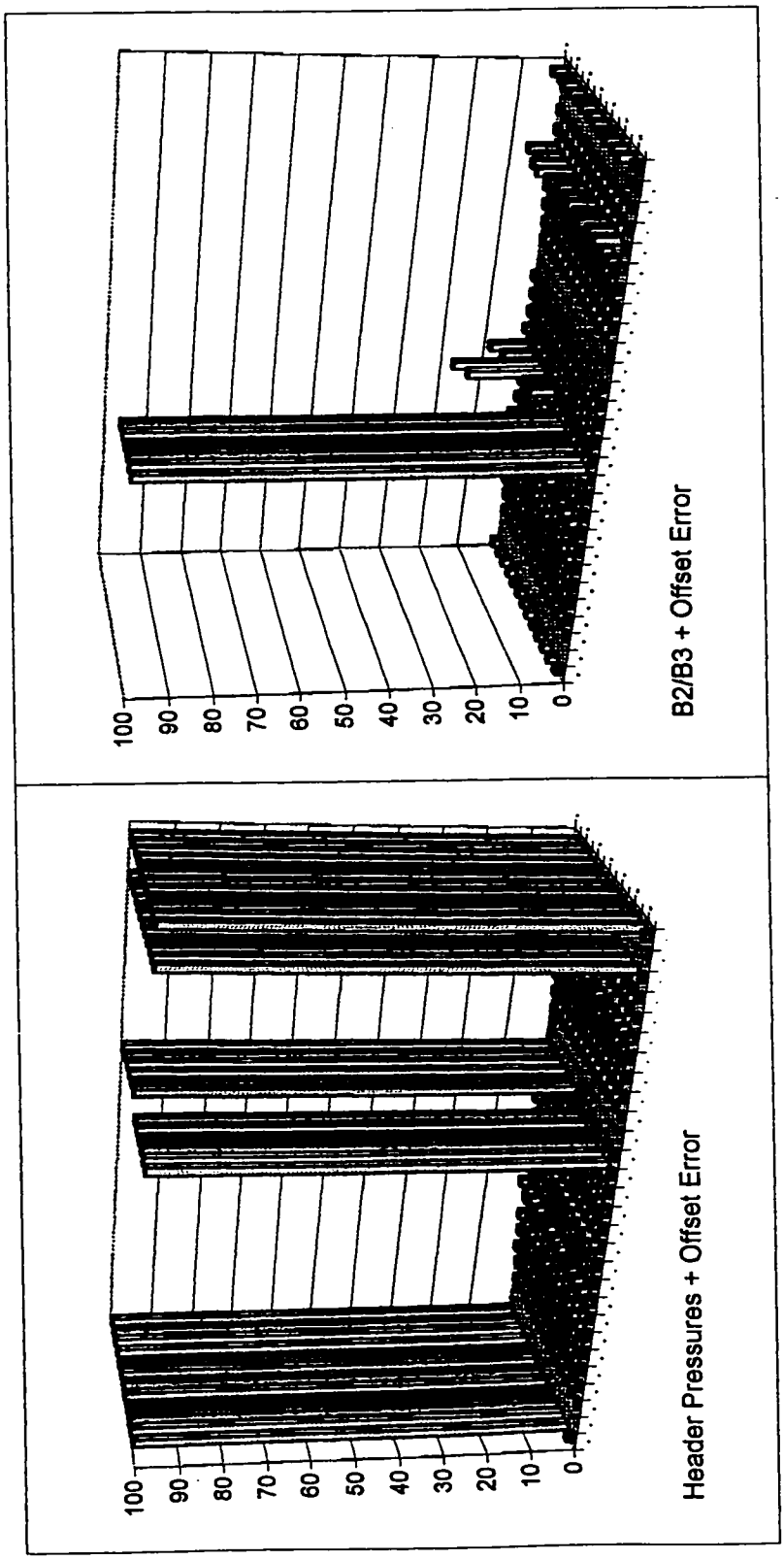


Figure 5.39: Sensitivity Results for HP's and B2/B3 for September Data

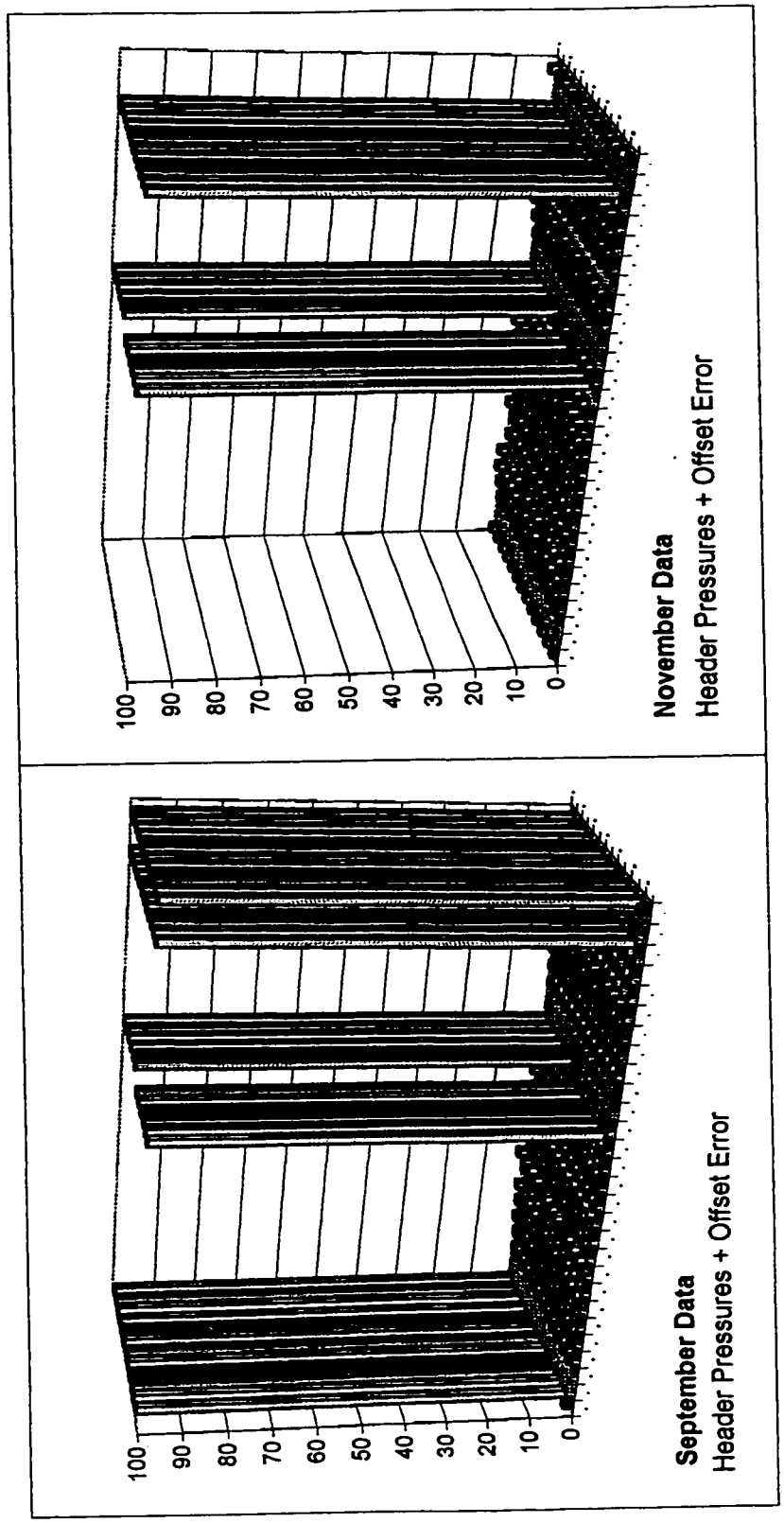


Figure 5.40: Sensitivity Results for Header Pressure for Sept. and Nov. Data

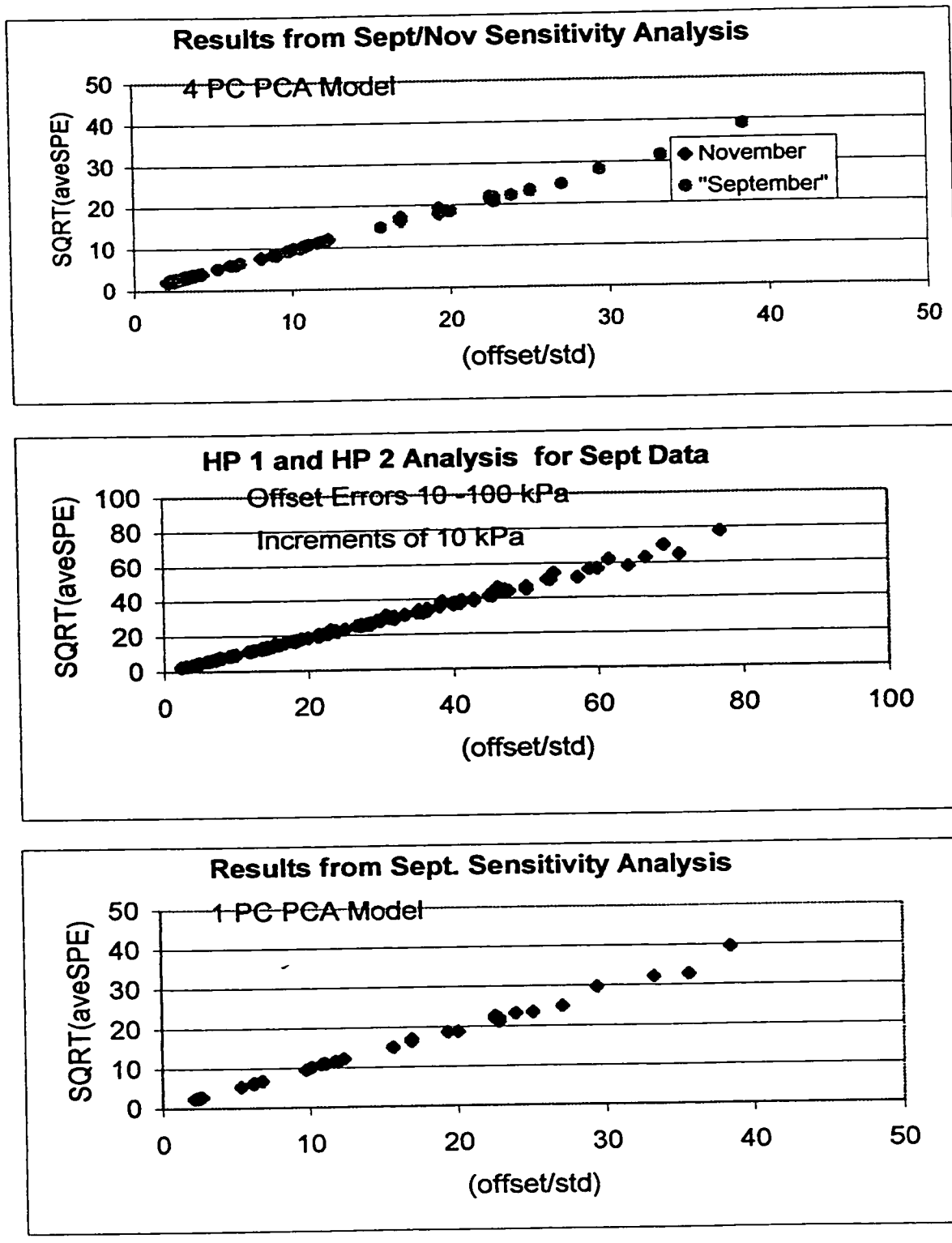


Figure 5.41: Analysis of Average SPE vs Offset Error/STD

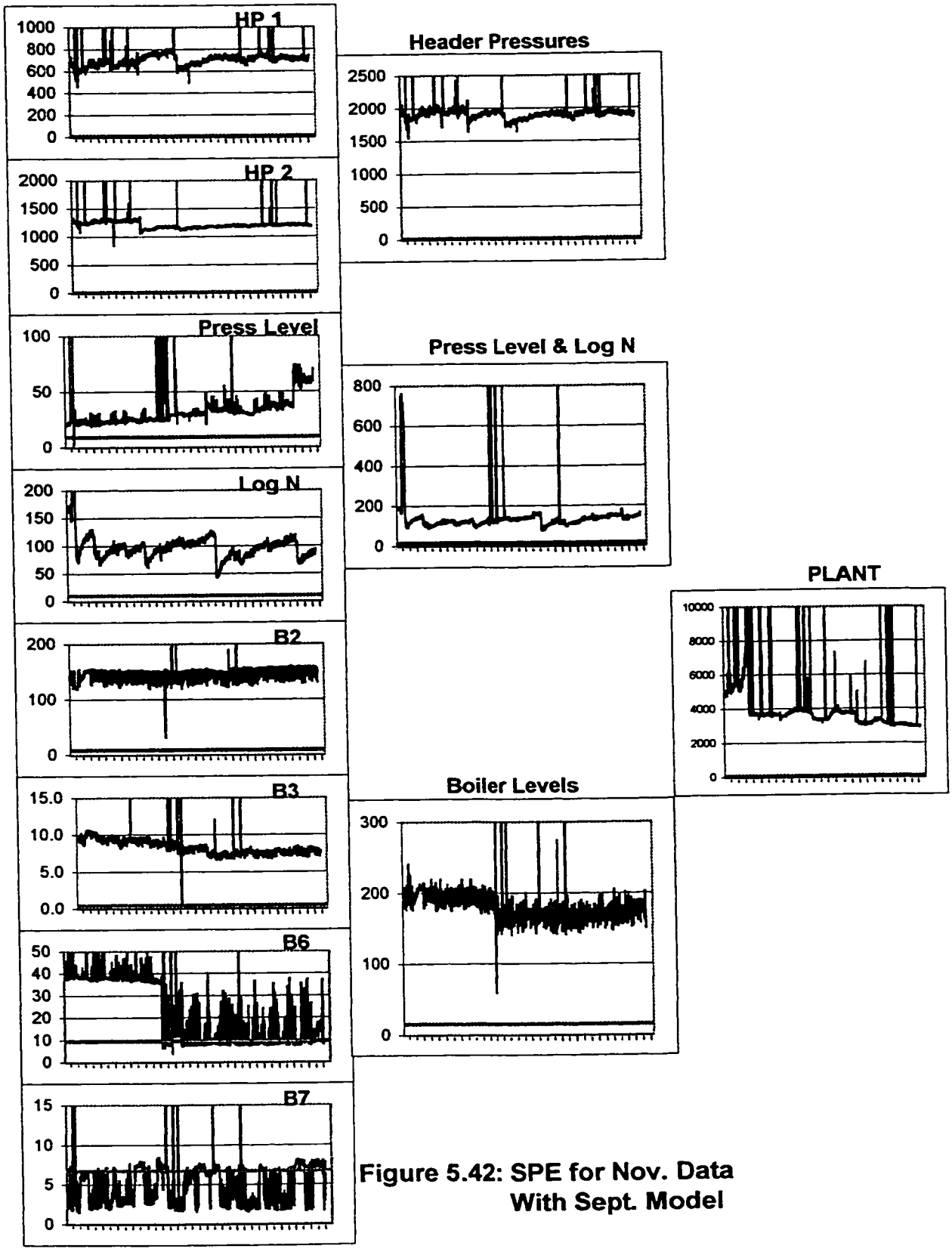


Figure 5.42: SPE for Nov. Data With Sept. Model

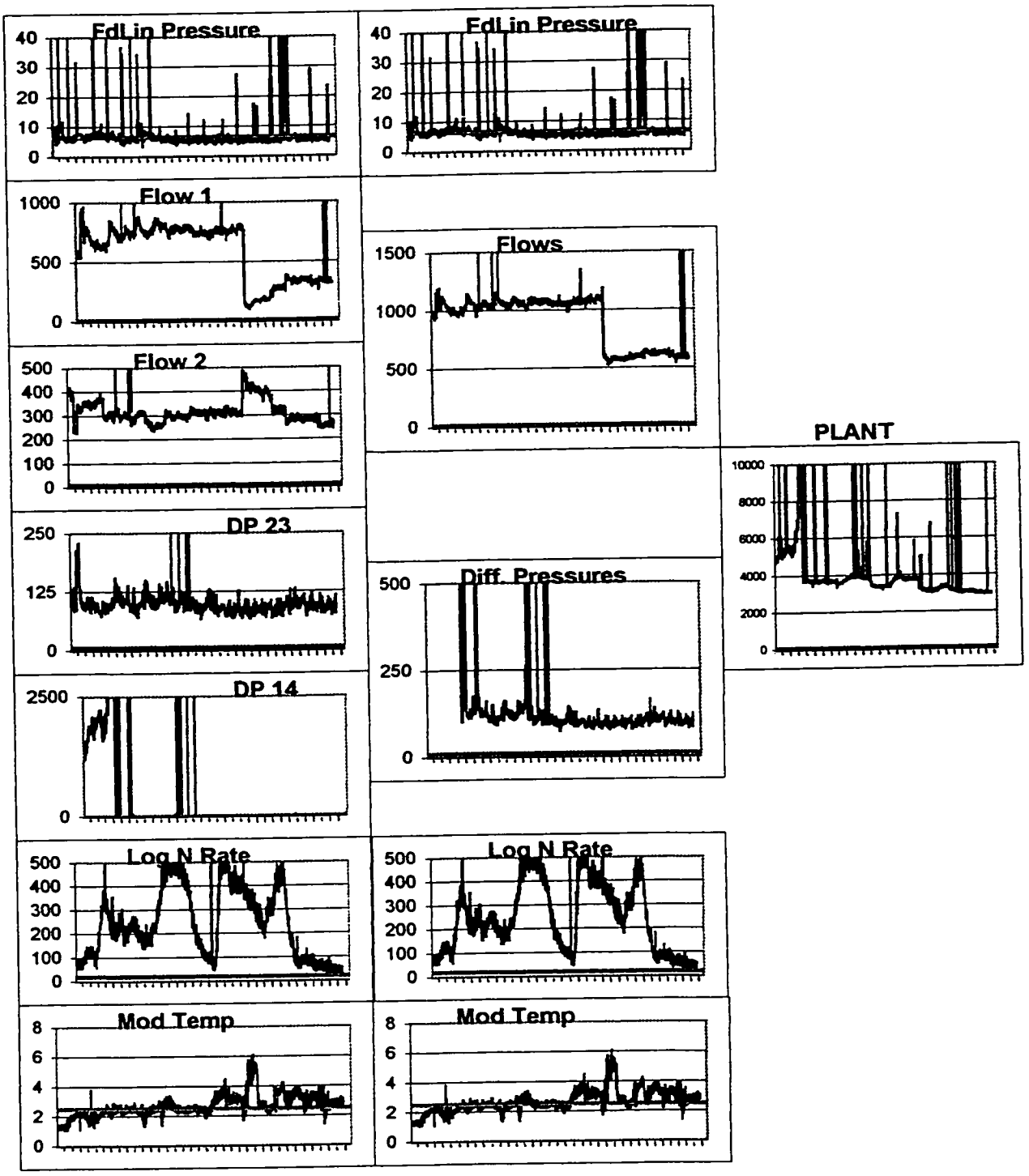


Figure 5.42: con't

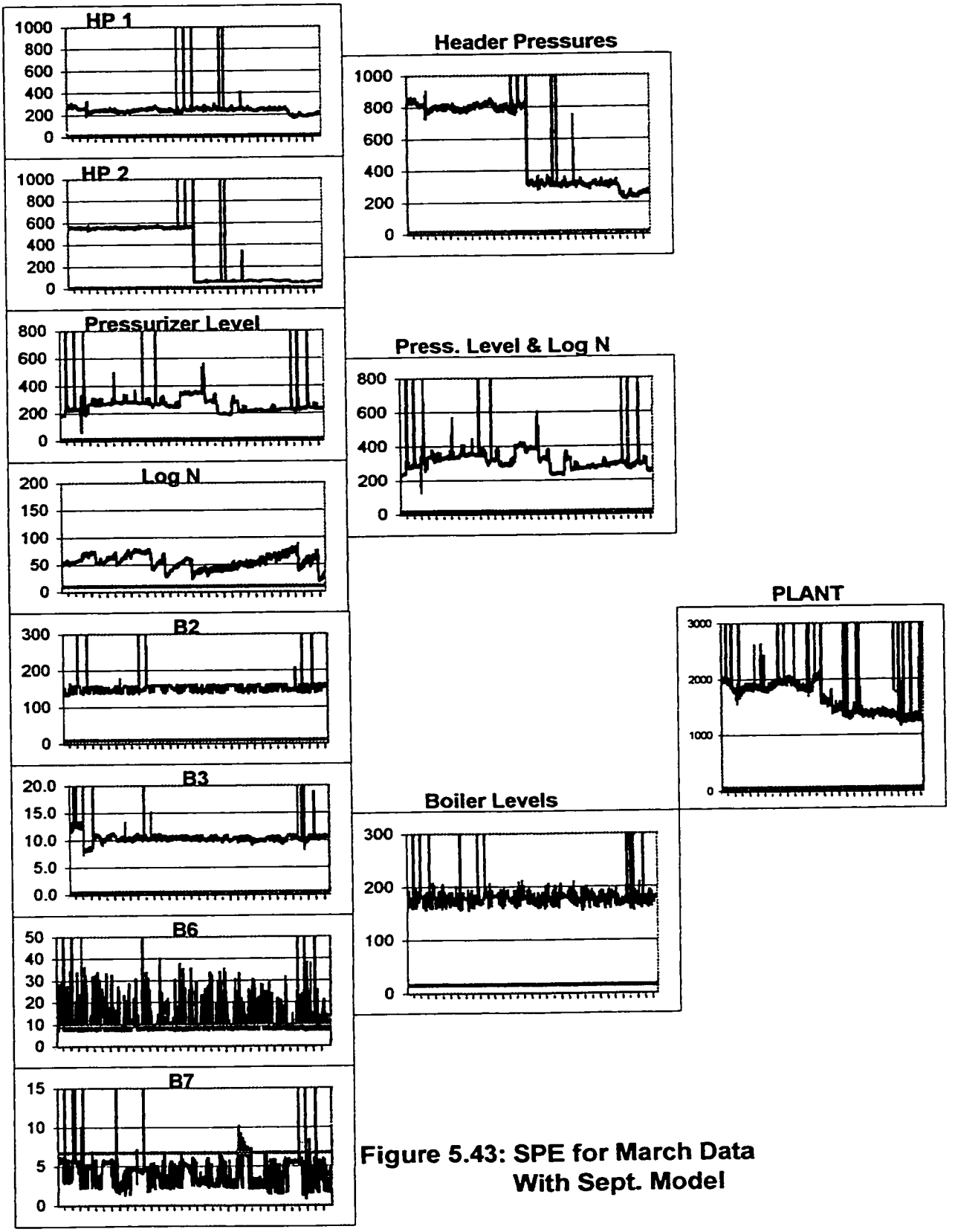


Figure 5.43: SPE for March Data With Sept. Model

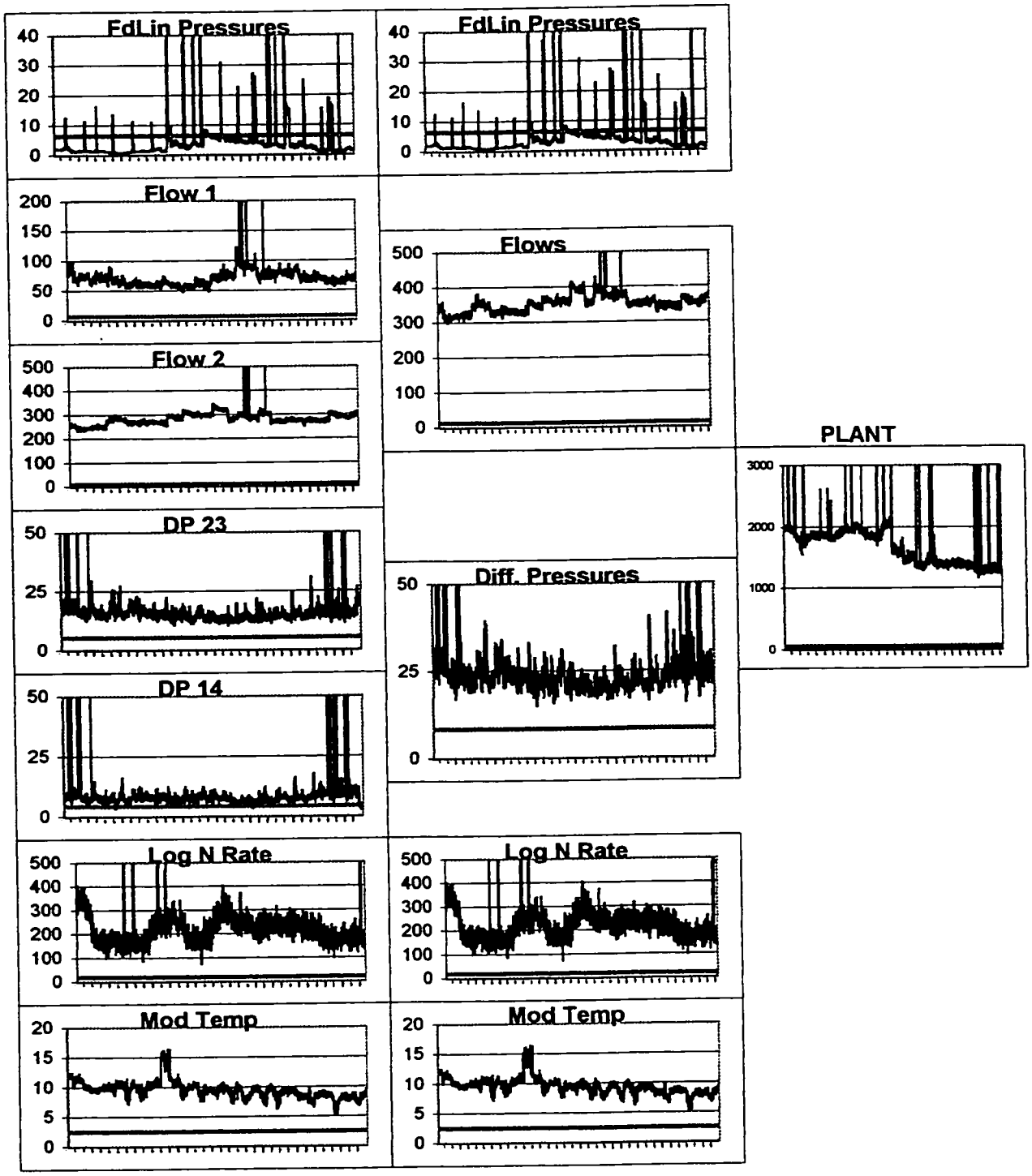


Figure 5.43: con't

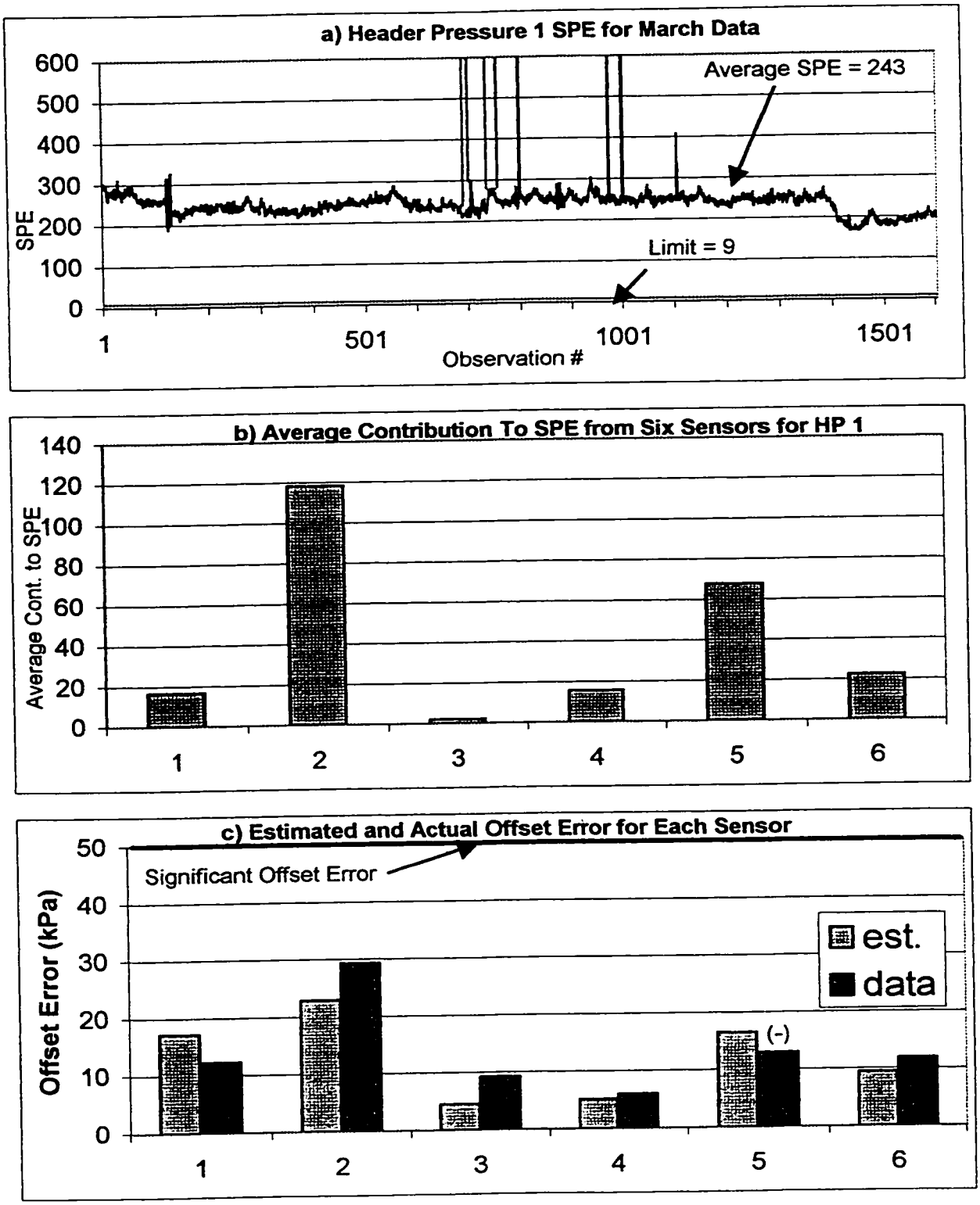


Figure 5.44: Auto Scaling Analysis for Header Pressure 1

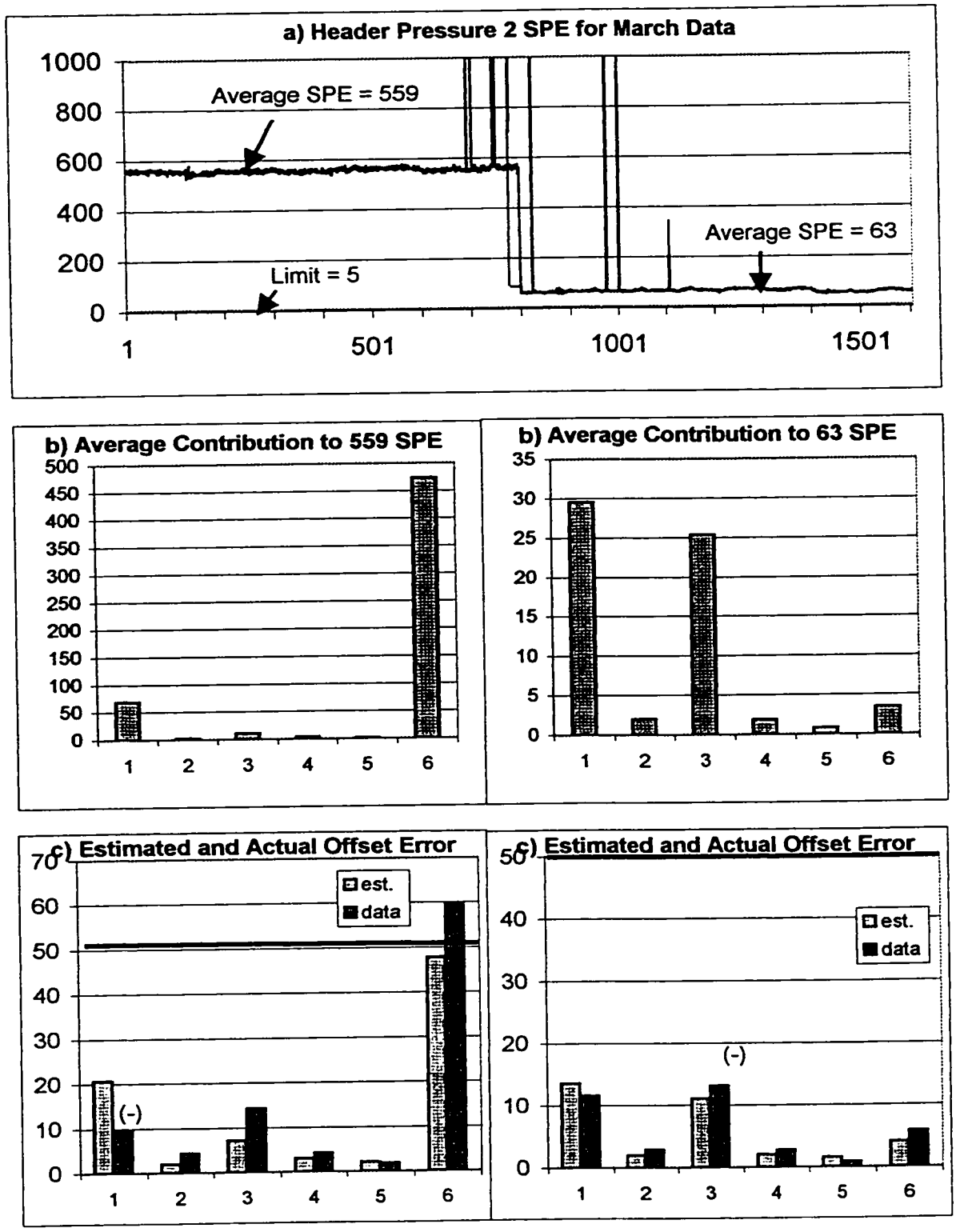


Figure 5.45: Auto Scaling Analysis for Header Pressure 2

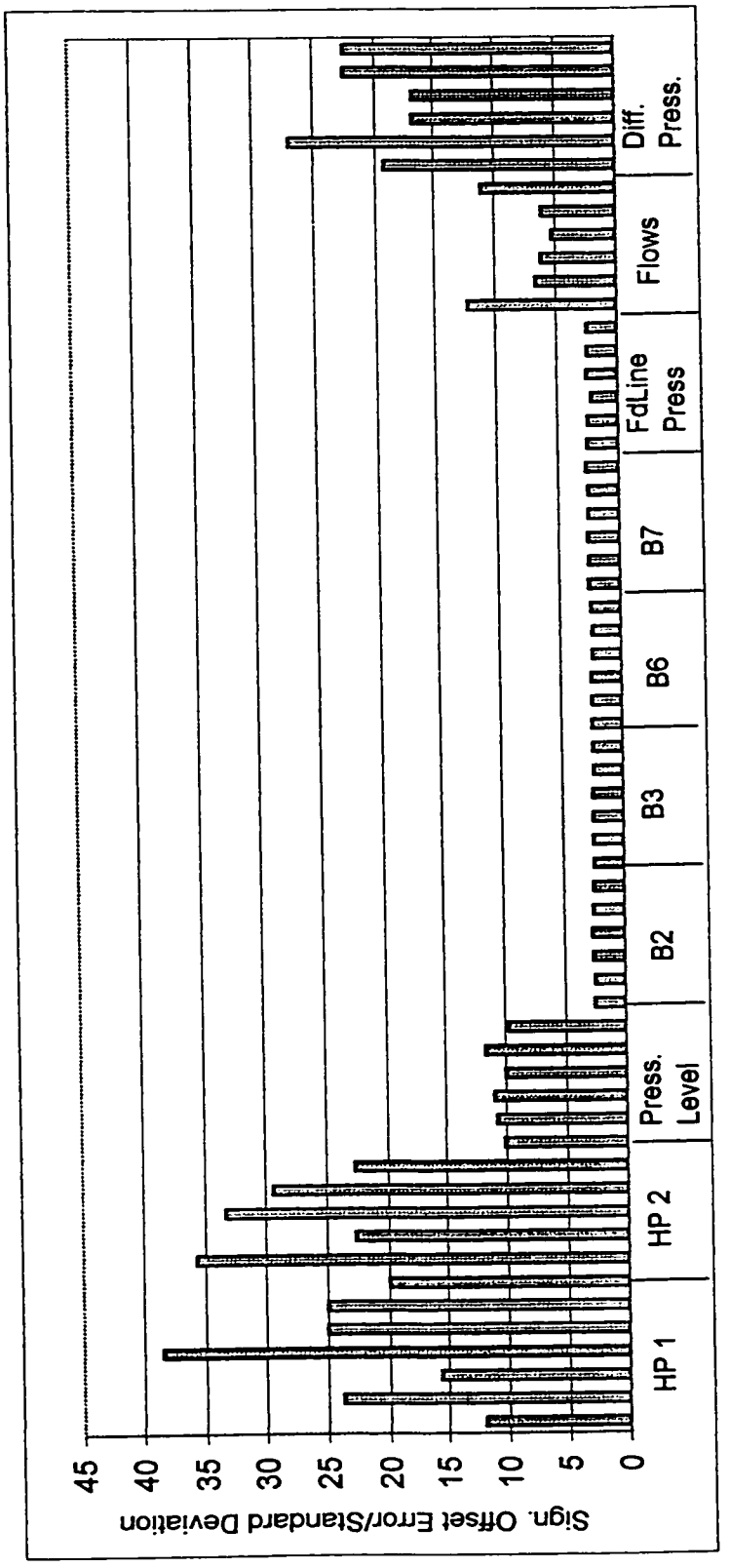
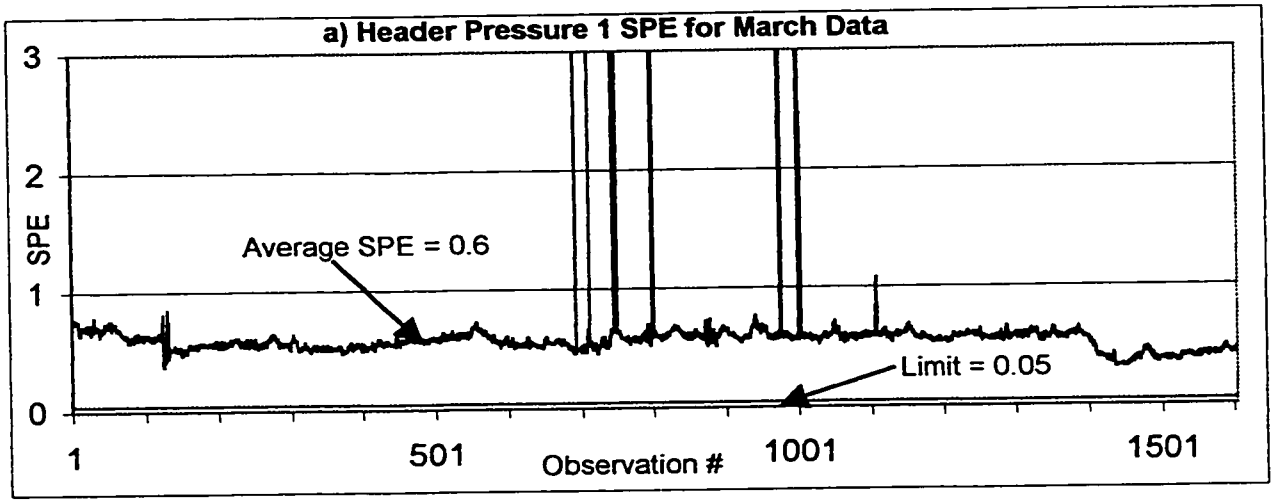
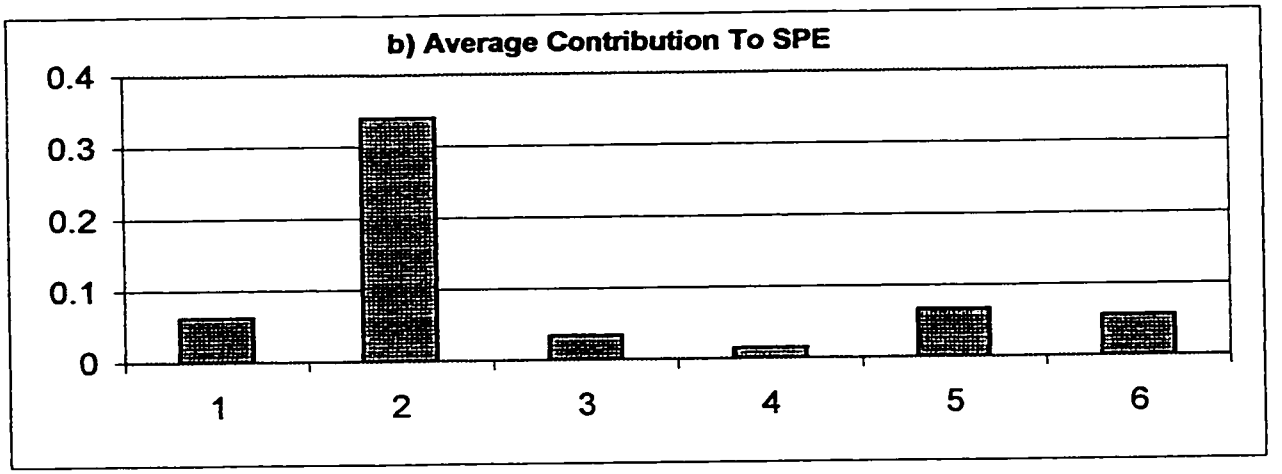
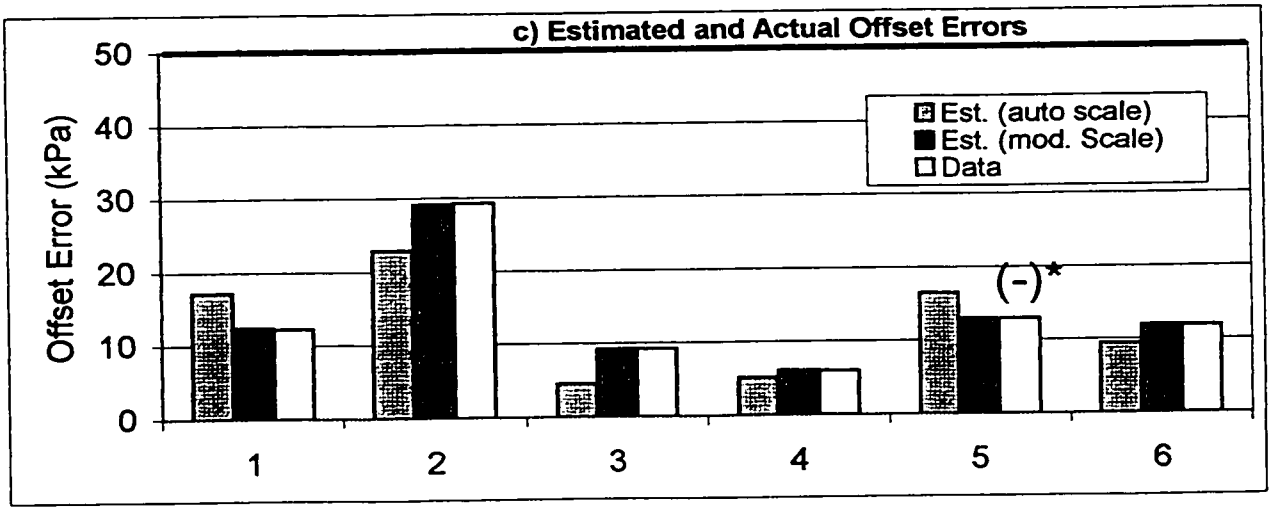


Figure 5.46: Ratio of Significant Offset Errors to Standard Deviations For March Data







* indicates actual offset error (=new(March) data - model(Sept) data) is negative

Figure 5.47: Offset Scaling Analysis for Header Pressure 1

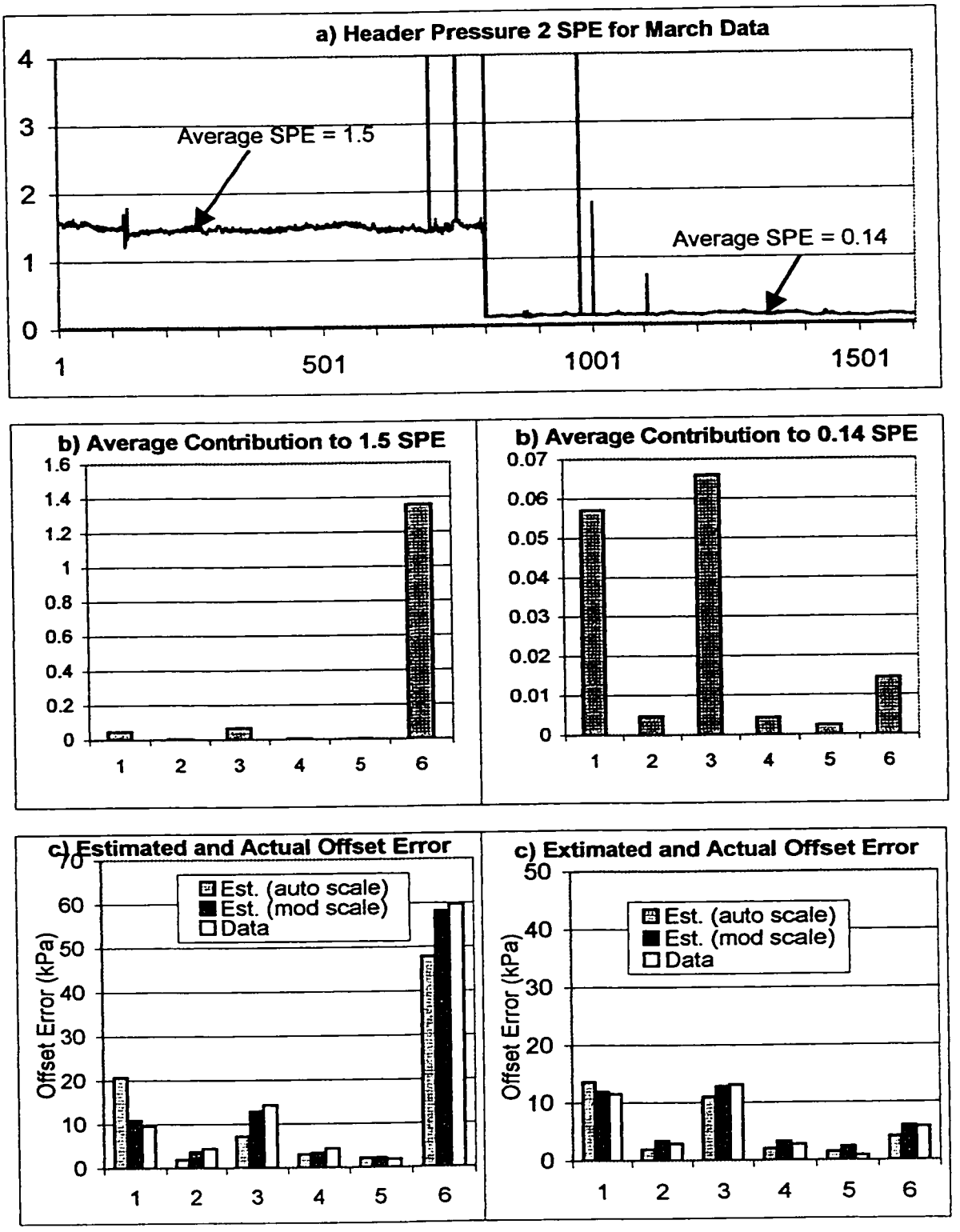


Figure 5.48: Offset Scaling Analysis for Header Pressure 2

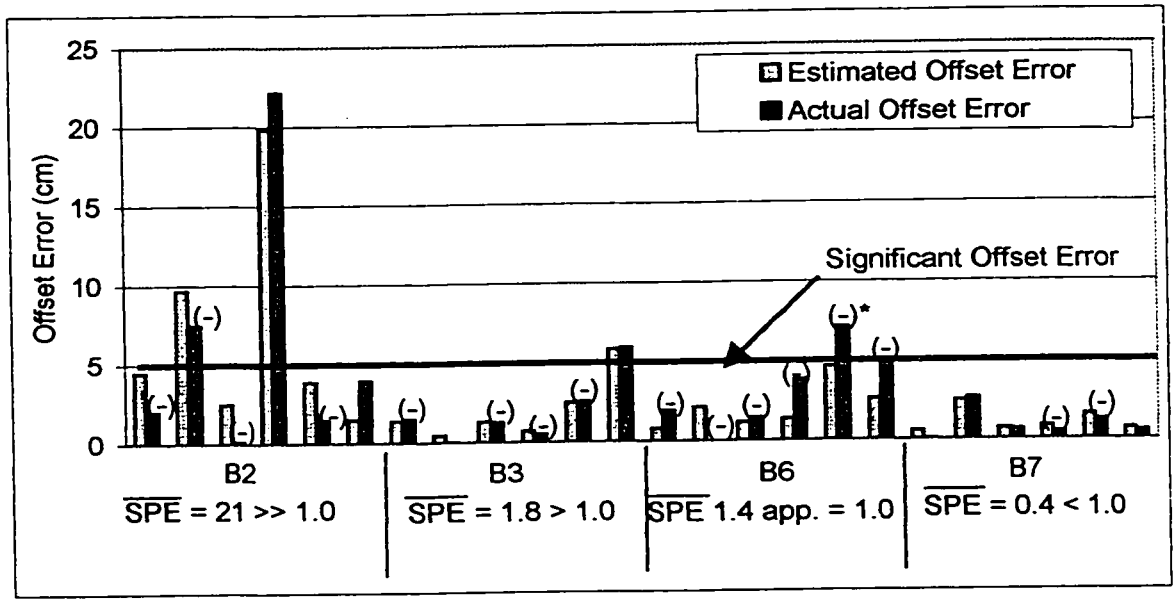
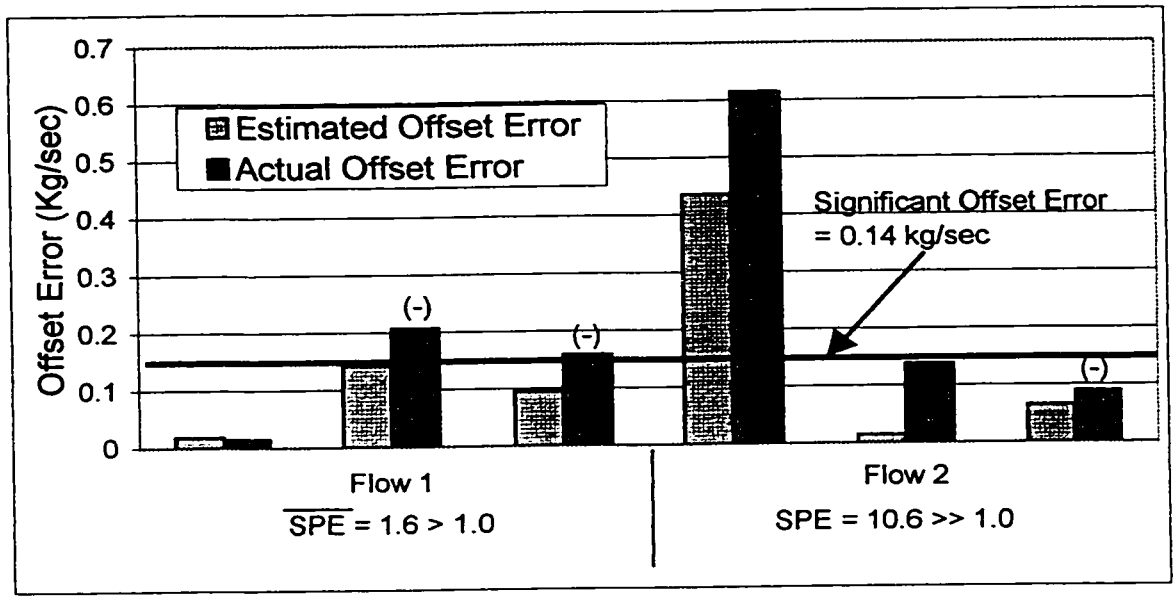


Figure 5.49: Offset Scaling Analysis for Boiler Levels



* indicates actual offset error (=new(March) data - model(Sept) data) is negative

Figure 5.50: Offset Scaling Analysis for Flow Rates

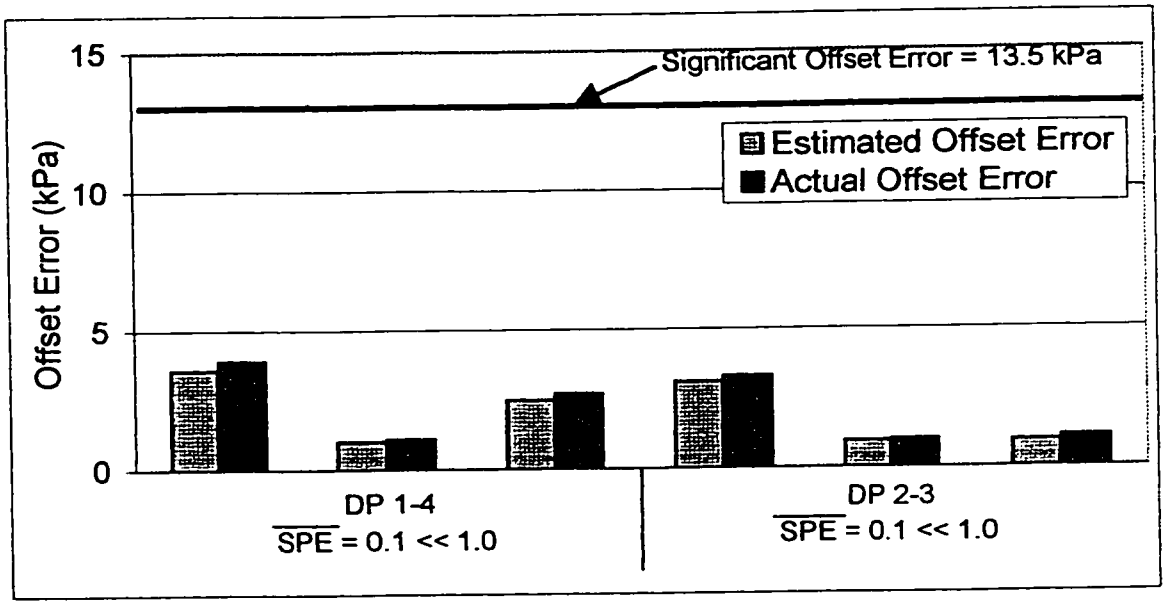


Figure 5.51: Offset Scaling Analysis for Differential Pressures

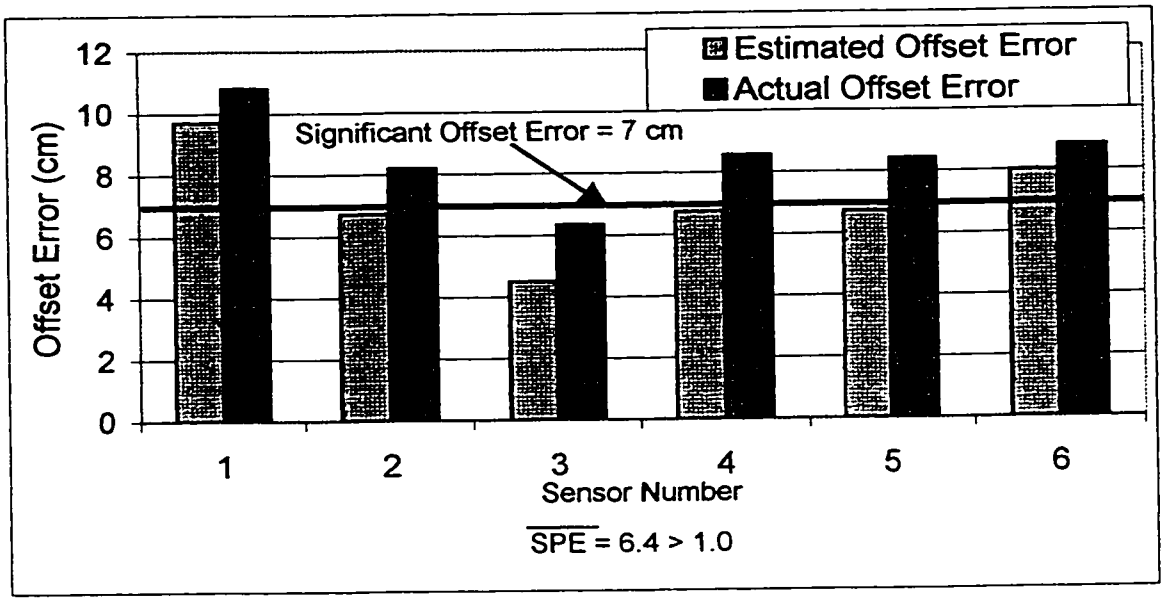


Figure 5.52: Offset Scaling Analysis for Pressurizer Levels

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Introduction

This chapter will be divided into three sections. The first section will review the success criteria related to each of the three hypotheses outlined in Chapter 1. The second section will summarize the results of the process analysis vs. process monitoring work. Finally, areas for future work will be given.

6.2 Review of Hypothesis

Recall from Chapter 1 that there were three main hypotheses to be investigated. They were:

1. Established Multivariate SPC techniques can be used for the analysis of historical datasets generated from CANDU nuclear power plants.
2. A hierarchical process monitoring methodology can be developed which will have the ability to deliver relevant information to different functional groups within a NPP.
3. A systematic methodology can be developed to quantify the sensitivity of a specific process monitoring application.

Each of these hypotheses will be discussed separately.

6.2.1 Hypothesis 1

The first hypothesis was to establish that multivariate SPC techniques could be used for NPP analysis. This has been successfully proven in several areas and is considered a significant contribution to the development of analysis tools for NPP in general. Standard PCA was very useful in identifying data associated with process trip tests. PCA models on individual variables were able to give some key insight into the process itself. Finally, using PCA on several months of data provided insight into how the plant configuration changed from month to month. The scores were able to represent the data in clusters related to different months. The loadings were able to identify variables that changed significantly over the months. The loadings could also distinguish between two general causes of changes in variables; sensor calibrations and process changes. Finally, the standard PCA was able to identify two anomalies in the data that was not known to AECL. First, it identified correlations among the levels of different boilers. Secondly, when considering all the data together, it appears to have detected some sort of process upset involving the boiler feedline pressure and Boilers 3 and 7. The above results satisfy the success criteria given in Chapter 1 of being able to handle NPP data, providing insight into the operation and providing basic diagnostics on anomalies found in the data. Therefore, it was concluded that hypothesis 1 was proven.

6.2.2 Hypothesis 2

The second hypothesis dealt with developing a hierarchical process monitoring methodology to deliver different information to different functional groups. The success criteria stated the methodology should be sensitive to both instrumentation and process

faults, should provide relevant information to different functional groups and should perform basic diagnostic tasks. As a result of this research, a multi-block, multi-level PCA algorithm was developed. Also, a prediction code was developed and tested. Both developments represent a contribution to the current state of the multi-block, multi-level PCA algorithms. The multi-block, multi-level model proved to be a very useful tool for NPP analysis. However, its ability to deliver relevant information to different functional groups was marginal, at best. This marginal performance could be attributed to the scaling used in the model development and is an area for future work.

6.2.3 Hypothesis 3

The goal of the third hypothesis was to develop a method of quantifying the sensitivity of a specific monitoring methodology. This has been accomplished by identifying the following formula:

$$\overline{\mathrm{SPE}} \approx \left(\frac{\text{applied offset error}}{\text{standard deviation}} \right)^2$$

This formula, which intuitively makes sense, was identified through the sensitivity analysis and confirmed while examining the CPCA model validity from month to month. It should be applicable to any dataset. This satisfies both success criteria outlined in Chapter 1 and hence this hypothesis was also considered proven.

6.3 Process Analysis vs. Process Monitoring

During the course of this work, a clear differentiation between process analysis and process monitoring was developed. The results from the investigation of the first hypothesis strongly indicate that multivariate SPC techniques are very useful for NPP

analysis. However, the results from the investigation of the second hypothesis would indicate that monitoring a NPP using multivariate SPC techniques is not as feasible. This could be due to a variety of reasons. First, as discussed in Section 5.4.3, it appears that auto-scaling is not the correct scaling method for this type of data. This point will be expanded on in the next section. Secondly, there may be so many normal plant configurations, a true steady state operation is never attained. In this case, SPC will not be applicable. However, over-riding both of these reasons, is the issue of defining the goals of the monitoring methodology. Again, from Section 5.4.3, it was determined that if the September data was used as the representative dataset, there are offset differences in the raw March data for certain variables that are greater than the significant offset error limit defined by AECL and shown in Table 3.2. Therefore, the ideal monitoring methodology would be indicating that faults are present in the March data. It is debatable as to whether this is truly the goal of the utility and AECL and perhaps, the goals of the monitoring methodology should be reexamined. If the goals were based more on how the plant is currently operating as opposed to how it theoretically should be operating, perhaps multivariate SPC would provide a feasible monitoring option.

6.4 Future Work

During the course of this research, several areas for future investigation have been identified. The future work will be discussed in two respects; further development of monitoring tools for nuclear power plants and further theoretical research into the multi-level, multi-block PCA algorithm.

With respect to the development of a tool to monitor a NPP, there are several interesting areas for future work. The first step that should be taken is to review the current results with AECL. This review would show that there are indeed significant offset errors, as they are currently defined, in data that has been considered normal or fault free. This could lead to a reevaluation of the goals and objectives of the monitoring methodology and hence multivariate SPC methods could prove feasible. Secondly, the process analysis using three months of data and the sensitivity analysis should be redone using the offset error scaling method outlined in Chapter 5. Again, this could lead to better monitoring results and also better process analysis results. Thirdly, the issue of identifying the proper dataset for the reference model should be examined. In sections 5.2 and 5.4.3, it was noted that the data from the three different months did not appear to represent the same normal, steady state conditions. Therefore, one reference dataset does not appear to be applicable. One possible solution for this could be to analyse the data using differencing. Here, differencing means using the difference between the individual sensor readings and an established values for a specific variables to develop the models and do the analysis. The established value could be the setpoint of the specific variable. This could eliminate false alarms due to the plant moving from one normal point to another. A second solution could be to use an adaptive reference dataset. In this scenario, the reference dataset would be updated as the plant moved to a new normal operating point. However, care would need to be taken to ensure the new reference dataset represented a true normal operating point and not a fault condition.

In terms of the theoretical development of the multi-level, multi-block PCA algorithm, there are again several avenues for investigation. Most of the work would center on providing statistical bases for various aspects of the model development. For example, the statistical validity of using 1.0 as a control limit when scaling with the significant offset errors could be investigated. Also, significance tests for the number of PC's required in models using NPP data could be examined. In Chapter 4, it was noted that the HPCA algorithm appears to explain more sum of squares for deflation using level 1 than does CPCA. At present, this is an experimental observation. Work could be completed to determine if this is a property of the algorithms. Finally, missing data has not been considered in the development of the multi-block, multi-level PCA algorithm. It is suspected that the standard method of handling missing data will be acceptable in models that have several sensors in the blocks in level 1. However, if a level 1 block contains only one or two sensors, the method may break down.

In summary, the Multivariate SPC techniques (both standard PCA and multi-block, multi-level PCA) are very useful for analyzing operational data from a nuclear power plant. However, their performance for monitoring the process in an on-line manner is marginal. This marginal performance can be attributed to the goals set for the monitoring methodology, the scaling method used in the analysis and the numerous normal plant operating states.

REFERENCES

1. A. Wyatt, "*Electric Power; Challenges and Choices*", The Book Press Ltd., P.O. Box 5971, Stn. 'A', Toronto, ON Canada, M5W 1P4, 1986

2. AECL, "*CANDU 6, Sharing the Success*", Atomic Energy of Canada Limited, 2251 Speakman Drive, Mississauga, ON Canada, L5K 1B2, August 1996

3. S. Groom, NBP-PLGS, "*Consequences of Foreign Materials Being Left in the PHTS at Point Lepreau*", Proceedings of the 1996 CNA/CNS Conference, Fredericton, NB, June 9-12, 1996

4. C. Andognini, "*Managing Nuclear Reactors*", Reprint from Electricity International, Oct. 1997

5. M. DeVerno, H. Pothier, J. de Grobois, M. Bosnich, C.Xian, J. Hinton, G.Gilks, "*Canadian CANDU Plant Historical Data Systems; A Review and Look to the Future*", Proceedings of the 1996 CNA/CNS Conference, Fredericton, NB, June 9-12, 1996

6. D.M. Marquardt, "*PQM - Product Quality Management*", E.I. du Pont de Nemours & Co. Engineering Department, Applied Statistics Group, Wilmington, Delaware, 19898. 1988 Edition

7. J.E. Jackson, "*A User's Guide to Principal Components*", Wiley-Interscience, John Wiley & Sons Inc., New York, 1991

8. J.F. MacGregor, T. Kourti, "*Statistical Process Control of Multivariate Processes*", Control Eng. Practice, Vol. 3, No. 3, pp. 403-414, 1995

9. A. Wilsky, "*A Survey of Design Methods for Failure Detection in Dynamic Systems*", Automatica, 12, 601-611, 1976

10. R. Isermann, "*Process Fault Detection Based on Modeling and Estimation Methods: A Survey*", Automatica 20, 387-404, 1984

11. M. Basseville, "*Detecting Changes in Signals and Systems – A Survey*", Automatica, 24(3) 309-326, 1988

12. J.J. Gertler, "*A Survey of Model Based Failure Detection and Isolation in Complex Plants*", IEEE Control Systems Magazine 8(6), 3-11, December, 1988
13. P.M. Frank, "*Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge Based Redundancy – A Survey and Some New Results*", Automatica, 26(3), 459-474, 1990

14. R. Isermann, P. Balle, "*Trends in the Application of Model Based Fault Detection and Diagnosis of Technical Processes*", Control Engineering Practice, 5(5), 709-719, 1997

15. E.S. Page, "*Continuous Inspection Schemes*", Boimetrika, Vol.41, pp 100-115, 1954

16. J.M. Lucus, "*The Design and Use of V-Mask Control Schemes*", Journal of Quality Technology, Vol.8, No.1, pp 1-12, January 1976

17. G.E.P. Box, W.G. Hunter, J.S. Hunter, "*Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building*", Wiley Series in Probability and Mathematical Statistics, Wiley, New York, Toronto, 1978

18. T.P. Ryan, "*Statistical Methods for Quality Improvement*", John Wiley & Sons Inc., 1989

19. T.J. Harris, W.H. Ross, "*Statistical Process Control Procedures for Correlated Observations*", The Canadian Journal of Chemical Engineering Vol. 69 pp. 48-57, February 1991

20. R.B. Crosier, "*Multivariate Generalizations of Cumulative Sum Quality-Control Schemes*", Technometrics, 30, 291-303

21. C.A. Lowry, W.H. Woodall, C.W. Champ, S.E. Rigdon, "*A Multivariate Exponentially Weighted Moving Average Control Chart*", Technometrics, 34, 46-53

22. I.T. Jolliffe, "*Principal Component Analysis*", Springer-Verlag, pp. 1-5, 1986

23. J.V. Kresta, J.F. MacGregor, T.E. Marlin, "*Mulitvariate Statistical Monitoring of Process Operating Performance*", The Canadian Journal of Chemical Engineering, Vol.69, pp.35-47, February 1991

24. S. Wold, "*Cross-Validatory Estimation of the Number of Components in Factor and Principal Component Models*", Technometrics, 20, pp. 397-405, 1978

25. P. Geladi, B.R. Kowalski, "*Partial Least Squares Regression: A Tutorial*", Anallytica Chemica Atca, Vol.185, pp. 1-17, 1986

26. P. Nomikos, J.F. MacGregor, "*Multivariate SPC Charts for Monitoring Batch Processes*", Technometrics, Vol. 37, No.1, pp.41-59, February 1995

27. S. Wold, P. Geladi, K. Esbessen, J. Ohman, "*Multi-Way Principal Components - and PLS-Analysis*", Journal of Chemometrics, Vol.1, pp.41-56, 1987

28. B.M. Wise, N.B. Gallagher, J.F. MacGregor, "*The Process Chemometrics Approach to Process Monitoring and Fault Detection*", Preprints of the IFAC Workshop on On-Line Fault Detection and Supervision in the Chemical Process Industries, Newcastle June 1995

29. P. Geladi, S. Wold, "*PCA of Multivariate Images*", Chemometrics and Intelligent Laboratory Systems, 5, pp.209-220, 1989

30. W. Ku, R.H. Storer, C. Georgakis, "*Disturbance Detection and Isolation by Dynamic Principal Component Analysis*", Chemometrics and Intelligent Laboratory Systems, 30, pp.179-196, 1995

31. S. Haykin, "*Neural Networks, A Comprehensive Foundation*", Macmillan College Publishing Company Inc., 1994

32. T. Sorsa, H.N. Koivo, "*Application of Artificial Neural Networks in Process Fault Diagnosis*", Automatica, Vol. 29, No.4, pp.843-849 1993

33. J.W. Hines, D.W. Miller, B.K. Hajek, "*Merging Process Models with Neural Networks for Nuclear Power Plant Fault Detection and Isolation*", 9th Power Plant Dynamics, Control and Testing Symposium Proceedings, Vol.2, pp. 54.01-54.12, 1995

34. J.B. Gomm, "*Process Fault Diagnosis Using a Self-Adaptive Neural Network With On-Line Learning Capabilities*", IFAC Workshop on On-Line Fault Detection and Supervision in the Chemical Process Industries, Newcastle, 1995

35. R.P. Leger, Wm.J. Garland, W.F.S. Poehlman, "*Fault Detection and Diagnosis Using Statistical Control Charts and Artificial Neural Networks*", Artificial Intelligence in Engineering, Vol. 12, pp. 35-47, 1998

36. R.J. Patton, J. Chen, "*Observer-Based Fault Detection and Isolation: Robustness and Applications*", Control Eng. Practice, Vol. 5, No. 5, pp. 671-682, 1997

37. R. Isermann, "*Supervision, Fault-Detection and Fault-Diagnosis Methods – An Introduction*", Control Eng. Practice, Vol. 5, No. 5, pp. 639-652, 1997

38. J. Reifman, "*Survey of Artificial Intelligence Methods for Detection and Identification of Component Faults in Nuclear Power Plants*", Nuclear Technology, Vol. 119, July 1997

39. A.S. Erbay, B.R. Upadhyaya, "*A Personal Computer-Based On-Line Signal Validation System for Nuclear Power Plants*", Nuclear Technology, Vol. 119, July 1997

40. O.Glockler, M.V. Tulett, "*Reactor Noise Measurements at Pickering-B Nuclear Generating Station of Ontario Hydro*", 9th Power Plant Dynamics, Control and Testing Symposium Proceedings, Vol.2, pp. 89.01- 89.15, 1995

41. Wm.J. Garland, personal communication, May 1997

42. Wm.J. Garland, "Nuclear Reactor Safety Design", Course Notes for Thailand Initiative, February, 1998

43. H.W. Hinds, "*On-Line Assessment of Safety-System Transmitter Accuracy*", Proceedings from the COG CANDU Systems & Surveillance Programs Workshop, November, 1996

44. National Instruments Corp., "*LabVIEW User Manual for Windows*", September 1994

45. H.W. Hinds, R. MacKay, "*Evaluation of CANDU Safety-System Calibration Accuracy Through Monitoring*", Proceedings form the CNS CANDU Maintenance Conference, November, 1995

46. Hinds H.W., Personal Communication, Jan. 30, 1997

47. Wm.J. Garland, W.F.S. Poehlman, et al., "*Proceedings of the Workshop on Performance Support Systems*", Performance Support Systems Group, Department of Engineering Physics, Department of Computer Science & Systems, McMaster University, Hamilton, ON June 15-17th, 1994

48. S. Wold, S. Hellberg, T. Lundstedt, M. Stostrom, "*PLS Modeling With Latent Variables in Two or More Dimensions*", Frankfurt PLS-Meeting, Version 2.1, September 1987

49. S. Wold, N. Kettaneh, K. Tjessum, "*Hierarchical Multiblock PLS and PC Models For Easier Model Interpretation and As An Alternative to Variable Selection*", Journal of Chemometrics, Vol. 10, 463-482, 1996

50. J.A. Westerhuis, T. Kourti, J.F. MacGregor, "*On The Use of Multiblock and Hierarchical PCA and PLS Models*", submitted to the Journal of Chemometrics, January 1998

51. R.P. Leger, Wm.J. Garland, J. Popovic, C. Bailey, H.W. Hinds, "*Instrumentation Monitoring Using Multivariate Statistical Projection Methods*", Proceedings of the 1997 CNS Conference, Toronto, June 1997.

52. H.W. Hinds, Private Communication, Dec. 18, 1996

53. AECL Safety Report (for plant used in study), Chapter 1: Site Evaluation, Section 1 Introduction, Reissued June 1996

54. A. Natalizio, "*CANDU 600 Overview*", IAEA Training Course on Safety Review and Assessment for Construction Permit, Lecture L3.7, Ankara, Turkey, Sept.9 – Oct.4, 1985

55. P. DeTina, "*A Cognitive Framework to Improve Human-Computer Interaction of the OPUS Event Generator*", A Thesis for the Degree of Master of Engineering, McMaster University, March 1995

56. W.A. Shewhart, "*Economic Control of Quality of Manufacturing Product*", Van Nostrand, Princeton, N.J., 1931

APPENDIX A

C and MATLAB Functions Used for Decoding Project Data

A1.1 C Function for Decoding Raw Data

LabView saves the data as 16-bit integers which contain two bytes, a High byte and a Low byte. The high and low bytes are combined into pairs to from words. The date and time was saved as unsigned integers while all the process variables were saved as signed integers. The data file structure for one time stamp is as follows:

Date -	2 words	4 bytes
Time -	2 words	4 bytes
Ch D -	14 words (13 variables + 1 check sum)	28 bytes
Time -	2 words	4 bytes
Ch E -	14 words (13 variables + 1 check sum)	28 bytes
Time -	2 words	4 bytes
Ch F -	14 words (13 variables + 1 check sum)	28 bytes
Time -	2 words	4 bytes
Ch G -	40 words (39 variables + 1 check sum)	80 bytes
Time -	2 words	4 bytes
Ch H -	40 words (39 variables + 1 check sum)	80 bytes
Time -	2 words	4 bytes
Ch J -	40 words (39 variables + 1 check sum)	80 bytes
	TOTAL	**352 bytes/observation**

The check sum word is used in the data acquisition program to determine if the proper number of variables are present. It is not used in this research.

The standard formula used to decode the data is:

number = High Byte * 256 + Low Byte.

In order to decode the date and times, the following formulas were used:

$$\text{date or time} = \text{1st word} * 65536\ (2^{16}) + \text{2nd word}$$

When the data was decoded using C, a problem was discovered. When LabView saves a 16-bit integer, it saves it as High byte, Low byte. However, when C attempts to read a 16-bit integer, it attempts to read it as Low byte, High byte. For example, LabView saves the number 9502 as:

High Byte = 37

Low Byte = 30

Number = 37 * 256 + 30 = 9502

However, when C attempts to read the integer, it calculates:

Low Byte = 37

High Byte = 30

Number = 30*256 + 37 = 7717.

In order to use C to decode the data, all numbers had to be read as unsigned integers. Then the high and low bytes had to be calculated, reversed and recombined. In order to decode the number 9502 correctly, the following algorithm had to be used:

Number = 7717

Low Byte = Whole part of 7717/256 = 30.1445 = 30

High Byte = fractional part of 7717/256 * 256 = (30.1445 - 30) * 256 = 37

Number = High Byte * 256 + Low Byte = 37 * 256 + 30 = 9502.

Also, if the High byte for signed integers (measured variables) was greater than 127, 256 was subtracted from the High byte.

The following 6 basic steps were completed in the C decoding function. It should be noted that the C decoding function was used to filter out the setpoints contained in the SDS2 data.

For each file available:

1. Decode the time and date.
2. Decode the raw sensor data.
3. Check for irrational values and remove if found. This step will be expanded on in the next section.
4. Calculate the 15 minute averages when required.
5. Convent the raw data to engineering units by dividing by the appropriate B-scale supplied by AECL.
6. Save the 15 minute averages in engineering units in binary format to be read into MATLAB.

A complete copy of the source code used for the decoding can be found at the end of this appendix.

A1.2 MATLAB Function to Read Data Decoded by the C Function

The second part of the data decoding step was to read the 15 minute averages into MATLAB. The entire decoding task was not completed in MATLAB because the looping required to read each successive 2 second time observation resulted in excessive computational times. Reading the data from the C function into MATLAB was relatively straight forward because both programs save float data in binary format in the same manner. The m-file used to read the C data had three basic functions:

1. Read in all the 15 minute averages from the C output
2. Read in the number of points used in each 15-minute sensor average
3. Convert the flow measurements into engineering units.

There was no B-scale factor for the flowrates. The following formula was used to convert the raw integer data to engineering units:

$$\text{Flow} = 53.3333 * \sqrt{\frac{\text{raw data}}{20480} - 0.09} \quad \text{(Kg/sec)}$$

A complete copy of the m-file used to read the C data into MATLAB can be found at the end of this appendix.

A1.3 Resolution

One issue was raised regarding the resolution of the signals. The A to D converter used for the data acquisition was a +/- 10V, 12 bit converter. However, the voltage range for the signal was only 0.9 V - 4.5V. This resulted in a resolution of 1 part in 737 $\left[\frac{(4.5-0.9)V}{20V} * \left(2^{12}\right) = 737\right]$ as opposed to 1 part in 4096 $\left[\left(2^{12}\right) = 4096\right]$. This resolution resulted in a rather large quantization level for each of the measured process variables. The quantization levels and desired reference accuracies are shown in Table 2. The reference accuracies were specified by AECL. Typically, they are 1% of the full scale range for each of the measured process variables and represent the calibration drift which is to be detected.

Measured Variable	Units	Range	Quantization Level	Reference Accuracy
Log N Rate	(%/sec)	N/A	N/A	N/A
Log N	(decades)	N/A	N/A	N/A
HT Pressure Header 1	(MPa)	4 - 12	0.01	0.1
HT Pressure Header 2	(MPa)	4 - 12	0.01	0.1
Pressurizer Level	(m)	0 - 10.3	0.014	0.13
Boiler #2 Level	(m)	-5.2 - 2.3	0.01	0.1
Boiler #3 Level	(m)	-5.2 - 2.3	0.01	0.1
Boiler #6 Level	(m)	-5.2 - 2.3	0.01	0.1
Boiler #7 Level	(m)	-5.2 - 2.3	0.01	0.1
Boiler Feedline Pressure	(MPa)	2 - 7	0.0068	0.06
HT Flow 1	(Kg/sec)	0 - 32	0.027	0.51
HT Flow 2	(Kg/sec)	0 - 32	0.027	0.51
HDR 1-4 Differential Pressure	(MPa)	0 - 2	0.0027	0.02
HDR 2-3 Differential Pressure	(MPa)	0 - 2	0.0027	0.02
Moderator Temperature	(°C)	N/A	N/A	N/A

TABLE A1: Quantization Levels and Reference Accuracy's for Process Variables

The large quantization levels resulted in course signals for some of the variables. This was especially evident in the header pressure signals where the normal signals seemed to vary by only one quantization level. This course signal was a concern for the multivariate statistical techniques used for fault detection because they rely on the variance/covariance structure of the data set. The course signal may cause small variances which are too small to be of use in the analysis. This concern will be examined in the results section of the report.

A 1.4 C Code

```
/*************************************************************************
        C-code to read AECL data

        Code uses matrices indices starting at 0
        Use variable numbers on AECL B scale sheet


*************************************************************************/
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "bscale.h"
#define GetMemory(num, type)  (type *)malloc((num) * sizeof(type))

main()
{
float rem, raw[1][176];
float data[1][75];
int hibyte, i, points[75], obs, avecounter, file_count, var_index;
long ptr, date_time[4], curtime, curdate, prevtime, starttime;
unsigned int *cbyte, lowbyte;

FILE *fp, *out_fp, *pts_fp, *sum_fp;
char op_file[50]="c:\\robnew\\readaecl\\outsept.dat";
char pts_file[50]="c:\\robnew\\readaecl\\ptssept.dat";
char sum_file[50]="c:\\robnew\\readaecl\\sumsept.dat";

cbyte = GetMemory(1,unsigned int);
if (cbyte == NULL)
{
        printf("Not enough memory for cbyte");
        exit(1);
}
// Initializations
//prevtime=151501; // Initial time for start of calibration disk
//prevtime=140721; // Initial time for start of March/96 data disk
prevtime=94036; // Initial time for start of Sept/96 data disk
// prevtime=5729;
starttime=prevtime;
avecounter=0;

for (i=0; i<75; i++)
```

```
{
        data[0][i]=0.0;
        points[i]=0;
}

for (file_count=0;file_count<352;file_count ++) // file_count goes from 0 - 359
{
        if ( (fp = fopen(file_names[file_count], "rb")) ==NULL)
                printf("Could not open file");
        printf(" file name %d = %s \n", file_count, file_names[file_count]);

obs=0;

while ( !feof(fp) )
{

        for (i=0; i<4; i++)
        {
                ptr= (long)352*(obs) + (i*2);
                fseek(fp, ptr, SEEK_SET);
                fread(cbyte, sizeof(unsigned int), 1, fp);
                lowbyte=*cbyte/256;
                rem=*cbyte;
                rem=(rem/256 -lowbyte)*256;
                hibyte=rem;
                date_time[i]=256.0*hibyte + lowbyte;
        } // end of loop for reading date and time

        if ( !feof(fp) ) // Check for end of file
        {
        curtime=(long)date_time[2]*65536 + date_time[3];
//      printf(" curtime %ld \n",curtime); //Uncomment to find starting time
//      break;
        curdate=(long)date_time[0]*65536 + date_time[1];

        for (i=4; i<176; i++)
        {
                ptr=(long)352*(obs) + (i*2);
                fseek(fp, ptr, SEEK_SET);
                fread(cbyte, sizeof(unsigned int), 1, fp);
                lowbyte=*cbyte/256;
                rem=*cbyte;
                rem=(rem/256 -lowbyte)*256;
                hibyte=rem;
```

```
        if(hibyte >127)
                hibyte=hibyte-256;
        raw[0][i]=256.0*hibyte + lowbyte;

} // End of loop for reading variables

// Check for irrational data;
for (i=0; i<75; i++)
{
        var_index=varindex[i];
        if ((raw[0][var_index])/bscale[var_index]==-32.0 ||
                (raw[0][var_index])/bscale[var_index]==-16.0)
        {
                printf(" curtime %ld \n",curtime);
                printf("Irration value found @ %d \n",var_index);
        }
        else
        {
                data[0][i]=data[0][i] + raw[0][var_index];
                points[i]=points[i]+1;
        }

}

// Calculate time for averaging
if (fmod(curtime,100) >= fmod(prevtime,100))
{
        avecounter=avecounter + (fmod(curtime,100) - fmod(prevtime,100));
}
else
        avecounter=avecounter + ((fmod(curtime,100)+60) -fmod(prevtime,100));

// Average loop
if (avecounter >= 900)
{
        for (i=0; i<75; i++)
        {
                var_index=varindex[i];
                if (points[i]> 0)
                {
                        data[0][i]=(data[0][i]/points[i])/bscale[var_index];
                }
                else
                {
```

```
                        data[0][i]=-32.0;
                    }
                }

                    // write average, points summary to files
                    if ( (out_fp = fopen(op_file, "ab")) ==NULL)
                        printf("Could not open output file");
                    if( (fwrite(data, sizeof(float), 75, out_fp)) != 75)
                        printf("Error writing to file");
                    fclose(out_fp);

                    if ( (pts_fp = fopen(pts_file, "ab")) ==NULL)
                        printf("Could not open output file");
                    if( (fwrite(points, sizeof(int), 75, pts_fp)) != 75)
                        printf("Error writing to file");
                    fclose(pts_fp);

                    if ((sum_fp = fopen(sum_file, "a")) == NULL)
                        printf("Could not open points file");
                    fprintf(sum_fp, "date start finish avecounter %ld %ld %ld %d \n",
                        curdate, starttime, curtime, avecounter);
                    fclose(sum_fp);

                    printf("date start finish avecounter %ld %ld %ld %d \n", curdate,
                        starttime, curtime, avecounter);
                    for (i=0; i<75; i++)
                    {
                        data[0][i]=0.0;
                        points[i]=0;
                    }
                    avecounter = 0;
                    starttime=curtime;
        } // End average loop

        obs=obs+1;
        prevtime=curtime;
        }// Close IF for checking for end of file

}// End While loop for reading one data file

fclose(fp);
}// End for loop for incrementing data file name

}// End Main loop
```

A 1.5 MATLAB Code

function [blkdata, numpts]=rdcdatr1();

```
% Function to read the averaged C data

fid=fopen('c:\robnew\readaecl\outnov.dat', 'rb', 'n')

alldat=fread(fid, [75,1700], 'float');
alldat=alldat';

fclose(fid);

fid=fopen('c:\robnew\readaecl\ptsnov.dat', 'rb', 'n')
numpts=fread(fid, [75,1700], 'int16', 'n');
numpts=numpts';

fclose(fid);

alldat(:,5)=53.3333*sqrt((alldat(:,5)/20480) - 0.09);
alldat(:,6)=53.3333*sqrt((alldat(:,6)/20480) - 0.09);

alldat(:,18)=53.3333*sqrt((alldat(:,18)/20480) - 0.09);
alldat(:,19)=53.3333*sqrt((alldat(:,19)/20480) - 0.09);

alldat(:,31)=53.3333*sqrt((alldat(:,31)/20480) - 0.09);
alldat(:,32)=53.3333*sqrt((alldat(:,32)/20480) - 0.09);

% Block Variables

rate=[alldat(:,1) alldat(:,14) alldat(:,27) alldat(:,40) alldat(:,52) alldat(:,64)];
logn=[alldat(:,2) alldat(:,15) alldat(:,28) alldat(:,41) alldat(:,53) alldat(:,65)];

head1=[alldat(:,3) alldat(:,16) alldat(:,29) alldat(:,42) alldat(:,54) alldat(:,66)];
head2=[alldat(:,4) alldat(:,17) alldat(:,30) alldat(:,43) alldat(:,55) alldat(:,67)];

flow1=[alldat(:,5) alldat(:,18) alldat(:,31)];
flow2=[alldat(:,6) alldat(:,19) alldat(:,32)];

dp14=[alldat(:,44) alldat(:,56) alldat(:,68)];
dp23=[alldat(:,45) alldat(:,57) alldat(:,69)];
```

```
plev=[alldat(:,7) alldat(:,20) alldat(:,33) alldat(:,46) alldat(:,58) alldat(:,70)];

b2=[alldat(:,8) alldat(:,21) alldat(:,34) alldat(:,47) alldat(:,59) alldat(:,71)];
b3=[alldat(:,9) alldat(:,22) alldat(:,35) alldat(:,48) alldat(:,60) alldat(:,72)];
b6=[alldat(:,10) alldat(:,23) alldat(:,36) alldat(:,49) alldat(:,61) alldat(:,73)];
b7=[alldat(:,11) alldat(:,24) alldat(:,37) alldat(:,50) alldat(:,62) alldat(:,74)];

fdline=[alldat(:,12) alldat(:,25) alldat(:,38) alldat(:,51) alldat(:,63) alldat(:,75)];

modt=[alldat(:,13) alldat(:,26) alldat(:,39)];

blkdata=[rate logn head1 head2 flow1 flow2 dp14 dp23 plev b2 b3 b6 b7 fdline modt];


function [data2]=block(data1);

% Function to block data for analysis

% DATA 1 is of the form:      DATA 2 is of the form:
% 1-6   Rate                      1-6 Head1
% 7-12  Log N                     7-12 Head2
% 13-18 Head1                     13-18 PLev
% 19-24 Head2                     19-24 Log N
% 25-27 Flow1                     25-30 B2
% 28-30 Flow2                     31-36 B3
% 31-33 DP1-4                     37-42 B6
% 34-36 DP2-3                     43-48 B7
% 37-42 PLev                      49-54 FdLin
% 43-48 B2                        55-57 Flow1
% 49-54 B3                        58-60 Flow2
% 55-60 B6                        61-63 DP1-4
% 61-66 B7                        64-66 DP2-3
% 67-72 FdLin                     67-72 Rate
% 73-75 ModT                      73-75 ModT

data2=data1(:,13:24);

data2=[data2 data1(:,37:42) data1(:,7:12) data1(:,43:66) data1(:,67:72) data1(:,25:36)
data1(:,1:6) data1(:,73:75)];
```

APPENDIX B

PCA and Multi-Block PCA Codes from Literature

B 1.1 PCA

Transform, center and scale the variables
For each dimension
    Choose start t
    Loop until convergence of t

$$p = \frac{X^T \cdot t}{t^T \cdot t}$$ % Calculate X loadings

normalize p to $\|p\| = 1.0$

$$t = \frac{X \cdot p}{p^T \cdot p}$$ % Calculate X scores

    end
$X = X - t \cdot p^T$ % Deflation
end

B 1.2 CPCA [48]

Start with a guessed t-vector, for instance the column in any Y_b-matrix with the largest variance

(1) $$q_b^T = \frac{t^T \cdot Y_b}{t^T \cdot t}$$

(2) Check convergence on t analogously to PCA.
If convergence, step (7), else step (3)

(3) $$u_b = \frac{Y_b \cdot q_b}{q_b^T \cdot q_b}$$

(4) Collect all u_b vectors (b=1,2,3, ... , B) in U

(5) $$w^T = \frac{t^T \cdot U}{t^T \cdot t}$$

Norm to $\|w\| = 1.0$

(6) $t = U \cdot w$
Return to step (1)

---

(7) Residuals $F_b = Y_b - t \cdot q_b^T$
Use the F_b matrices as Y_b in next model dimension

B 1.3 H-PCA [49]

1. Transform, center and scale the data appropriately.
2. The sequence of steps 3-9 is run through for each model dimension. We start with the model dimension index adim =1.
3. Use one of the X-columns, e.g. the one with the largest variance, as starting vector for the X-super-score t. Normalize t to unit length 1.0: $\|t\| = 1.0$.
Set iteration counter niter = 1.
Set t-difference to an arbitrary large value diff=100.
4. Save old t for test of convergence
t_o=t
5. Block loadings q. With missing data, the ordinary NIPALS modification is used, making the summations only over the 'present' elements in x_k and the corresponding elements in t^Tt.

$$p_k^T = \frac{t^T \cdot X_k}{t^T \cdot t}$$

6. Block scores r_b. With missing data, again the NIPALS modification is used (see step 5)

$$r_b = \frac{dX_b p_b}{K_b}$$

The modifier d normally equals 1.0 but can for instance be set to the value below to weight large blocks
$d = 1 + 0.5 \log_{10} K_b$
7. Check convergence; stop if reached. Criterion is typically 10-8. Nitermax is typically 200.

$$d = \frac{diff^T \cdot diff}{t^T \cdot t}$$

if d < criterion or iter > nitermax
    break and go to step 9;
end;
8. One PC round on the super-level. R is the matrix of X-block scores.

(a) $$w^T = \frac{t^T \cdot R}{t^T \cdot t}$$

(b) $$t = R \cdot w$$

if admin > 1 (second and higher model dimensions)
    $t = t - [T(T^T \cdot t)]$; (here T is matrix of super-scores up to admin-1)
    correct w so that t=Rw, i.e. $w = (R^TR)^{-1}R^Tt$
end;
    $d = \|t\|$
    t=t/d
    w=w/d

(c) diff=t-t_{old}
(d) Update iteration counter, return to step 5.
Niter=niter+1

9. Save result vectors in various matrices. Residuals for next dimension. For the vectors x_k:

$$x_k = x_k - t \cdot p_k^T$$

10. If warranted, continue with the next dimension. Then return to step 3.

B 1.4 Consensus PCA (CPCA) [50]

Transform, center and scale
For each dimension
Choose start t_T
Loop until convergence of t_T

$$p_p = \frac{X_b^T \cdot t_T}{t_T^T \cdot t_T}$$ % X_b block variable loadings

normalize p_b to $\|p_b\| = 1.0$

$$t_b = \frac{X_b \cdot p_b}{m_{xb}}$$ % X_b block scores (block scaling)

$$T = [t_1 \ldots t_b]$$ % Combine all block scores in T

$$w_T = \frac{T^T \cdot t_T}{t_T^T \cdot t_T}$$ % Super weight

normalize w_T to $\|w_T\| = 1.0$

$$t_T = T \cdot w_T$$ % Super score

end

$$p_b = \frac{X_b^T \cdot t_T}{t_T^T \cdot t_T}$$

$$X_b = X_b - t_T \cdot p_b^T$$ % Deflation

end

B 1.5 Hierarchical PCA (HPCA) [50]

Transform, center and scale
For each dimension
Choose start t_T
Loop
Normalize t_T to $\|t_T\| = 1.0$

$$p_p = \frac{X_b^T \cdot t_T}{t_T^T \cdot t_T}$$ % X_b block variable loadings

Break if convergence of t_T

$t_b = X_b \cdot p_b$ % X_b block scores

normalize t_b to $\|t_T\| = 1.0$ % normalize t_b

$T = [t1 \ldots tB]$ % Combine all block scores in T

$$w_T = \frac{T^T \cdot t_T}{t_T^T \cdot t_T}$$ % Super weight

$t_T = T \cdot w_T$ % Super score

end

$X_b = X_b - t_T \cdot p_b^T$ % Deflation

end

B 1.6 Predictions for CPCA [48]

Score values for a new observation vector with data y_b^T are obtained as follows (the block loops are not explicitly shown):

For a=1 to A % dimension loop
Begin

$$(u_b) = \frac{y_b^T \cdot q_b}{q_b^T \cdot q_b}$$ % block scores

collect all u_b into the vector u_T

$(t_a) = u^T \cdot w$ % score

$f_b = y_b - (t_a)q_b$ % residuals

$y_b = f_b$ % next dim.

End

B 1.7 Prediction for H-PCA [49]

(1) We come in with a new data vector (row vector, appropriately transformed, centered and scaled) on the X-side, denoted by z_T.

(2) The block loadings P_b are used to compute the X-block score values (dimension one) for the new vector.

(3) These block score values are then used to calculate the super-X-score t_{new}.

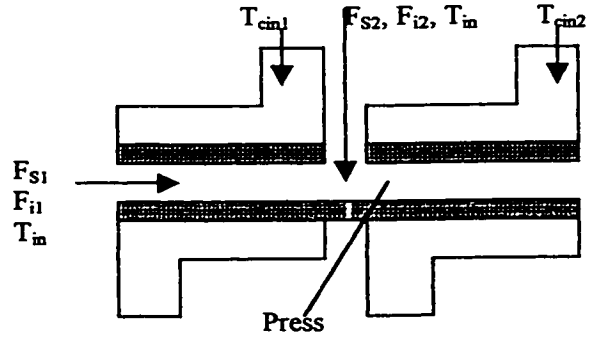
(4) Residuals of z_T are calculated $(z_k - t_{new} \cdot p_k^T)$, and then used analogously for dimension two, etc. until X-block and X-super-score values have been calculated for all model dimensions.

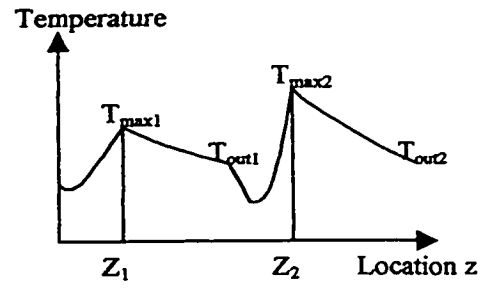
(5) From the residuals after the last model dimension, block residuals standard deviations and overall residual standard deviations can be calculated and compared with the 'normal' values from the model estimation.

APPENDIX C

Low-density Polyethylene (LDPE) Test Results

The following tests were conducted on the LDPE test data set contained in the MacStat* tutorial. Below is a description of the process and the variables in the data set.





Section 1		Section 2	
x_1	Inlet Temp. of Reaction Mixture, K (T_{in})	x_9	Inlet Temp. of Reaction Mixture, K (T_{in})
x_2	Initiator Flowrate, g/s (F_{i1})	x_{10}	Initiator Flowrate, g/s (F_{i2})
x_3	Flow of Solvent to Reactor, (g/s) (F_{s1})	x_{11}	Flow of Solvent to Reactor, (g/s) (F_{s2})
x_4	Inlet Temp. of Coolant, K (T_{cin1})	x_{12}	(Inlet Temp. of Coolant, K (T_{cin2})
x_5	Pos. of Max. Temp., % of Reac. Len. (z_1)	x_{13}	Pos. of Max. Temp., % of Reac. Len. (z_2)
x_6	Max. Temp. of Reaction Mix., K (T_{max1})	x_{14}	Max. Temp. of Reaction Mix. K (T_{max2})
x_7	Outlet Temp. K (T_{out1})	x_{15}	Outlet Temp. K (T_{out2})
x_8	Pressure of the Reactor, atm (Press)	x_{16}	Pressure of the Reactor, atm (Press)

A simple PCA analysis was completed with the following results:

1. on entire 16 variables (xmod): SSexp = 27.5%
2. on 1st 8 variables (X1): SSexp = 39.5%
3. on 2nd 8 variables (X2): SSexp = 41.8%

* MacStat is a multivariate SPC computer code developed by the Chemical Engineering Department of McMaster University

Using the HPCA algorithm from Wold, the following analysis was completed [49]

t_{Start}	SSexp X1	SSexp X2	t_{Start}	SSexp X1	SSexp X2
x_1	39.2%	8.38%	x_9	39.2%	8.38%
x_2	39.2%	8.38%	x_{10}	6.3%	41.7%
x_3	39.2%	8.38%	x_{11}	6.3%	41.7%
x_4	39.2%	8.38%	x_{12}	6.3%	41.7%
x_5	39.2%	8.38%	x_{13}	6.3%	41.7%
x_6	39.2%	8.38%	x_{14}	6.3%	41.7%
x_7	39.2%	8.38%	x_{15}	6.3%	41.7%
x_8	6.3%	41.7%	x_{16}	6.3%	41.7%

Table C.1 – Results from HCPA Analysis

The following highlights were noted from the above analysis:

1. Norming t_c; also found norming v (super weights) would not work

2. Not norming or scaling t_b's

3. deflating using t_c

Based on the results from Table C.1, it was concluded there was two solutions.

Solution #1

SSexp X1 = 39.2% ⇐ Same as PCA solution for X1
SSexp X2 = 8.38%

$$\begin{pmatrix} v_1 = 0.0497 \\ v_2 = 0.0106 \end{pmatrix}$$

tc is an eigenvector of $0.0497 \cdot \left(X_1 X_1^T\right) + 0.0106 \cdot \left(X_2 X_2^T\right)$
The associated eigenvalue is equal to 8.0

Solution #2

SSexp X1 = 6.3%
SSexp X2 = 41.7% ⇐ Same as PCA solution for X2

$$\begin{pmatrix} v_1 = 0.0072 \\ v_2 = 0.0478 \end{pmatrix}$$

t_c is an eigenvector of $0.0072 \cdot (X_1 X_1^T) + 0.0478 \cdot (X_2 X_2^T)$
The associated eigenvalue is equal to 8.0

It should be noted that in the above analysis, t_c is an eigenvector of $c\sum_{b=1}^{B} v_b X_b X_b^T$.

In the standard PCA analysis, t is an eigenvector of XX^T. For PCA, the associated eigenvalue of the first score, t, is equal to 215.95.

The following starting points for t_c were also tried:

1. all values in t_c were started at 1 $\Rightarrow$ Solution #1 was obtained

2. random guesses of t_c were used, similar to neural network training

Generally, it was found that the t_b with the largest variance dictated the solution. For example, if t_{b1} had the largest variance, Solution #1 was obtained.

If both the t_b's and t_c were normed, the following solution was found:

SSexp X1 = 26.9%
SSexp X2 = 28.1%
> Average SSexp = 27.5%, which was the same as PCA

$$\begin{pmatrix} v_1 = 0.5832 \\ v_2 = 0.5832 \end{pmatrix}$$

The same result was obtained for any starting guess for t_c.

In this case, t_c is an eigenvalue of $0.5832 \cdot (X_1 X_1^T) + 0.5832 \cdot (X_2 X_2^T)$. The associated eigenvalue is equal to 125.94.

Finally, as stated above, the algorithm deflated using t_c. If the tb's were used for deflation, the sum of squares explained were as follows:

SSexp X1 = 35.9%
SSexp X2 = 37.5%

APPENDIX D

Codes for Multi-Block, Multi-Level PCA Model Development and Prediction

D 1.1 Code for Multi-Block, Multi-Level PCA Model Development

```
function [tcall,wall,palllev1,norm_plev1,talllev1,norm_tlev1,...
        palllev2,norm_plev2,talllev2,norm_tlev2,...
        SPElev1,SPElev2,SPElev3,x_mean,x_weig,x_mod_sca,x,...
        ssexpxlev0all,ssexpxlev1all,ssexpxlev2all,ssexpxlev3all]...
        =cpca3lv4(x);

% Code to develop CPCA model
% Uses routines from MACSTAT
% Look for missing data
[x,missx,misflagx]=missing(x);
% Mean Center and Auto Center x
[x_mean, x]=mcenter(x,missx,misflagx);
[x_weig,x,missx,misflagx]=mauto(x,missx,misflagx);

% Scale by the desired offset errors
%w=[20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 14.3 14.3 14.3 14.3
14.3 14.3 ...
%   14.3 14.3 14.3 14.3 14.3 14.3 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 ...
%   20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 29.4 29.4 29.4 29.4 29.4 29.4 7.1 7.1 7.1
7.1 ...
%   7.1 7.1 74.1 74.1 74.1 74.1 74.1 74.1 8.5 8.5 8.5 8.5 8.5 8.5 0.19 0.19 0.19];
%x = missx.*((ones(905,1)*w).*x);

x_mod_sca=x;
% Do CPCA calculation
% My code

% INPUT DATA
noblklev1=15
noblklev2=8
%varperblk=3.0;
```

```
admin=1; % Number of PC's
type=2; % Type of analysis; 1=norm scores, 2=norm loadings
deflate=1; % Method of deflation; 1=Level 1, 2=Level 2, 3=Level 3

[numobs,numvar]=size(x);

%for i=1:noblklev1
% mlev1(i)=varperblk;
%end

mlev1(1)=6; mlev1(2)=6; mlev1(3)=6; mlev1(4)=6; mlev1(5)=6;
mlev1(6)=6; mlev1(7)=6; mlev1(8)=6; mlev1(9)=6; mlev1(10)=3;
mlev1(11)=3; mlev1(12)=3; mlev1(13)=3; mlev1(14)=6; mlev1(15)=3;

mlev2(1)=2; mlev2(2)=2; mlev2(3)=4; mlev2(4)=1;
mlev2(5)=2; mlev2(6)=2; mlev2(7)=1; mlev2(8)=1;

% Divisions for SS cal for Level 2
mlev2ss(1)=12; mlev2ss(2)=12; mlev2ss(3)=24; mlev2ss(4)=6;
mlev2ss(5)=6; mlev2ss(6)=6; mlev2ss(7)=6; mlev2ss(8)=3;

% Calculate original SS
% Level 0
for i= 1:numvar
 ssxolev0(i) = sum(sum(x(:,i).^2));
end

% Level 1
mm = 1;
for i = 1:noblklev1
 if i > 1;
  mm=sum(mlev1(1:i-1))+1;
 end;
 mmm = sum(mlev1(1:i));
 ssxolev1(i) = sum(sum(x(:,mm:mmm).^2));
end;

% Level 2
mm = 1;
for i = 1:noblklev2
 if i > 1;
  mm=sum(mlev2ss(1:i-1))+1;
 end;
 mmm = sum(mlev2ss(1:i));
 ssxolev2(i) = sum(sum(x(:,mm:mmm).^2));
end;
```

```
% Level 3
ssxolev3 = sum(sum(x.^2));

% Start Calculations
for numpc=1:admin

 tc=ones(numobs,1);
 if type == 1;
  tc=tc/norm(tc);
 end;

 diff=ones(numobs,1);
 diff=diff*100.0;

 for it=1:300
  it
  plev1=[];
  plev2=[];

% Level 1 Calculations
  mm = 1;
  for i = 1:noblklev1
   if i > 1;
    mm=sum(mlev1(1:i-1))+1;
   end;
   mmm = sum(mlev1(1:i));
   pplev1=(x(:,mm:mmm)'*tc)/(tc'*tc);
   plev1=[plev1;pplev1];
  end;

% Check for convergence
  tc_check=(diff'*diff)/(tc'*tc)
  if tc_check < 1e-9 | it==250
% Update Level 2 Loadings
   mm = 1;
   for i = 1:noblklev2
    if i > 1;
     mm=sum(mlev2(1:i-1))+1;
    end;
    mmm = sum(mlev2(1:i));
    pplev2=(xlev2(:,mm:mmm)'*tc)/(tc'*tc);
    plev2=[plev2;pplev2];
   end;
   break
  end
```

```
  mm = 1;
  for i = 1:noblklev1
   if i > 1;
    mm=sum(mlev1(1:i-1))+1;
   end;
   mmm = sum(mlev1(1:i));
   if type == 2;
    norm_plev1(i,numpc)=norm(plev1(mm:mmm));
    plev1(mm:mmm)=plev1(mm:mmm)/norm_plev1(i,numpc);
   end;
%    tlev1(:,i)=(x(:,mm:mmm)*plev1(mm:mmm))/mlev1(i);
   tlev1(:,i)=(x(:,mm:mmm)*plev1(mm:mmm));

   if type == 1;
    norm_tlev1(i,numpc)=norm(tlev1(:,i));
    tlev1(:,i)=tlev1(:,i)/norm(tlev1(:,i));
   end
  end;

  xlev2=tlev1;

% Level 2 Calculations
  mm = 1;
  for i = 1:noblklev2
   if i > 1;
    mm=sum(mlev2(1:i-1))+1;
   end;
   mmm = sum(mlev2(1:i));
   pplev2=(xlev2(:,mm:mmm)'*tc)/(tc'*tc);
   plev2=[plev2;pplev2];
  end;

  mm = 1;
  for i = 1:noblklev2
   if i > 1;
    mm=sum(mlev2(1:i-1))+1;
   end;
   mmm = sum(mlev2(1:i));
   if type == 2;
    norm_plev2(i,numpc)=norm(plev2(mm:mmm));
    plev2(mm:mmm)=plev2(mm:mmm)/norm_plev2(i,numpc);
   end;
%    tlev2(:,i)=(xlev2(:,mm:mmm)*plev2(mm:mmm))/mlev2(i);
   tlev2(:,i)=(xlev2(:,mm:mmm)*plev2(mm:mmm));
   if type == 1;
```

```
      norm_tlev2(i,numpc)=norm(tlev2(:,i));
      tlev2(:,i)=tlev2(:,i)/norm(tlev2(:,i));
     end
    end;

% Level 3 Calculations
    T=tlev2;

    w=(T'*tc)/(tc'*tc);

    if type == 2;
     norm_w=norm(w);
     w=w/norm_w;
    end;

    tc_old=tc;

    tc=T*w;

    if type == 1;
     norm_tc=norm(tc);
     tc=tc/norm_tc;
     w=w/norm_tc;
    end

    diff=tc-tc_old;

   end   % END OF Tc CONVERGENCE LOOP


   mm=1
  %Deflation
   for i = 1:noblklev1
    if i > 1;
     mm=sum(mlev1(1:i-1))+1;
    end;
    mmm = sum(mlev1(1:i));
    if deflate == 1
     x(:,mm:mmm)=x(:,mm:mmm)-tlev1(:,i)*plev1(mm:mmm)';
    end

    if deflate == 2
     if i==1 | i==2
      x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,1)*plev1(mm:mmm)';
     end
     if i==3 | i==4
```

```
    x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,2)*plev1(mm:mmm)';
   end
   if i==5 | i==6 | i==7 | i==8
    x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,3)*plev1(mm:mmm)';
   end
   if i==9
    x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,4)*plev1(mm:mmm)';
   end
   if i==10 | i==11
    x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,5)*plev1(mm:mmm)';
   end
   if i==12 | i==13
    x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,6)*plev1(mm:mmm)';
   end
   if i==14
    x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,7)*plev1(mm:mmm)';
   end
   if i==15
    x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,8)*plev1(mm:mmm)';
   end
  end

  if deflate == 3
   x(:,mm:mmm)=x(:,mm:mmm)-tc*plev1(mm:mmm)';
  end
 end;

% SS Calculations
% Level 0
for i=1:numvar
 ssxnlev0(i)=sum(sum(x(:,i).^2));
 ssexpxlev0(i)=(ssxolev0(i)-ssxnlev0(i))/ssxolev0(i);
end

% Level 1
 mm=1
 for i = 1:noblklev1
  if i > 1;
   mm=sum(mlev1(1:i-1))+1;
  end;
  mmm = sum(mlev1(1:i));
  ssxnlev1(i)=sum(sum(x(:,mm:mmm).^2));
  ssexpxlev1(i)=(ssxolev1(i)-ssxnlev1(i))/ssxolev1(i);
%   ssxo(i)=ssxn(i);
 end;
```

```
% Level 2
 mm=1
 for i = 1:noblklev2
  if i > 1;
   mm=sum(mlev2ss(1:i-1))+1;
  end;
  mmm = sum(mlev2ss(1:i));
  ssxnlev2(i)=sum(sum(x(:,mm:mmm).^2));
  ssexpxlev2(i)=(ssxolev2(i)-ssxnlev2(i))/ssxolev2(i);
%   ssxo(i)=ssxn(i);
 end;
% Level 3
 ssxnlev3=sum(sum(x.^2));
 ssexpxlev3=(ssxolev3-ssxnlev3)/ssxolev3;


% Print out Results
 ssexpxlev1'
 ssexpxlev2'
 ssexpxlev3'

% Save loadings, scores and SSexp
% Level 0
 if numpc == 1
  ssexpxlev0all=ssexpxlev0';
 else
  ssexpxlev0all=[ssexpxlev0all ssexpxlev0'];
 end

% Level 1
 if numpc == 1
  palllev1=plev1;
  talllev1=tlev1;
  ssexpxlev1all=ssexpxlev1';
 else
  palllev1=[palllev1 plev1];
  talllev1=[talllev1 tlev1];
  ssexpxlev1all=[ssexpxlev1all ssexpxlev1'];
 end

% Level 2
 if numpc == 1
  palllev2=plev2;
  talllev2=tlev2;
  ssexpxlev2all=ssexpxlev2';
 else
```

```
   palllev2=[palllev2 plev2];
   talllev2=[talllev2 tlev2];
   ssexpxlev2all=[ssexpxlev2all ssexpxlev2'];
  end

% Level 3
  if numpc ==1
   tcall=tc;
   wall=w;
   ssexpxlev3all=ssexpxlev3';
  else
   tcall=[tcall tc];
   wall=[wall w];
   ssexpxlev3all=[ssexpxlev3all ssexpxlev3'];
  end

% SPE Calculations
% Level 1
  mm=1
  for i = 1:noblklev1
   if i > 1;
    mm=sum(mlev1(1:i-1))+1;
   end;
   mmm = sum(mlev1(1:i));
   SPElev1(:,i)=(sum((x(:,mm:mmm)').^2))';
  end;

% Level 2
  mm=1
  for i = 1:noblklev2
   if i > 1;
    mm=sum(mlev2ss(1:i-1))+1;
   end;
   mmm = sum(mlev2ss(1:i));
   SPElev2(:,i)=(sum((x(:,mm:mmm)').^2))';
  end;

 SPElev3=(sum((x').^2))';

 %SPEmod(:,numpc)=SPE;

 end;  % END OF PC LOOP
```

D 1.2 Code for Multi-Block, Multi-Level PCA Prediction

```
function [x_org_pre,x,SPElev0pre,SPElev1pre,SPElev2pre,SPElev3pre,SPEpre,...
```

```
    talllev1pre, talllev2pre, tcallpre]=...
    precpc31(x,palllev1,palllev2,wall,x_mean,x_weig,...
    norm_tlev1,norm_tlev2,norm_plev1,norm_plev2);

% Code to do predictions from CPCA model
% Uses routines from MACSTAT

% Look for missing data
[x,missx,misflagx]=missing(x);

% Mean Center and Auto Center x

[row,col]=size(x);
for i=1:row
  x(i,:)=x(i,:)-x_mean;
end

%Supply own scaling
%x_weig=[20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 14.3 14.3 14.3
14.3 14.3 14.3 ...
%   14.3 14.3 14.3 14.3 14.3 14.3 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 ...
%   20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 29.4 29.4 29.4 29.4 29.4 29.4 7.1 7.1 7.1
7.1 ...
%   7.1 7.1 74.1 74.1 74.1 74.1 74.1 74.1 8.5 8.5 8.5 8.5 8.5 8.5 0.19 0.19 0.19];

for i=1:row
  x(i,:)=x(i,:).*x_weig;
end
x_org_pre=x;

%xhat=zeros(4,16);

% INPUT DATA
noblklev1=15
noblklev2=8
%varperblk=3.0;
admin=4;
type=2; % Type of analysis; 1=norm scores, 2=norm loadings
deflate=3; % Method of deflation; 1=Level 1, 2=Level 2, 3=Level 3


mlev1(1)=6; mlev1(2)=6; mlev1(3)=6; mlev1(4)=6; mlev1(5)=6;
mlev1(6)=6; mlev1(7)=6; mlev1(8)=6; mlev1(9)=6; mlev1(10)=3;
mlev1(11)=3; mlev1(12)=3; mlev1(13)=3; mlev1(14)=6; mlev1(15)=3;
```

```
mlev2(1)=2; mlev2(2)=2; mlev2(3)=4; mlev2(4)=1;
mlev2(5)=2; mlev2(6)=2; mlev2(7)=1;  mlev2(8)=1;

% Divisions for SS cal for Level 2
mlev2ss(1)=12; mlev2ss(2)=12; mlev2ss(3)=24; mlev2ss(4)=6;
mlev2ss(5)=6; mlev2ss(6)=6; mlev2ss(7)=6;  mlev2ss(8)=3;

% Calculate original SS
% Level 1
mm = 1;
for i = 1:noblklev1
 if i > 1;
  mm=sum(mlev1(1:i-1))+1;
 end;
 mmm = sum(mlev1(1:i));
 ssxolev1(i) = sum(sum(x(:,mm:mmm).^2));
end;

% Level 2
mm = 1;
for i = 1:noblklev2
 if i > 1;
  mm=sum(mlev2ss(1:i-1))+1;
 end;
 mmm = sum(mlev2ss(1:i));
 ssxolev2(i) = sum(sum(x(:,mm:mmm).^2));
end;

% Level 3
ssxolev3 = sum(sum(x.^2));

for numpc=1:admin

% Calculation of Level 1 Scores
 mm = 1;
 for i = 1:noblklev1
  if i > 1;
   mm=sum(mlev1(1:i-1))+1;
  end;
  mmm = sum(mlev1(1:i));
%   tlev1(:,i)=(x(:,mm:mmm)*palllev1(mm:mmm,numpc))/mlev1(i);
  tlev1(:,i)=(x(:,mm:mmm)*palllev1(mm:mmm,numpc));
 end;
%TEST
 if type ==1
  for i=1:noblklev1
```

```
    tlev1(:,i)=tlev1(:,i)/norm_tlev1(i,numpc);
   end
  end
%TEST
  if type == 2
   for i=1:noblklev1
    tlev1(:,i)=tlev1(:,i)/norm_plev1(i,numpc);
   end
  end

% Calculation of Level 2 Scores

  xlev2=tlev1;

  mm = 1;
  for i = 1:noblklev2
   if i > 1;
    mm=sum(mlev2(1:i-1))+1;
   end;
   mmm = sum(mlev2(1:i));
%   tlev2(:,i)=(xlev2(:,mm:mmm)*palllev2(mm:mmm,numpc))/mlev2(i);
   tlev2(:,i)=(xlev2(:,mm:mmm)*palllev2(mm:mmm,numpc));
  end;
%TEST
  if type ==1
   for i=1:noblklev2
    tlev2(:,i)=tlev2(:,i)/norm_tlev2(i,numpc);
   end
  end
%TEST
  if type == 2
   for i=1:noblklev2
    tlev2(:,i)=tlev2(:,i)/norm_plev2(i,numpc);
   end
  end


  tc=tlev2*wall(:,numpc);

 % Deflate and calculate prediction
  mm = 1;
  for i = 1:noblklev1
   if i > 1;
    mm=sum(mlev1(1:i-1))+1;
   end;
   mmm = sum(mlev1(1:i));
```

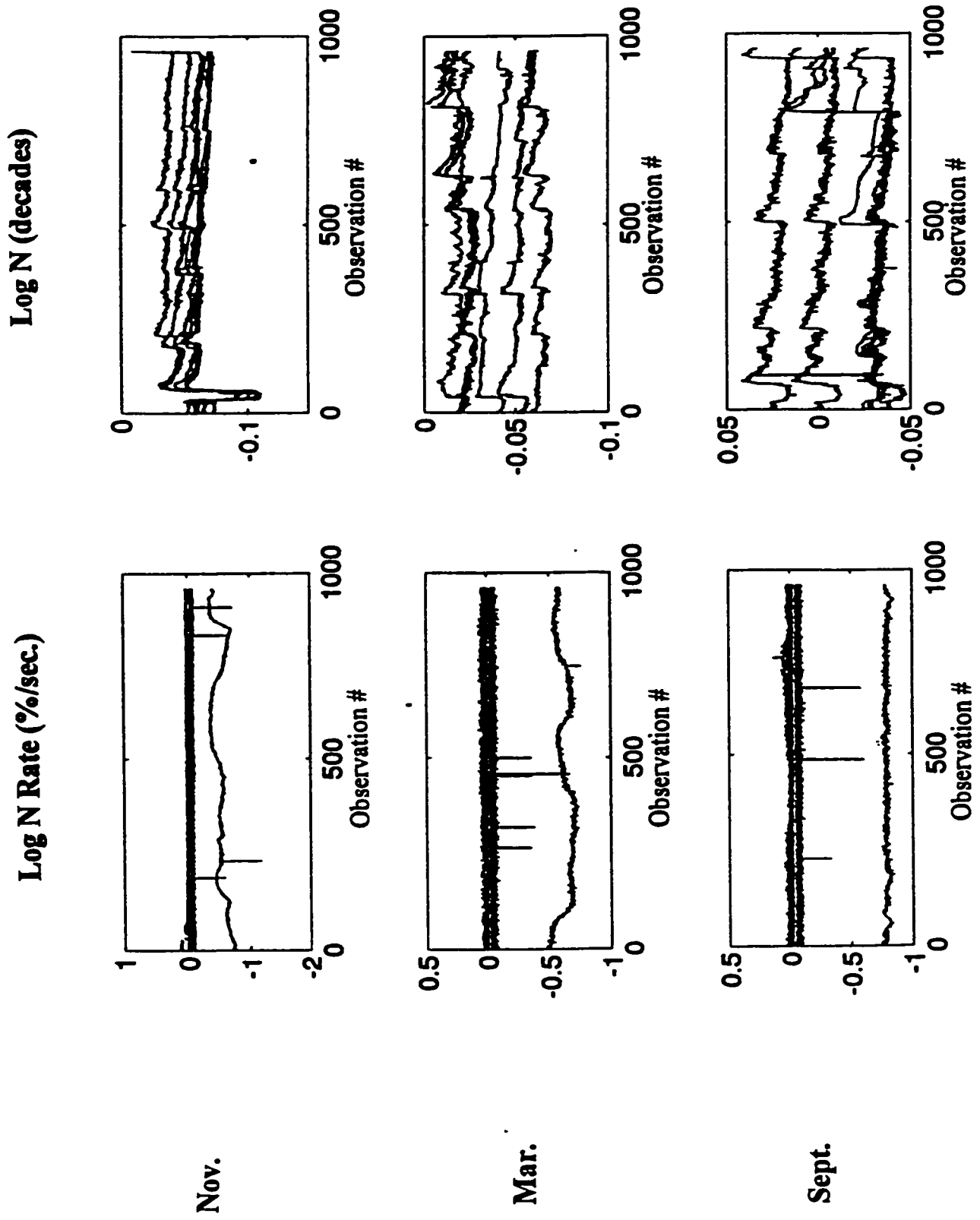
```
  if deflate == 1
   x(:,mm:mmm)=x(:,mm:mmm)-tlev1(:,i)*palllev1(mm:mmm,numpc)';
  end

  if deflate == 2
   if i==1 | i==2
    x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,1)*palllev1(mm:mmm,numpc)';
   end
   if i==3 | i==4
    x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,2)*palllev1(mm:mmm,numpc)';
   end
   if i==5 | i==6 | i==7 | i==8
    x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,3)*palllev1(mm:mmm,numpc)';
   end
   if i==9
    x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,4)*palllev1(mm:mmm,numpc)';
   end
   if i==10 | i==11
    x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,5)*palllev1(mm:mmm,numpc)';
   end
   if i==12 | i==13
    x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,6)*palllev1(mm:mmm,numpc)';
   end
   if i==14
    x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,7)*palllev1(mm:mmm,numpc)';
   end
   if i==15
    x(:,mm:mmm)=x(:,mm:mmm)-tlev2(:,8)*palllev1(mm:mmm,numpc)';
   end
  end

  if deflate == 3
   x(:,mm:mmm)=x(:,mm:mmm)-tc*palllev1(mm:mmm,numpc)';
  end


%   x(:,mm:mmm)=x(:,mm:mmm) -tc*palllev1(mm:mmm,numpc)';
%   x(:,mm:mmm)=x(:,mm:mmm)-tlev1(:,i)*palllev1(mm:mmm,numpc)';
%   x(:,mm:mmm)=x(:,mm:mmm) -t(:,i)*pall(:,((numpc*blockno)-(blockno-i)))';
%   xhat(:,mm:mmm)=xhat(:,mm:mmm) + tc*pall(:,((numpc*blockno)-(blockno-i)))';
 end;


% Save scores
% Level 0
```

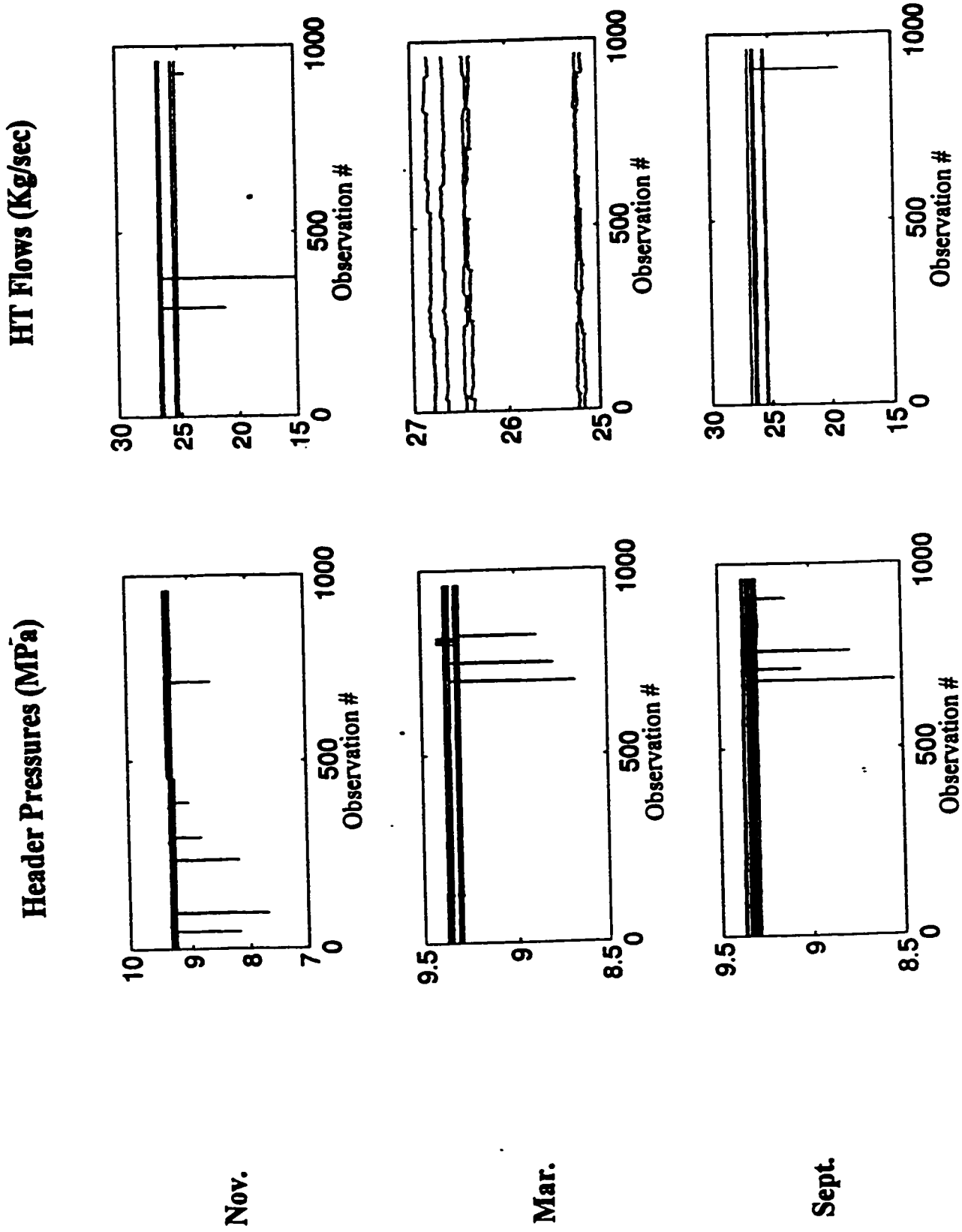
```
% if numpc == 1
%   ssexpxlev0all=ssexpxlev0';
% else
%   ssexpxlev0all=[ssexpxlev0all ssexpxlev0'];
% end

% Level 1
 if numpc == 1
   talllev1pre=tlev1;
%   ssexpxlev1all=ssexpxlev1';
 else
   talllev1pre=[talllev1pre tlev1];
%   ssexpxlev1all=[ssexpxlev1all ssexpxlev1'];
 end

% Level 2
 if numpc == 1
   talllev2pre=tlev2;
%   ssexpxlev2all=ssexpxlev2';
 else
   talllev2pre=[talllev2pre tlev2];
%   ssexpxlev2all=[ssexpxlev2all ssexpxlev2'];
 end

% Level 3
 if numpc ==1
   tcallpre=tc;
%   ssexpxlev3all=ssexpxlev3';
 else
   tcallpre=[tcallpre tc];
%   ssexpxlev3all=[ssexpxlev3all ssexpxlev3'];
 end

% Calculate SPE
% SPE Calculations
% Level 0
 for i=1:row
   for j=1:col
     SPElev0pre(i,j)=(sum((x(i,j)').^2))';
   end
 end

% Level 1
 mm=1
 for i = 1:noblklev1
   if i > 1;
```

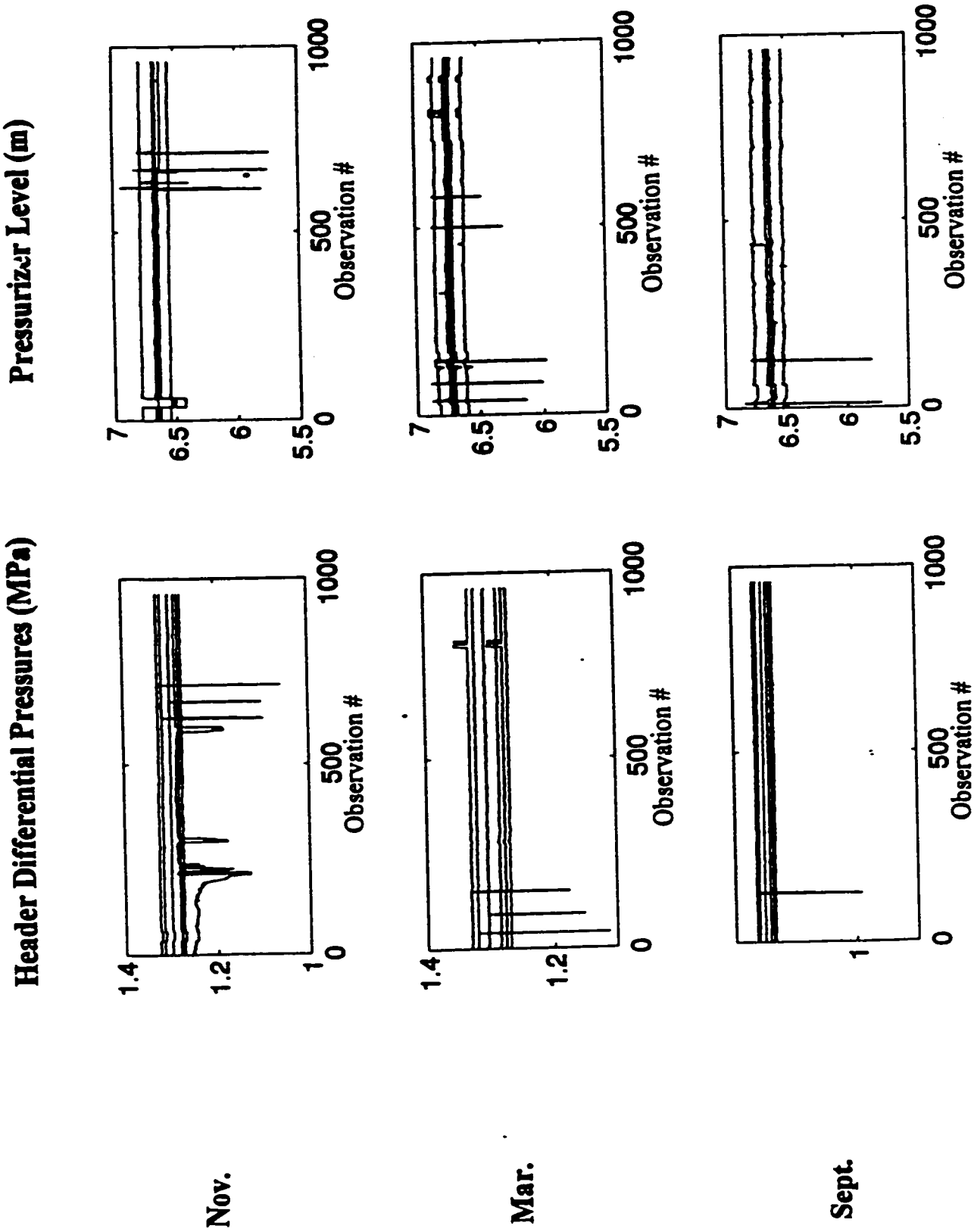
```
    mm=sum(mlev1(1:i-1))+1;
   end;
   mmm = sum(mlev1(1:i));
   SPElev1pre(:,i)=(sum((x(:,mm:mmm)').^2))';
  end;

% Level 2
  mm=1
  for i = 1:noblklev2
   if i > 1;
     mm=sum(mlev2ss(1:i-1))+1;
   end;
   mmm = sum(mlev2ss(1:i));
   SPElev2pre(:,i)=(sum((x(:,mm:mmm)').^2))';
  end;

  SPElev3pre=(sum((x').^2))';


  SPE=(sum((x.^2)'))';

  SPEpre(:,numpc)=SPE;
%  tcpre(:,numpc)=tc;

 end; % End of PC Loop
```

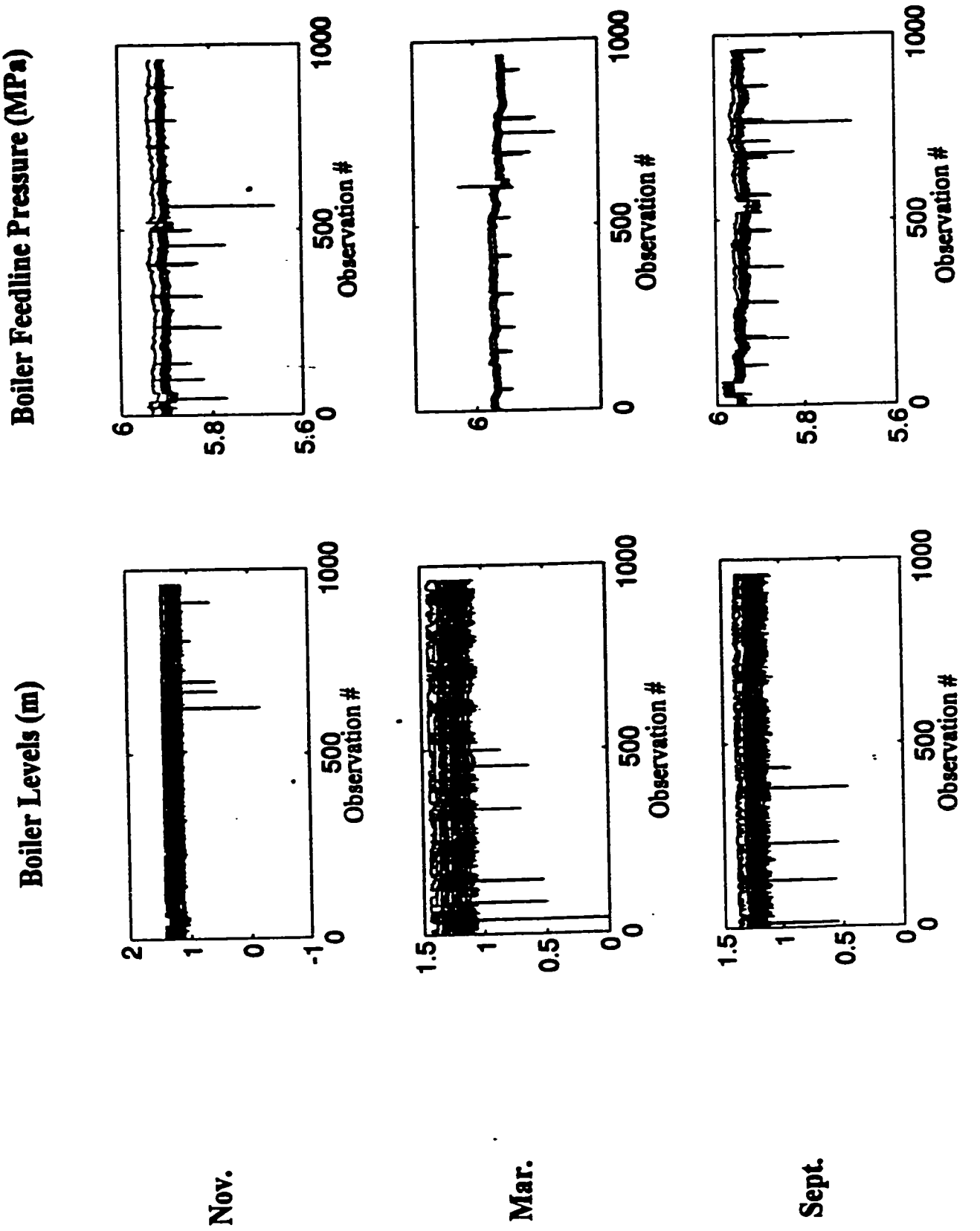
APPENDIX E

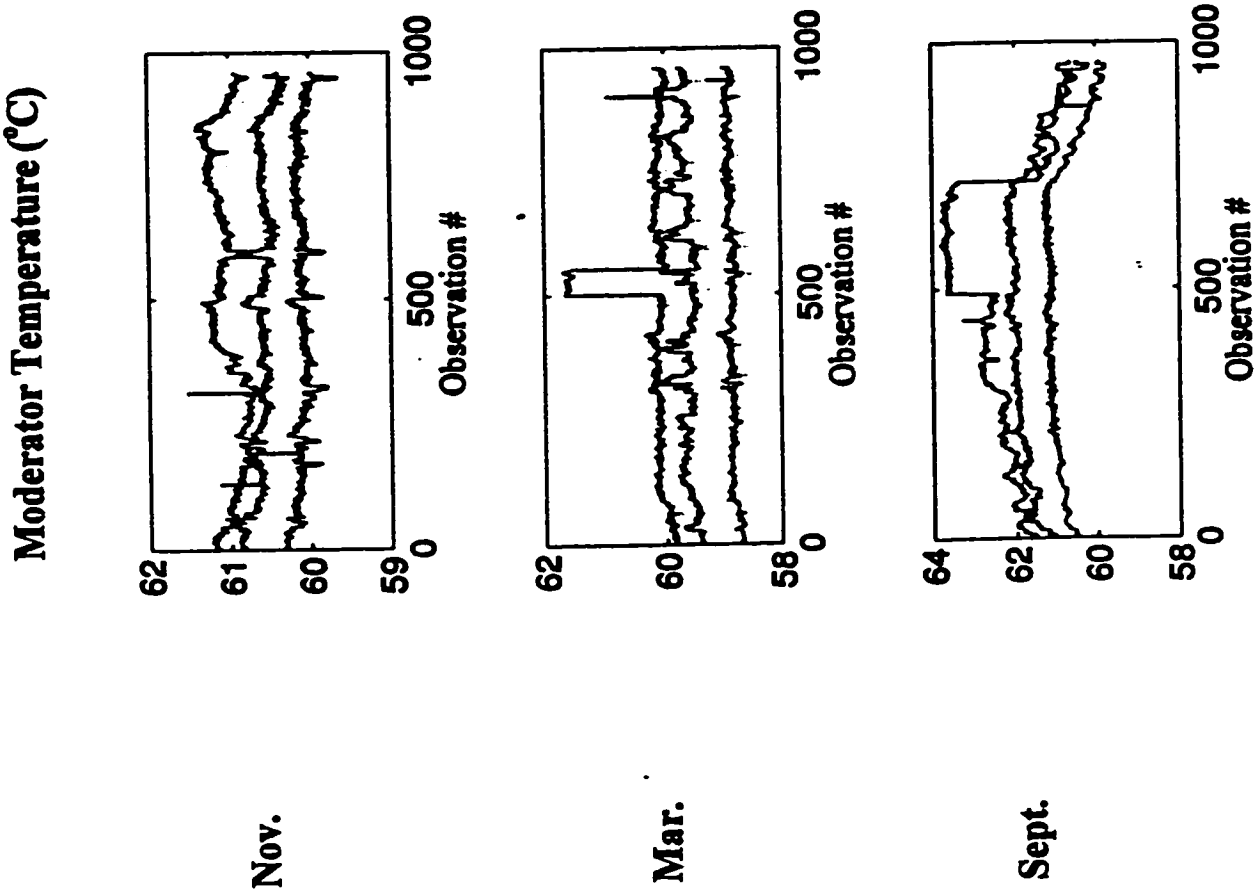
Raw Data


Log N (decades)
Log N Rate (%/sec.)
Observation #
Nov.
Mar.
Sept.


HT Flows (Kg/sec)
Header Pressures (MPa)
Nov.
Mar.
Sept.
Observation #
0
500
1000
30
25
20
15
27
26
25
10
9
8
7
9.5
9
8.5


Pressurizer Level (m)
Header Differential Pressures (MPa)
Observation #
Nov.
Mar.
Sept.


Boiler Feedline Pressure (MPa)
Boiler Levels (m)
Nov.
Mar.
Sept.
Observation #
0
500
1000
6
5.8
5.6
2
1
0
-1
1.5
0.5


Moderator Temperature (°C)
Nov.
Mar.
Sept.
62
61
60
59
64
58
0
500
1000
Observation #

Appendix F

Proofs for Sum of Squares Explained

In Chapter 4, it was noted that the sum of squares explained for one block in Level 1 is equal to the average of the sum of squares explained for the individual variables in that block. Similarly, the sum of squares explained for a block in Level 2 is equal to the average of the sum of squares explained for the blocks in Level 1 which combine to make the block in Level 2. A proof for these observations will be given below.

This proof is based on the assumption that the original variables are mean-centered and auto-scaled. It is also assumed that there is no missing data. Given these assumptions, the sum of the squares for each individual original variable is equal to the number of observations, as shown below:

$$SS_{original} = \sum_{i=1}^{n} x_i^2$$

$$\text{but : } x_i = \frac{(x_i - \bar{x})}{\sigma}$$

$$\therefore SS_{original} = \sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{\sigma}\right)^2 = \sum_{i=1}^{n}\frac{(x_i - \bar{x})^2}{\sigma^2} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$\text{However : } \sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \text{ from Equation 2.1}$$

$$\therefore SS_{original} = \frac{1}{\left(\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)}\sum_{i=1}^{n}(x_i - \bar{x})^2 = n - 1$$

Therefore, the original sum of squares can be considered a constant, C, for all variables. Now, consider how the sum of squares explained is calculated for different levels in the model.

Individual Variables:

$$SS_{Explained} = \frac{SS_{Original} - SS_{New}}{SS_{Original}} = \frac{C - \sum_{i=1}^{n} X_{New,i}}{C}$$

Therefore, the average of the first six variables would be equal to:

$$= \frac{\sum_{j=1}^{6}\left(\frac{C - \sum_{i=1}^{n} X_{New,i,j}}{C}\right)}{6} = \frac{\frac{1}{C}\sum_{j=1}^{6}\left(C - \sum_{i=1}^{n} X_{New,i,j}\right)}{6} = \frac{6C - \sum_{j=1}^{6}\sum_{i=1}^{n} X_{New,i,j}}{6C}$$

From Appendix D, the formula used to calculate the sum of squares for Level 1 is the equation derived above, as shown below:

$$SS_{Explained}\ (\text{Level 1}) = \frac{\sum_{j=1}^{6}\sum_{i=1}^{n} X_{Original,i,j} - \sum_{j=1}^{6}\sum_{i=1}^{n} X_{New,i,j}}{\sum_{j=1}^{6}\sum_{i=1}^{n} X_{Original,i,j}}$$

$$= \frac{6C - \sum_{j=1}^{6}\sum_{i=1}^{n} X_{New,i,j}}{6C}$$

Similarly, for Level 2, the sum of squares explained is calculated as follows:

$$SS_{Explained}\ (\text{Level 2}) = \frac{\sum_{j=1}^{12}\sum_{i=1}^{n} x_{Original,i,j} - \sum_{j=1}^{12}\sum_{i=1}^{n} x_{New,i,j}}{\sum_{j=1}^{12}\sum_{i=1}^{n} x_{Original,i,j}}$$

$$= \frac{12C - \sum_{j=1}^{12}\sum_{i=1}^{n} x_{New,i,j}}{12C}$$

The average of the sum of squares explained for the first two blocks in Level 1 is equal to:

$$= \frac{SS_{Explained,Block1} + SS_{Explained,Block2}}{2}$$

$$= \frac{\left(\frac{6C - \sum_{j=1}^{6}\sum_{i=1}^{n} x_{New,i,j}}{6C} + \frac{6C - \sum_{j=7}^{12}\sum_{i=1}^{n} x_{New,i,j}}{6C}\right)}{2}$$

$$= \frac{12C - \sum_{j=1}^{12}\sum_{i=1}^{n} x_{New,i,j}}{12C}$$

The above average is equal the sum of squares explained for Level 2.

APPENDIX G

Sensitivity Results for November and March Data

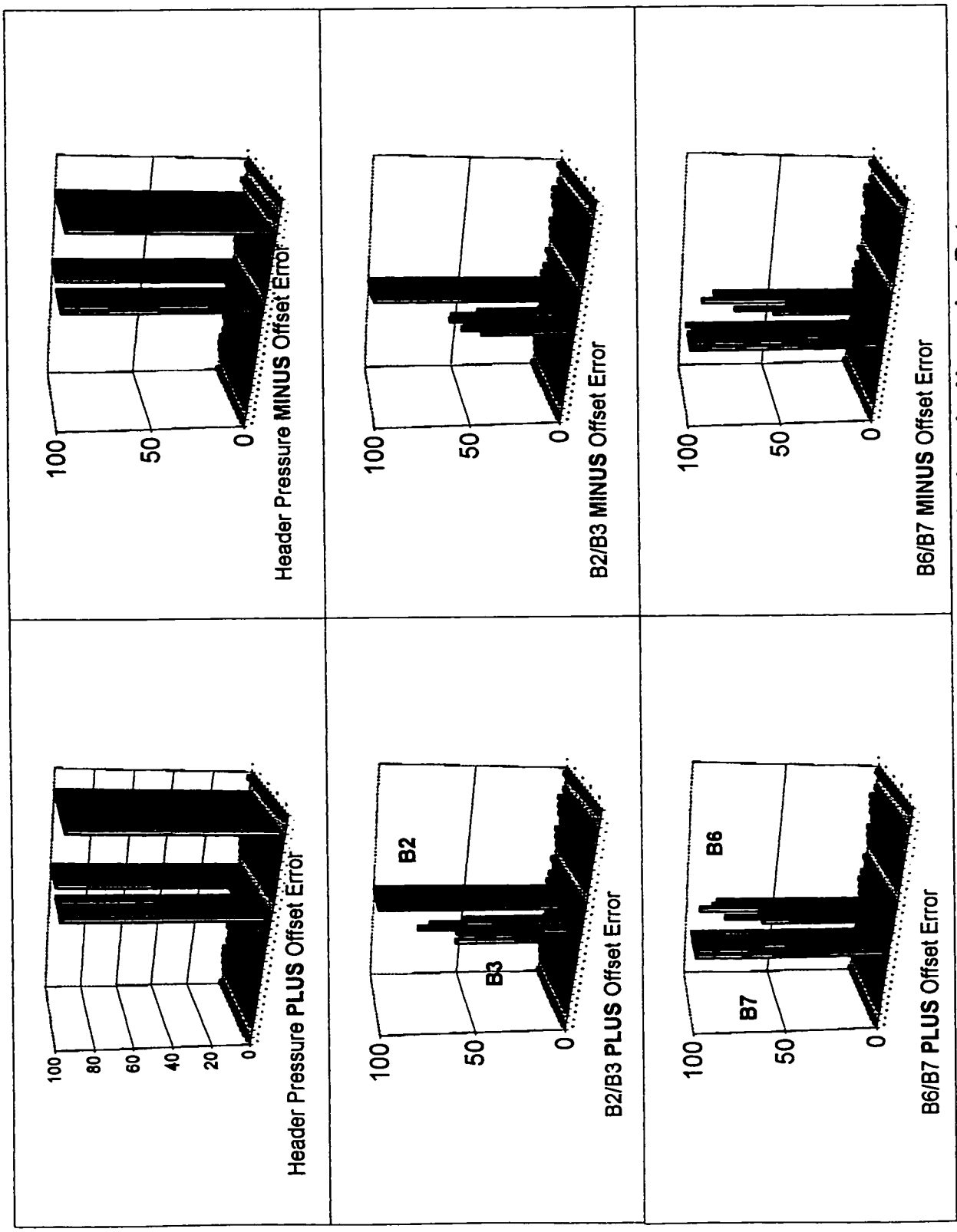


Figure G.1: Sensitivity Results for Header Pressures and Boiler Levels, November Data

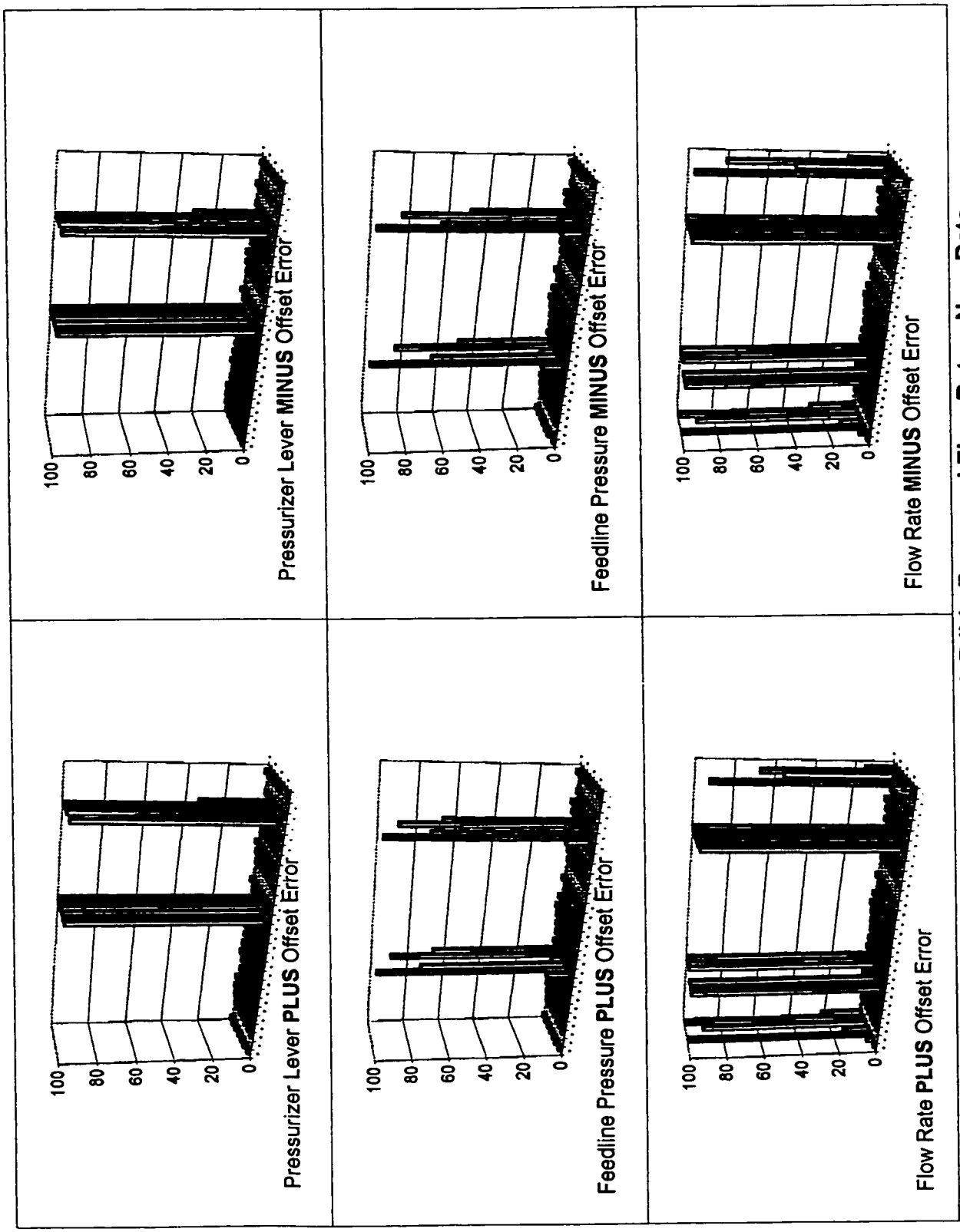


Figure G.2: Sensitivity Results for Press. Level, FdLin Press and Flow Rates, Nov. Data

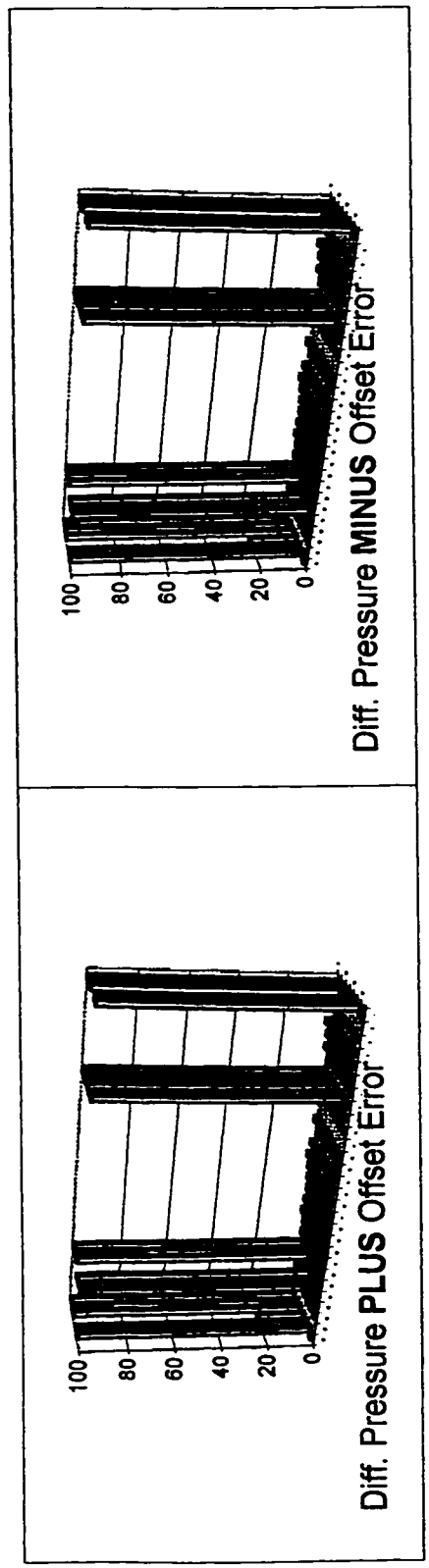


Figure G.3: Sensitivity Results for Diff. Pressures, November Data

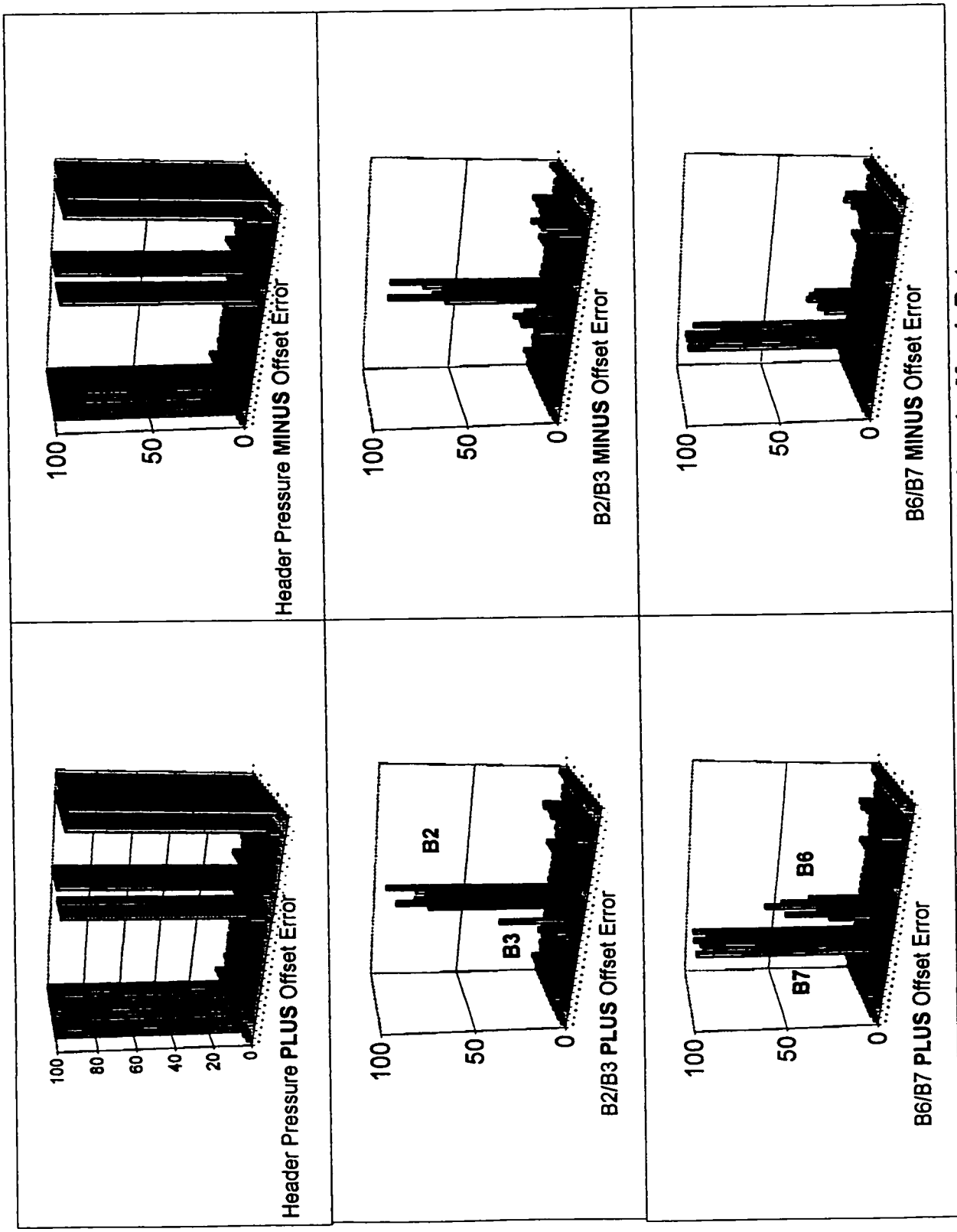


Figure G.4: Sensitivity Results for Header Pressures and Boiler Levels, March Data

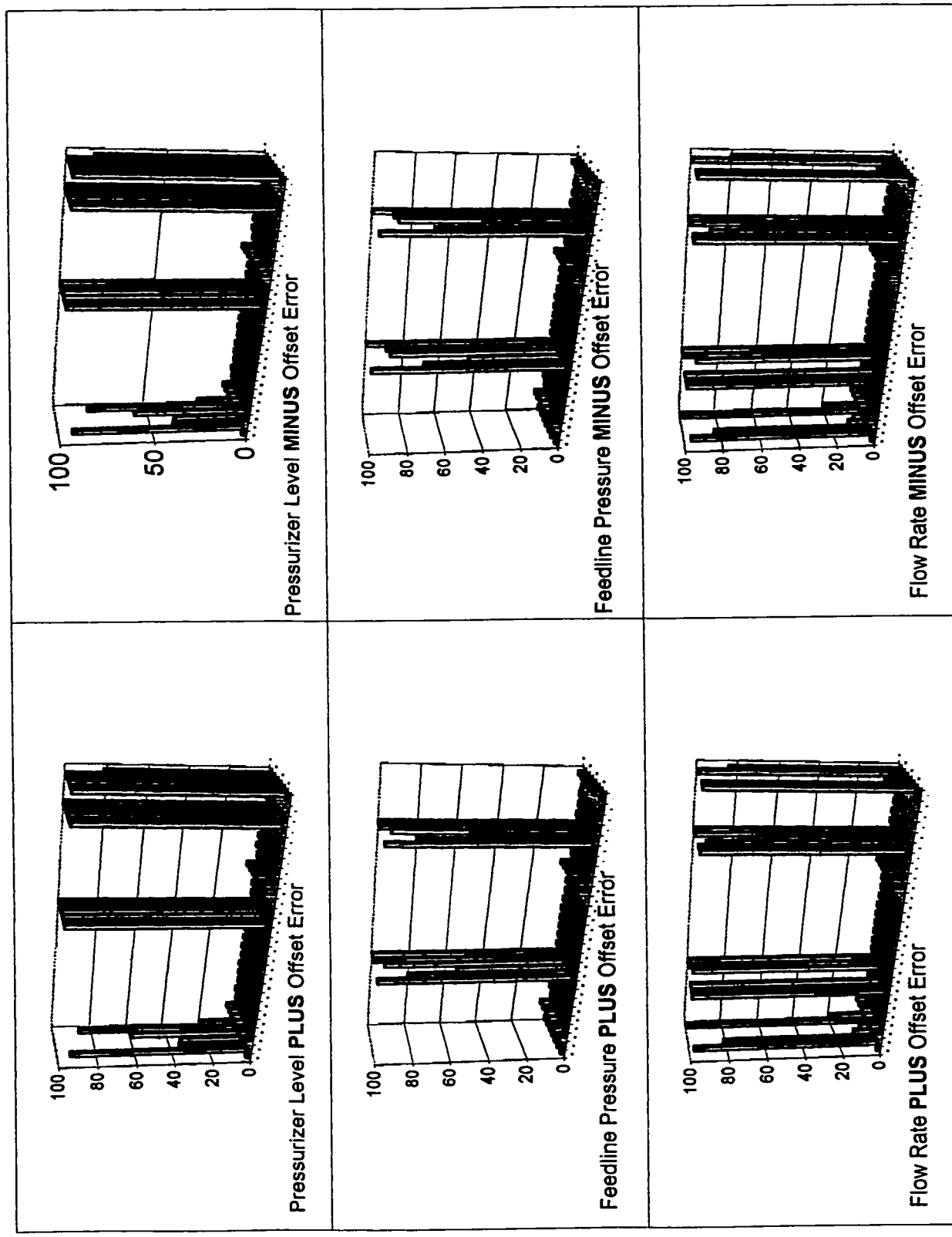


Figure G.5: Sensitivity Results for Press. Level, FdLin Press and Flow Rates, Mar. Data

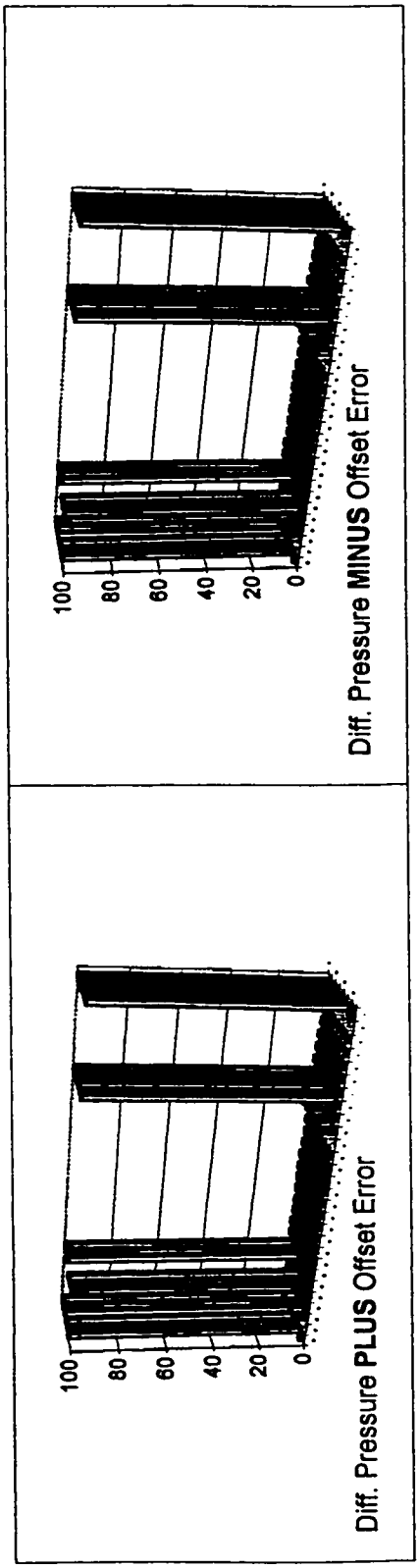


Figure G.6: Sensitivity Results for Diff. Pressures, March Data