To the Graduate Council:

I am submitting herewith a dissertation written by Claude Irvine entitled "A Study of Nuclear Plant Heat Rate Optimization Using Nonlinear Artificial Intelligence and Linear Statistical Analysis Models." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with major in Nuclear Engineering.

Robert Uhrig, Major Professor

We have read this dissertation
And recommend its acceptance:

Accepted for the Council:

Associate Vice Chancellor and
Dean of The Graduate School

A Study of Nuclear Plant Heat Rate Optimization Using Nonlinear Artificial Intelligence and Linear Statistical Analysis Models

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Claude Irvine
August 2000

Copyright © 2000
by
Claude Irvine

All rights reserved.
No part of this dissertation may be reproduced in any form
without permission in writing from the author.

ACKNOWLEDGEMENTS

I would like to express thanks to the Nuclear Engineering Department at the University of Tennessee for providing an excellent research environment, including but not limited to, the graduate courses, professors, and technical support staff. And while I am gratefully to all, I am particularly thankful to my advisor, Dr. Robert Uhrig, for his encouragement and support in this endeavor. He has read a number of revisions of the dissertation and the importance of this work is in due in no small part to his suggestions and comments.

I am also appreciative of Dr. Yovan Lukic and others out at Arizona's Palo Verde Nuclear Generation Station for providing the nuclear data to analyze, and to Idaho National Engineering and Environmental Laboratory for the financing of this project.

Lastly, the greatest debt is owed to my son and daughter, David and Jennifer, whom I have encouraged over the many years to do their homework and excel in their academic studies; in return have encouraged me to do my homework and complete this dissertation.

ABSTRACT

The emphasis of this dissertation is on developing methods by which a combination of multivariate analysis techniques (MAT) and artificial intelligence (AI) procedures can be adapted to on-line, real time monitoring systems for improving nuclear plant thermal efficiency. Present-day first principle models involve performing a heat balance of plant systems and the reactor coolant system. Typical variables involved in the plant data acquisition system usually number one-to-two thousand. The goal of the current work is twofold. First, simulate the heat rate with MAT and AI computer models. The second objective is to selectively reduce the number of predictors to only the most important variables, induce small perturbations around normal operating levels, and evaluate changes in the magnitude of plant efficiency. It is anticipated that making small changes will improve the thermal efficiency of the plant and lead to supplementary cost savings.

Conclusions of this report are several. A sensitivity analysis showed the reduction of input variables by dimensionality reduction, i.e., principal component analysis or factor analysis, removes valuable information. Predictors can simply be eliminated from the input space, but dimensionality reduction of the input matrix is not an alternative option. However, perturbation modeling does require data to be standardized and collinear variables removed. Filtering of input data is not recommended except to remove outliers.

It's ascertained that perturbation or sensitivity analysis differs from prediction modeling in that two additional requirements are necessary besides the criterion prediction. One is the magnitude of the criterion result given an input perturbation, and second, is the directionality of the model. Directionality is defined as the positive or negative movement of the heat rate (criterion) given a predetermined increase/decrease in predictor value, or input perturbation. While the criterion prediction is still important, it is directionality that determines whether a model is capturing proper changes in system process information. Final results showed that although the secondary-side of a nuclear plant might meet thermodynamic conditions for a steady-flow system, temporal information is needed by the model in order to capture system process

information. Modeling of the data is governed by quasi-static range theory, which states data must be closely spaced (in time) and prior temporal information is necessary. The conclusion reached is the perturbation model of a nuclear plant is a time-dependent, dynamic system; all indications as of date show it is also nonlinear. Hence a time-dependent nonlinear modeling method, such as a neural network with time delayed inputs, is needed for sensitivity modeling.

TABLE OF CONTENTS

Chapter 1 1
Introduction 1
1.1 Preface 1
1.2 Thermal Performance of Nuclear Plants 4
1.2.1 Rankine Cycle 5
1.2.2 Thermal Efficiency 6
1.2.3 Heat Rate 7
1.2.4 Influence of Plant Parameters on Heat Rate 11
1.3 Data Analysis Concerns 12
1.3.1 Variance 13
1.3.2 Linear Correlation 14
1.4 Prediction Modeling 16
1.5 Variable Selection Criteria 18
1.6 Organization of the Dissertation 19
Chapter 2 21
Review of Literature 21
2.1 Preface 21
2.2 Current Nuclear Plant Thermal Efficiency Perspectives 21
2.3 New Approaches to Modeling, and Merger of AI 24
2.4 Factor Analysis Background 28
Chapter 3 32
Review of Mathematics 32
3.1 Mathematics Preliminaries 32
3.2 Multiple Regression 33
3.2.1 Multiple Regression using Standardized Data 38
3.3 Factor Analysis 40
3.3.1 Principal Components Model vs. Classical Factor Analysis Model 41
3.3.2 Factor Mathematics 42
3.3.2.1 Relationships between Factor Pattern and Structure 46
3.3.2.2 Relationships between Observed and Reproduced Correlation Matrices 47
3.3.3 Number of Common Factors 51
3.3.4 Variance Analysis 52
3.3.5 Summary 58
3.4 Principal Components Regression 58
3.4.1 Linear Transformations 59
3.4.2 Principal Component Analysis (PCA) 60
3.4.3 Relating Principal Component Scores to Factor Scores 63
3.4.4 Principal Component Regression 65
3.4.5 PCA Dimensionality Issues 66
3.5 Partial Least Squares 67
3.5.1 PLS Regression 67
3.5.2 Selection of Principal Variables 72
Chapter 4 73
State Variable Analysis 73
4.1 Preface 73
4.2 Computing Factor Scores 73
4.2.1 Addressing Indeterminacy of Factor and Pattern Results 74
4.2.2 Factor Score Prediction 76
4.3 PLS Estimates 76
4.4 Deciding Upon the Number of Common Factors or Principal Components 77
4.5 Neural Network Estimates 78
4.6 Multiple Regression Estimates 80

4.7 Multiple Regression Estimates using Factor and PCA Scores 83
Chapter 5 84
Set Point Variable Analysis 84
5.1 Preface 84
5.2 Predictor Perturbations 85
5.2.1 PET and MPET Units 85
5.2.2 Predictor Perturbations Specific to Neural Networks 86
5.3 MR-MR, PLS-PLS Mappings 87
5.3.1 Independence of the Type of State Variables 88
5.3.2 Multicollinearity 90
5.3.3 High Collinearity of State Variables 91
5.4 Final Results with Noncollinear state matrix 92
5.5 Considerations for Future Work 93
Chapter 6 95
Results 95
6.1 Preface 95
6.2 Methodology and Data Limitations 95
6.3 Lessons Learned 96
6.4 Findings and Recommendations 97
6.5 Afterthoughts 104
6.6 Summary 105
Chapter 7 107
Quasi-static Assumptions 107
7.1 Preface 107
7.2 Regularization 107
7.3 Derivation and Meaning of Quasi-Static Range of a Variable 107
7.3.1 The QSR of Predictor Feedwater Flow for Turbine Heat Rate 112
7.3.2 U2 Data Results 112
7.3.3 Consistency of Directionality and Magnitude 113
7.4 Summary 114
Chapter 8 115
Conclusions 115
8.1 Summation of Main Points 115
8.2 Overall Conclusion and Recommendations for Future Study 118
References 120
Appendix 125
Appendix B Listing of Computer Codes 270
File 1 Group_clean.m 270
File 2 Group_state.m 271
File 3 Minresv3.m 274
File 4 W.m 279
File 5 Wmatrix.m 279
File 6 wsfind.m 280
File 7 SetPointPLS.m 280
File 8 trainNN.m 284
File 9 sorteig.m 284
File 10 PTVV.m 285
File 11 preprocessv2.m 286
File 12 pls_reverse.m 291
File 13 pls_onestep.m 292
File 14 perturb.m 292
File 15 mr.m 294
File 16 inputs.m 294
File 17 Group_Setpoint.m 296
File 18 Group_fix.m 301

File 19 fmap.m 302
File 20 addpoints.m 303
Vita 305

LIST OF TABLES

Table 1 List of Cycle Parameters and Their Influence on Heat Rate[9] 12
Table 2 MSE values for factor score prediction 76
Table A-1 MSE for various neural network models 145
Table A-2 MSE values for various factor scores using multiple regression 161
Table A-3 State variable descriptions 164
Table A-4 Comparison of the ten most important setpoint variables affecting heat rate, for different state variable scores (MR-MR mappings) 193
Table A-5 Comparison of the ten most important setpoint variables affecting heat rate, for different state variable scores (PLS-PLS mappings) 194
Table A-6 Comparison of the ten most important setpoint variables affecting heat rate, for PCA state variable scores (MR-MR mappings) 195
Table A-7 Comparison of the ten most important setpoint variables affecting heat rate, for FACTOR state variable scores (MR-MR mappings) 196
Table A-8 Comparison of the ten most important setpoint variables affecting heat rate, for different state variable scores (MR-MR mappings), (original 43 state variables reduced to 21 by removing highly collinear variables) 197
Table A-9 Comparison of the ten most important setpoint variables affecting heat rate, for different state variable scores and mappings (original 43 state variables reduced to 21 by removing highly collinear variables) 198
Table A-10 Comparison of the ten most important setpoint variables affecting heat rate, for 21 SD state variable scores (MR-NN mappings), top row--MPET, bottom row--median PET 199
Table A-11 Set Point variable descriptions 202
Table A-12 List of Set Point Variables that were removed (shown in 2nd column) because of high collinearity (>.95) with other set point variables (shown in 1st column) 203
Table A-13 List of State Variables that were removed (shown in 2nd column) because of high collinearity (>.95) with other state variables (shown in 1st column) 204
Table A-14 Variable ranking, from least to most important 228
Table A-15 Comparison of Plant Parameters to Reference Directionalities 257

LIST OF FIGURES

Figure 1 Fitting Data to a Model 3
Figure 2 Rankine Cycle[27] 5
Figure 3 Funnel Diagram 16
Figure 4 Collapsing of N-space to m-space 52
Figure 6 Plot showing two criterion points separated in time 109
Figure 7 Plot showing quasi-static range of predictors A, B, and C 109
Figure 8 Plot showing system classifications: Quasi-static, Dynamic, and Static 110
Figure A-1 Factor scores obtained from four different pattern matrices 126
Figure A-2 PLS estimate, 5 stages, predictors are 43 state variables, no pre-scaling 127
Figure A-3 PLS estimate, 10 stages, predictors are 43 state variables, no pre-scaling 128
Figure A-4 PLS estimate, 15 stages, predictors are 43 state variables, no pre-scaling 129
Figure A-5 PLS estimate, 20 stages, predictors are 43 state variables, no pre-scaling 130
Figure A-6 MSE versus # PLS stages, predictors are 43 state variables, no pre-scaling 131
Figure A-7 MSE versus # PLS stages, predictors are 43 state variables, SD pre-scaling 132
Figure A-8 PLS estimate, predictors are factor scores (5 dimensional), SD pre-scaling 133
Figure A-9 MSE versus # PLS stages, predictors are factor scores (5 dimensional), SD pre-scaling 134
Figure A-10 PLS estimate, predictors are principal component scores (5 dimensional), SD pre-scaling .. 135
Figure A-11 MSE versus # PLS stages, predictors are principal component scores (5 dimensional), SD pre-scaling 136
Figure A-12 Eigenvalues for 43 state variables 137
Figure A-13 Scree plot for principal components 138
Figure A-14 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 2 hidden nodes (tansig) and linear output layer [VV every 4, except 216:289] 139
Figure A-15 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 2 hidden nodes (tansig) and linear output layer, [VV every 4, all 216:289] 140
Figure A-16 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 2 hidden nodes (tansig) and linear output layer, [VV every 4] 141
Figure A-17 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 2 hidden nodes (tansig) and linear output layer, [No VV] 142
Figure A-18 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 2 hidden nodes (tansig) and linear output layer, [5% noise in VV.T] 143
Figure A-19 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 2 hidden nodes (tansig) and linear output layer, [5% noise in VV.T & VV.P] 144
Figure A-20 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 4 hidden nodes (tansig) and linear output layer, [5% noise in VV.T] 146
Figure A-21 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 4 hidden nodes (tansig) and linear output layer, [5% noise in VV.T & VV.P] 147
Figure A-22 Multiple regression estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling 148
Figure A-23 Beta Coefficients for multiple regression estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling 149
Figure A-24 Error Bar Plot for multiple regression estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling 150
Figure A-25 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 2-layer NN with 1 hidden node (linear) and linear output layer, [5% noise in VV.T] 151
Figure A-26 Multiple regression estimate, predictors are principal component scores (5 dimensional), SD pre-scaling 152
Figure A-27 Beta Coefficients for multiple regression estimate, predictors are principal component scores (5 dimensional), SD pre-scaling 153
Figure A-28 Error Bar Plot for multiple regression estimate, predictors are principal component scores (5 dimensional), SD pre-scaling 154
Figure A-29 Multiple regression estimate, predictors are 43 state variables, SD pre-scaling 155

Figure A-30 Beta Coefficients for multiple regression estimate, predictors are 43 state variables, SD pre-scaling 156
Figure A-31 Multiple regression estimate, predictors are 43 state variables, no pre-scaling 157
Figure A-32 Beta Coefficients for multiple regression estimate, predictors are 43 state variables, no pre-scaling 158
Figure A-33 Error Bar Plot for multiple regression estimate, predictors are 43 state variables, no pre-scaling 159
Figure A-34 Interactive fitting and visualization of response surface for linear multiple regression estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling 160
Figure A-35 Multiple regression estimate, predictors are 5 highest correlated state variables with heat rate, SD pre-scaling 162
Figure A-36 MR MSE and R^2 for Common and PCA Factors 163
Figure A-37 Change in set point signal for a .5% perturbation 165
Figure A-38 Change in set point signal for a .5% perturbation 166
Figure A-39 Change in heat rate for a 5% change in input, MR-MR mappings 167
Figure A-40 Change in heat rate for a 5% change in input, MR-NN mappings, NN overtrained 168
Figure A-41 Change in heat rate for a 5% change in input, MR-NN mappings, all PCA inputs to NN with cross-validation 169
Figure A-42 Change in heat rate for a .5% change in input, MR-NN mappings, 21 SD scores with cross-validation 170
Figure A-43 Typical residual plot showing change in heat rate mapping (MAP2) using a NN with cross-validation (21 SD scores as predictors, heat rate as criterion) 171
Figure A-44 Change in heat rate for a .5% change in input, MR-NN mappings, 5 PCA inputs to NN with cross-validation 172
Figure A-45 Change in heat rate for a .5% change in input, MR-NN mappings, 5 FACTOR inputs to NN with cross-validation 173
Figure A-46 Change in heat rate for a .5% change in input, MR-MR mappings, 5 PCA inputs 174
Figure A-47 Change in heat rate for a .5% change in input, MR-NN mappings, (per=3, nodes=4, loops=5, tansig-purelin, .005 pert, epochs=300, 21 SD inputs) 175
Figure A-48 Change in heat rate for a negative .5% change in input, MR-NN mappings, (per=3, nodes=4, loops=5, tansig-purelin, -.005 pert, epochs=300, 21 SD inputs) 176
Figure A-49 Change in MPET for a +.5% and -.5% perturbation, MR-NN mappings, (per=3, nodes=4, loops=5, tansig-purelin, +.005 & -.005 pert, 21 SD inputs) 177
Figure A-50 Change in set point signal for a .5% perturbation 178
Figure A-51 Change in set point signal for a .5% perturbation 179
Figure A-52 Change in set point signal for a .5% perturbation 180
Figure A-53 Change in MPET for a +.05% and -.05% perturbation, MR-NN mappings, (per=3, nodes=4, loops=5, tansig-purelin, +.0005 & -.0005 pert, 21 SD inputs) 181
Figure A-54 Change in heat rate for a negative .05% change in input, MR-NN mappings, (per=3, nodes=4, loops=5, tansig-purelin, .0005 pert, epochs=300, 21 SD inputs) 182
Figure A-55 Change in set point signal for a .05% perturbation 183
Figure A-56 Change in heat rate for a .05% change in input, MR-NN mappings, (per=3, nodes=4, loops=5, tansig-purelin, .0005 pert, epochs=300, 21 SD inputs) 184
Figure A-57 Change in heat rate for a .05% change in input, MR-NN mappings, (no cross-validation, nodes=4, loops=5, tansig-purelin, .0005 pert, epochs=3000, 21 SD inputs) 185
Figure A-58 Change in heat rate for a .05% change in input, MR-NN mappings, (no cross-validation, nodes=4, loops=5, tansig-purelin, .0005 pert, epochs=3000, 21 SD inputs) 186
Figure A-59 Change in heat rate for a +5% perturbation in each setpoint variable (FACTOR Scores) 187
Figure A-60 Sensitivity plot showing change in heat rate per +5% perturbation in each setpoint variable (FACTOR Scores) 188
Figure A-61 Change in heat rate for a +5% perturbation in each setpoint variable (PCA Scores) 189
Figure A-62 Sensitivity plot showing change in heat rate per +5% perturbation in each setpoint variable (PCA Scores) 190
Figure A-63 Change in heat rate for a +5% perturbation in each setpoint variable (43 SD Scores) 191

Figure A-64 Sensitivity plot showing change in heat rate per +5% perturbation in each setpoint variable (43 SD Scores) 192
Figure A-65 Change in heat rate as a function of number of state variable PCA components, set point variable -- TRB HEAT RATE GROSS UNCORR and FW PMP A IN ENTH, MR-MR mappings, 21 state matrix, pert = .0005, (all values shown are absolute) 200
Figure A-66 Change in variance explained for state variables, 20 major PCA components (rank 43) and 20 major PCA components (rank 21) 201
Figure A-67 Winter plot of heat rate (and linear NN, MR, PLS estimates) 208
Figure A-68 Spring plot of heat rate (and linear NN, MR, PLS estimates) 209
Figure A-69 Summer plot of heat rate (and linear NN, MR, PLS estimates) 210
Figure A-70 Fall plot of heat rate (and linear NN, MR, PLS estimates) 211
Figure A-71 Winter plot of heat rate (and linear NN (no VV points), MR, PLS estimates) 212
Figure A-72 Winter plot of heat rate (and NN, MR, PLS estimates) 213
Figure A-73 Spring plot of heat rate (and NN, MR, PLS estimates) 214
Figure A-74 Summer plot of heat rate (and NN, MR, PLS estimates) 215
Figure A-75 Fall plot of heat rate (and NN, MR, PLS estimates) 216
Figure A-76 Winter plot, NN estimate of heat rate 217
Figure A-77 Figure A-76 rotated 218
Figure A-78 Winter plot, MR estimate of heat rate 219
Figure A-79 Figure A-78 rotated 220
Figure A-80 Winter plot, NN estimate of heat rate (for constant change) 221
Figure A-81 Spring plot, NN estimate of heat rate (for constant change) 222
Figure A-82 Summer plot, NN estimate of heat rate (for constant change) 223
Figure A-83 Fall plot, NN estimate of heat rate (for constant change) 224
Figure A-84 Winter plot, MR estimate of heat rate (for constant change) 225
Figure A-85 Winter plot, NN estimate of heat rate for all variables 1-55 226
Figure A-86 Winter plot, MR estimate of heat rate for all variables 1-55 227
Figure A-87 Winter perturbation plot, FW Pump(A) Speed 230
Figure A-88 Winter perturbation plot, FW Pump(B) Speed 231
Figure A-89 Winter perturbation plot, Total Feedwater Flow 232
Figure A-90 Winter perturbation plot, Total Feedwater Flow 233
Figure A-91 Spring perturbation plot, Total Feedwater Flow 234
Figure A-92 Spring perturbation plot, Total Feedwater Flow 235
Figure A-93 Summer perturbation plot, Total Feedwater Flow 236
Figure A-94 Summer perturbation plot, Total Feedwater Flow 237
Figure A-95 Fall perturbation plot, Total Feedwater Flow 238
Figure A-96 Fall perturbation plot, Total Feedwater Flow 239
Figure A-97 Winter perturbation plot, Total Feedwater Flow (for constant change) 240
Figure A-98 Spring perturbation plot, Total Feedwater Flow (for constant change) 241
Figure A-99 Summer perturbation plot, Total Feedwater Flow (for constant change) 242
Figure A-100 Fall perturbation plot, Total Feedwater Flow (for constant change) 243
Figure A-101 Winter daily perturbation plot, Feedwater Flow (12-12-97) 244
Figure A-102 Winter daily perturbation plot, Feedwater Flow (12-13-97) 245
Figure A-103 Winter daily perturbation plot, Feedwater Flow (12-14-97) 246
Figure A-104 Winter daily perturbation plot, Feedwater Flow (12-15-97) 247
Figure A-105 Winter daily perturbation plot, Feedwater Flow (12-16-97) 248
Figure A-106 Winter perturbation plot, Condensate pump(C) flow 249
Figure A-107 Winter perturbation plot, Condensate pump(C) flow 250
Figure A-108 Summer perturbation plot, Avg Main Steam Pressure 251
Figure A-109 Spring perturbation plot, Avg Main Steam Pressure 252
Figure A-110 Spring daily perturbation plot, Avg Main Steam Pressure (3-11-98) 253
Figure A-111 Spring daily perturbation plot, Avg Main Steam Pressure (3-12-98) 254
Figure A-112 Spring daily perturbation plot, Avg Main Steam Pressure (3-13-98) 255
Figure A-113 Spring daily perturbation plot, Avg Main Steam Pressure (3-14-98) 256
Figure A-114 Winter perturbation plot, Total Feedwater Flow, with Ridge Regression 258

Figure A-115 Spring perturbation plot, Total Feedwater Flow, with Ridge Regression 259
Figure A-116 Summer perturbation plot, Total Feedwater Flow, with Ridge Regression 260
Figure A-117 Fall perturbation plot, Total Feedwater Flow, with Ridge Regression 261
Figure A-118 Plot of Feedwater Flow 15 minute time separation between data points 262
Figure A-119 Plot of NN, MR, PLS estimates, 15 minute time separation between data points 263
Figure A-120 Perturbation plot with U2 data, 15 minute time separation between data points, NN with LM training and 14 hidden nodes 264
Figure A-121 Perturbation plot with U2 data, 15 minute time separation between data points, NN with LM training and 14 hidden nodes, Ridge Regression estimate 265
Figure A-122 Perturbation plot with U2 data, 30 minute time separation between data points, NN with LM training and 14 hidden nodes 266
Figure A-123 Perturbation plot with U2 data, 45 minute time separation between data points, NN with LM training and 14 hidden nodes 267
Figure A-124 Plot of NN, MR, PLS estimates, using an Elman neural network 268
Figure A-125 Elman network, 1 delay, 15 minute time separation between data points, 18 nodes 269

NOMENCLATURE

a_{jp}	pattern coefficient or loading
$\mathbf{a}_k$	column vector (MR only)
A	common factor pattern matrix or symmetric matrix (MR only)
b	weight or regression coefficient
b	weight or regression coefficient vector
B	weight matrix
C	minor product moment of **X**
communality	sum of the squares of the common-factor coefficients, or portion of total unit variance that is accounted for by the common factors
complexity	number of common factors used to describe the variable
cutoff	the magnitude squared of the largest covariance between X and Y deviations
e_j	error loading
E_j	error factor
e	error or residual
E	error matrix
enthalpy	property, h, defined as the sum of internal energy and the product of pressure and volume
entropy	property, S, so that the definition of entropy change is $\Delta S = \oint_{rev} \frac{\delta Q}{T}$
f	minimizing function
$f(\mathbf{x})$	multivariate density function
f	vector of common factors
F_p	common factor with index p
F	common factor matrix or error matrix (PLS derivations only)
factors	hypothetical constructs
h or h_j^2	communality
h	enthalpy
i	observation index (subscript)

$\mathbf{I}$	identity matrix
irreversible process	a process which is not reversible
j	variable index (subscript)
m	number of factors
$\mathbf{M}$	matrix containing both common and uniqueness coefficients
minres	minimum residuals
n	number of observed variables
N	number of observations
p	factor index (subscript)
pv	pressure-volume
$\mathbf{p}$	scores vector
$\mathbf{P}_{\text{scores}}$	scores matrix
pattern	matrix of common factor coefficients (loadings)
$\mathbf{q}$	weight vector
Q	heat energy
δQ	inexact differential, such that $Q_{1-2} = \int_1^2 \delta Q$ but $\int_1^2 \delta Q \neq Q_2 - Q_1$
$\mathbf{Q}$	weight matrix
r	correlation
r_{zz}	reliability coefficient
$\mathbf{R}$	correlation matrix
$\hat{\mathbf{R}}$	reproduced correlation matrix
reversible process	if by any means possible both the system and surroundings can return to the same state they were in before the process started, it is called a reversible process
s,S	entropy
s_j	specificity loading
s_{jp}	structure coefficient
S_j	specificity factor

$\mathbf{S}$	common factor structure matrix
specificity	uniqueness that is due to the particular selection of variables
structure	matrix of correlations between the variables and the factors
$\mathbf{t}$	scores vector
Ts	temperature-entropy
$\mathbf{T}_{scores}$	scores matrix
total variance	sum of communality and uniqueness (total magnitude is unity for small standard deviates)
u	uniqueness coefficient or loading
$\mathbf{u}$	score vector
$\mathbf{U}$	uniqueness matrix
$\mathbf{U}_{scores}$	scores matrix
uniqueness	contribution of the unique factor, or portion of total unit variance that is not accounted for by the common factors
unreliability	uniqueness that is due to unreliability of measurement
$\mathbf{V}$	covariance matrix
$\lvert\mathbf{V}\rvert^{1/2}$	generalized variance
$\mathbf{w}$	weight vector
W	work
$\mathbf{W}$	weight matrix
x, X	independent variable
x	water quality
$\overline{X}_j$	variable mean
$\bar{\mathbf{x}}$	vector mean
$\mathbf{X}$	independent variable matrix
y, Y	dependent variable
$\hat{y}$	estimated dependent variable
$\mathbf{y}$	vector of dependent variables or unique factors

$\mathbf{Y}$	dependent variable matrix or uniqueness matrix
Y_p	uniqueness factor
z	standard deviate
z_j	variable with index j
z_{nN}	variable value with variable index n and observation N
$\mathbf{Z}$	standard deviate matrix
z^*	small standard deviate
$\mathbf{z}^*$	vector of small standard deviates
$\mathbf{z}_x^*$	vector of small standard deviate predictors
$\mathbf{z}_y^*$	vector of small standard deviate criterions
$\mathbf{Z}^*$	small standard deviate matrix
σ	standard deviation
σ^2	variance
σ_j^2	variance for a given variable index j
σ_p^2	total contribution of a common factor to the variance of all variables
Ω	contribution of all common factors to the variance of all variables
Θ	correlation matrix of common factors
$\Re$	space of real numbers

LIST OF ABBREVIATIONS

AI	artificial intelligence
ANN	artificial neural network
BD	blowdown
BWR	boiling water reactor
CRD	control rod drive
DCA	drain cooler approach – temperature difference between the water leaving drain cooler and inlet feedwater
FA	factor analysis
HP	high pressure
ISCV	Instrumentation, Surveillance, Calibration and Verification
LAN	local area network
LP	low pressure
LSB	last stage bucket
MP	mathematic preliminaries
MPET	mean percentage error
MR	multiple regression
MS(s)	moisture separator(s)
MSE	mean squared error
MSRs	moisture separator reheaters
MW	mega-watts
NN	neural network
PCA	principal component analysis
PCR	principal component regression
PEANO	Process Evaluation and Analysis by Neural Operators
PEDS	Power Enhancement Delivery Software
PET	percentage error
PLS	partial least squares

psi, psia	pounds per square inch, pounds per square inch absolute
PWR	pressurized water reactor
PV	principal variables
QSR	quasi-static range (as applied to a predictor)
RMS	root mean squared
RPM	revolutions per minute
SD	standardized
SG	steam generator
SSE	sum squared error
TTD	terminal temperature difference – temperature difference between the heating steam and the cycle steam leaving the reheater
TPP	Thermal Performance Program
TVA	Tennessee Valley Authority
WAN	wide area network

Chapter 1

Introduction

1.1 Preface

The goals of this dissertation are several. Listed below, in order from least to greatest importance, are the main areas of concentration:

1. Determine and validate a smaller, optimal selection of input variables (collectively designated as *predictors*) from a much larger set for use in modeling the thermodynamic behavior of nuclear power plants.
2. Utilize the predictors to model the efficiency (or heat rate) of a nuclear power plant.
3. Determine predictors that will optimize the heat rate.

The work involves applying the predictors as inputs to an artificial intelligence (AI) prediction model that measures the *criterion* variable, heat rate. It is expected the performance characteristics of the AI model will capture sufficient process information to allow not only for the prediction of heat rate with existing scientific data, but also provide future projections based upon slightly modified values of the predictor matrix. Since heat rate is a calculated output of all thermodynamic efficiency models, and hence, the reference or criterion variable to judge improvements in efficiency by examining increases/decreases in its value, the goal was to find a set of predictors that would have the greatest influence on heat rate.

There are several unique contributions of this dissertation that have significance, which are listed below.

1. Application of *classical factorial analysis* and *principal variables* in determining a set of predictors for use in modeling the thermodynamic behavior of nuclear power plants. (This contribution is unique and original in thought. No known engineering research has

implemented common factors and principal variables in determining thermodynamic model predictors.)

2. A reduction of the number of input variables for modeling the thermodynamic behavior of a nuclear power plant, and validation of these variables. (Uniqueness and originality of this contribution is specific to the use of common factor space. No known engineering research has used common factorial analysis in reducing the input space for modeling the thermodynamic behavior of a nuclear power plant.)
3. A comparison of three methods of modeling the thermodynamic behavior of a nuclear power plant. (This contribution is unique. No known engineering research has implemented a combination of AI techniques (MR, PLS, and NN) with lower-dimensionality factors (PCA and FACTOR scores) as thermodynamic model predictors.)

Most of the scientific methodology for determining heat rate has been focused on developing mathematical and empirical models based upon a comprehensive thermodynamic analysis of nuclear power plant's systems. Values for thermal efficiency and power output were calculated. Improvements in efficiency to date have mostly been limited to refinements to the existing models. Unfortunately the lack of mobility and portability of these models and/or comparable systems are typical drawbacks since many include site/plant specific modifications. The models themselves are usually quite complicated. Lately, artificial intelligence has emerged as a new tool in the area of thermodynamic optimization, with engineering efforts concentrated on simplifying the modeling process. However, most attempts tend to fit the various AI or math models to the data, while little concern is given to the predictors themselves.

Relating knowledge about the known data by providing useful information to the engineer/scientist to make inferences is one of the objectives of statistical theory -- trying to explain the underlying behavior of the data. It also serves as the basis of this report, attempting to determine relationships among variables. Of the six possible guidelines (see discussion in Section 1.5) for selecting predictors, one item is shared in common – that is correlation. Correlation is also the driving factor behind efficiency model quality. Hence, a natural choice for a variable selection model would be based on correlation. In fact, experimental

data can be fit to a correlation model using factor analysis. A factor analysis model, in addition to its use in the categorization and selection of predictor variables, provides a numerical measure on variance composition. Other methods, including engineering judgement, decisions based upon the correlation matrix, and principal variables, are instrumental in selecting the type of predictors. Figure 1 Fitting Data to a Model outlines the necessary concepts. A set of predictors is diverted to a factor model instead of sent directly to a thermodynamic efficiency model. By way of factor analysis and engineering criteria, a smaller

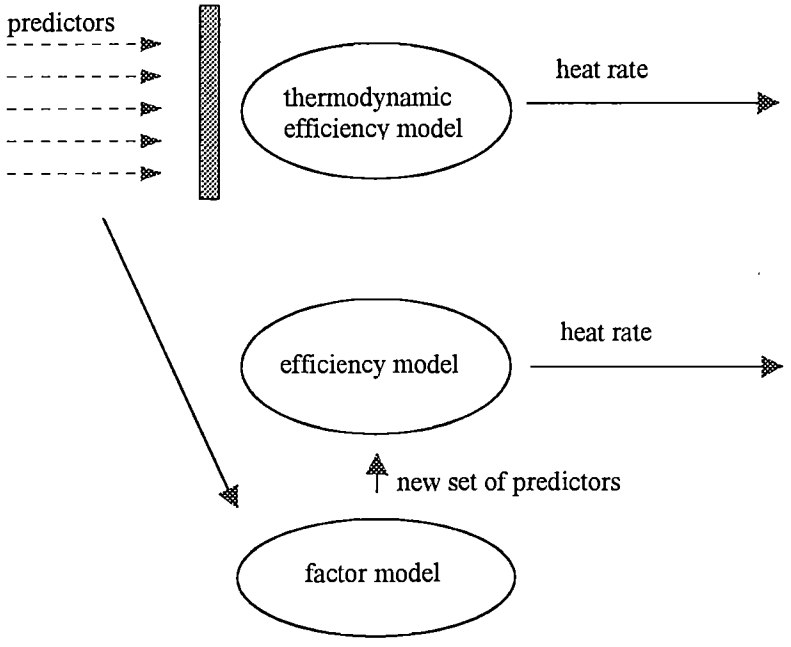

Figure 1 Fitting Data to a Model

number of predictors are selected -- only those needed for the general mapping. To quantify the heat rate, the new predictors are inputs to a simpler design model, e.g., a neural network or partial least squares based efficiency model.

So far the major emphasis has been related to data analysis, which is the major thrust of this dissertation. However, there are two separate categories discussed within this introductory chapter. Section 1.2 begins by covering the basic topics of thermal performance. Ideally one would like to consider the analysis of the data without regard to its application, yet a short background on efficiency and heat transfer calculations related to nuclear plants is quite helpful in understanding the types of variables involved. The discussion

digresses afterwards primarily to data analysis topics: variance and correlation (Sections 1.3.1 and 1.3.2), modeling (Section 1.4), and variable selection criteria (Section 1.5). Section 1.6 provides an outline for the remaining chapters.

1.2 Thermal Performance of Nuclear Plants

Thermal performance is a measure of how well nuclear plants convert thermal energy to electrical energy. An overall indicator of thermal performance in a nuclear plant is heat rate. Heat rate, which is also proportional to the inverse of the plant's efficiency, is the ratio of heat supplied to that of the power output. It is expressed in *Btu/kW-hr*. Naturally, a lower heat rate implies a higher efficiency rating, thus the more efficient the steam cycle and the higher the electrical output for a given input.

The need for optimization of thermal performance of nuclear plants has been clear for some time. Primarily, the cost savings can be very substantial. For example, assume a 1200 *MW(e)* nuclear plant with a thermal efficiency of 34.13 % has an increase of 0.1 %, to 34.23 %. Corresponding heat rates are 10,000 *Btu/kW-hr* and 9970.8 *Btu/kW-hr*, respectively. Power calculations for the original values are:

$$1200\,(MW)\times 1000\,(kW/MW)\times 8760\,(h/yr)\times 10000\,(Btu/kWhr)\approx 1.0512\times 10^{14}\,(Btu/yr),$$

and for the increased efficiency are:

$$1200\,(MW)\times 1000\,(kW/MW)\times 8760\,(h/yr)\times 9970.8\,(Btu/kWhr)\approx 1.0481\times 10^{14}\,(Btu/yr).$$

The small difference in power between the two amounts to only 3.0695×10^{11} *Btu/yr*. However, when an assumed cost of $2.00 per million *Btu* is used, the total savings in one year is roughly 0.6 million dollars. From an economic perspective, it is any wonder that optimization of thermal performance is of major interest. The end result is that even small changes in efficiency may have a significant economic impact.

To examine thermal performance, normally a comprehensive thermodynamic analysis (heat balance) of a power plant's systems is performed. This eventually results in a mathematical model for thermal efficiency

and produces a value for calculated power output. A much simpler heat rate calculation requires only a few parameters and is best explained by looking at the entire plant cycle, which is discussed next.

1.2.1 Rankine Cycle

Both nuclear and coal power plants utilize some form of a Rankine cycle. A Rankine cycle is the ideal simple steam power-plant steady-flow cycle. Figure 2 Rankine below shows typical flow, *pv* (pressure-specific volume), and *Ts* (temperature-entropy) diagrams of a Rankine cycle (Wiley, 1960). The cycle begins when dry saturated steam (working fluid) enters a turbine and expands isentropically to pressure p_2, at which steam is condensed to a saturated liquid, state 3. The saturated liquid is then pumped isentropically back into the heat source at pressure p_4, where at constant pressure it is heated to the saturation temperature and then evaporated to state 1 again -- the working fluid gradually changes from a saturated liquid to a saturated vapor of 100 percent quality. This completes the typical Rankine cycle.

Improvements can be made to the cycle by adding superheat, which is heating the steam leaving the heat

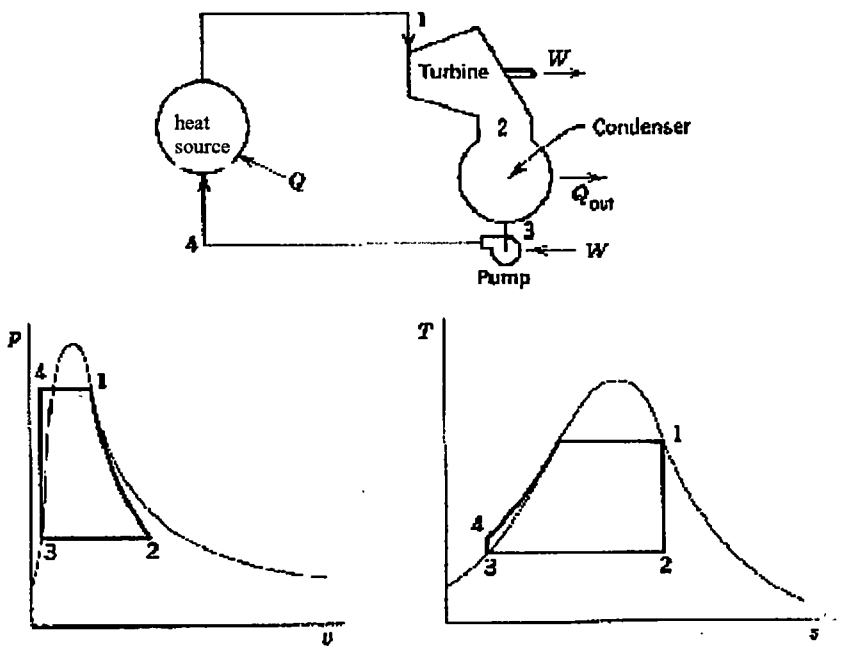

Figure 2 Rankine Cycle[27]

source further at constant pressure. This causes a peak in higher temperature and shift in position at state 1; it also reduces the moisture content of the working fluid as it moves through the turbine. In fossil plants, a combination of boiler and superheater is sometimes referred to as a steam generator. PWR type nuclear plants have a reactor core as the primary heat source with two heat transport loop configurations connected by a heat exchanger called the steam generator. The first configuration is the primary reactor loop, simply a heat transport loop with an all-liquid working fluid. The second heat transport configuration is the secondary loop, a power conversion subsystem. Additional concepts using regenerative cycles employ open and closed feedwater heaters for raising the temperature of the condensate leaving the pump, thus helping in the efficiency of the cycle. Adding a reheat cycle also helps. In a reheat cycle, the steam is removed from the turbine after part of its expansion, heat is added to it at constant pressure, and then it is returned to the turbine for further expansion to the condenser pressure. The prime motivation for this design is similar to superheat, it limits the exhaust moisture content of the turbine steam to a predefined value and raises the average temperature at which thermal energy is added to the working fluid.

1.2.2 Thermal Efficiency

Design enhancements are implemented to increase the thermal efficiency of the cycle, where thermal efficiency, η, is defined as the ratio of the net work output of a cycle to the gross heat added. Work and heat added are expressed in the same units. This can be defined mathematically as

$$\eta = \frac{work_{cycle}}{q_{4\text{-}1}} = \frac{w_{turbine} - w_{in,pump}}{q_{4\text{-}1}} = \frac{(h_1 - h_2) - (h_4 - h_3)}{(h_1 - h_4)} \qquad (1.2.2\text{-}1)$$

where w indicates work, q heat, and h enthalpy. Equation *1.2.2-1* is best analyzed if enthalpy changes for the basic Rankine cycle are applied to each component of a steady-flow system as follows:

$$
\begin{aligned}
\textit{steam generator}\ (p_1 = p_4) &\Rightarrow q_{4\text{-}1} && \approx (h_1 - h_4) \\
\textit{turbine}\ (s_2 = s_1) &\Rightarrow w_{\text{turbine}} && \approx (h_1 - h_2) \\
\textit{condenser}\ (p_3 = p_2) &\Rightarrow q_{\text{out}} && \approx (h_2 - h_3) \\
\textit{pump}\ (s_4 = s_3) &\Rightarrow w_{\text{in,pump}} && \approx (h_4 - h_3)
\end{aligned}
$$

Naturally, the power output and heat output of the cycle are related since the same steam flow passes through the turbine and heating system. Power, by definition, is the rate of generating or using energy; it is determined by performing an energy balance of the turbine, which is the product of the mass flow rate of the working fluid and change in enthalpy of the working fluid entering and exiting the turbine. Hence, changes in flow rate and flow distribution affect the overall efficiency by subsequently varying various enthalpy values throughout the system.

1.2.3 Heat Rate

Heat rate is proportional to the inverse of efficiency and the term can be applied to individual components, i.e., *turbine* and *unit* heat rate. If the steam cycle uses turbine-driven feedwater pumps, the heat delivered to these turbines from the steam generator less the heat returned to the steam generator via feedwater ($q_{4\text{-}1}$) divided by the generator output is called *net turbine heat rate*. If motor-driven feedwater pumps are used in place of turbine-driven feedwater pumps, it is called *gross turbine heat rate*. In nuclear plants, heat generated by the reactor core divided by the gross electrical output is called *gross unit heat rate*. In situations where the net electrical output (turbine output minus unit electrical load) is used instead of the gross electrical output, the heat rate is called *net unit heat rate*. Here, the unit electrical load typically consists of the power consumption of the reactor coolant pumps, condensate pumps, condenser circulating water pumps, etc.

The parameters needed in a simple heat rate calculation are:

1. Final feedwater temperature to determine the feedwater enthalpy, h_4 (*Btu/lb*).
2. Main steam generation pressure to determine the steam enthalpy, h_1 (*Btu/lb*).

3. Feedwater flow ($\dot{m}_{fw}$) in *lb/hr*. This allows for conversion of enthalpy values to *Btu/hr*.

4. Either the gross or net electrical output in *kW*.

Thus,

$$heat\,rate = \frac{\dot{m}_{fw}(h_1 - h_4)}{electrical\,output}\left(\frac{Btu}{kW - hr}\right) \qquad (1.2.3\text{-}2)$$

Equation *1.2.3-2* assumes the mass flow rate of the main steam is the same as the feedwater flow into the steam generator.

In addition, there are corrections for steam quality and additional feed flows. For *BWR* reactor vessels supplemental feedwater may be provided via control rod drive (CRD) flow. The numerator in equation *1.2.3-2* is replaced by

$$\dot{m}_{fw}(h_1 - h_4) + \dot{m}_{CRD}(h_1 - h_{CRD}) \qquad (1.2.3\text{-}3)$$

where $\dot{m}_{CRD}$ is the control rod drive flow rate in *lb/hr* and h_{CRD} is makeup water enthalpy in *Btu/lb*. At particular times *PWR* steam generators may occur blowdown (BD). Blowdown aids in preventing the degradation mechanisms frequent to steam generators, i.e., plugged tubes, deposit accumulations on the tube surfaces, and fouling of the moisture separators. A correction to the numerator in equation *1.2.3-2* becomes

$$\dot{m}_{fw}(h_1 - h_4) + \dot{m}_{BD}(h_1 - h_{BD}) \qquad (1.2.3\text{-}4)$$

where $\dot{m}_{BD}$ is the steam generator blowdown rate in *lb/hr* at the time data is taken and h_{BD} is makeup water enthalpy in *Btu/lb*.

Moisture carried over in the steam generation is corrected for by replacing the enthalpy value, h_1, with

$$h_1 = h_f + x h_{fg} \qquad (1.2.3\text{-}5)$$

where the symbol x is the *quality* of the liquid-vapor mixture. Quality is defined as the fraction, by mass, of vapor in a liquid-vapor mixture. Limiting values range from 0 to 100 %, where 0 % stands for the saturated liquid alone and 100 % or 1.0 is for saturated vapor alone – although the term tends to be confusing because the adjective saturated implies a soaked condition, saturated vapor exists with no liquid present. In cases where the liquid is compressed or the vapor is in a superheated state, the term quality has no meaning. The subscript *f* refers to saturated liquid and the subscript *fg* refers to the difference between the property values for saturated vapor and saturated vapor at the same pressure. The latter is normally referred to as the latent heat of vaporization, defined as

$$h_{fg} = h_g - h_f \qquad (1.2.3\text{-}6)$$

where h_g and h_f are the enthalpy values of saturated vapor and saturated liquid, respectively, at the same pressure. Unfortunately, moisture content leaving the steam generator is not continuously monitored by the plant's data acquisition system. Typically an assumed value, varying between 0.2 - .04 % is used for the moisture content in BWR or PWR heat balances.

Since there are several definitions of heat rate, distinguishing the different energy mechanisms is necessary. Net electric output, W_{net}, is defined by

$$W_{net} = W_{gross} - W_{load} \qquad (1.2.3\text{-}7)$$

where W_{gross} and W_{load} are the gross electrical output and plant electrical load(s), respectively, in *kW*. A simplified heat energy balance for a reactor core is given by

$$Q_{reactor} = Q_{steam_cycle} - Q_{added} + Q_{losses} \qquad (1.2.3\text{-}8)$$

where $Q_{reactor}$ is the heat generated by the core, Q_{steam_cycle} is the heat energy going to the steam cycle, Q_{added} is the heat energy supplied by pressurizer heaters, reactor coolant pumps, etc., and Q_{losses} includes a combination of miscellaneous heat losses from makeup, seal injection, coolant pumps, radiation, etc., (in PWRs this would include heat losses due to blowdown and letdown, for BWRs this would be heat losses from the reactor cleanup system). Equation *1.2.3-8* determines the core thermal power.

Determination of the heat energy for the steam cycle is obtained by a calorimetric calculation of the mass-enthalpy loops. Again, this depends upon the type of reactor, PWR or BWR. Primary energy comes from the core, via the reactor coolant loops (designated A, B, etc.) and secondary power is on the secondary side. A BWR having only one continuous loop, distinguishes the feedwater mass/enthalpy trains as A, B, etc. Even though a PWR has two heat-balance power calculations, the reactor power is controlled on the secondary side. Uncertainties in the secondary power propagate back to the reactor power control with a one-to-one ratio. A false 1 % high feedwater flow reading on the secondary side means that the secondary power is less than what the feed flow and temperature measurements say it is. This implies the plant thermal power on the primary side is 1 % low, and a subsequent decrease in *MW* electrical output generated [10].

Finally, using the above notation, the definitions of heat rate for a PWR and BWR, respectively, are

$$turbine\ heat\ rate \cong \frac{\dot{m}_{fw}(h_1 - h_4) + \dot{m}_{BD}(h_1 - h_{BD})}{W_{gross}}$$
$$\equiv \frac{\dot{m}_{fw}((h_f + xh_{fg}) - h_4) + \dot{m}_{BD}((h_f + xh_{fg}) - h_{BD})}{W_{gross}} \qquad (1.2.3\text{-}9)$$
$$\equiv \frac{\dot{m}_{fw}((h_f + xh_{fg}) - h_4) + \dot{m}_{CRD}((h_f + xh_{fg}) - h_{CRD})}{W_{gross}}$$

$$gross\ unit\ heat\ rate = \frac{Q_{reactor}}{W_{gross}} \qquad (1.2.3\text{-}10)$$

$$net\ unit\ heat\ rate = \frac{Q_{reactor}}{W_{net}} \qquad (1.2.3\text{-}11)$$

If the mass flow rate of the main steam is not the same as the feedwater flow into the steam generator, Equation *1.2.3-9* needs to be modified as follows

$$turbine\ heat\ rate \cong \frac{(\dot{m}_{steam}h_1 - \dot{m}_{fw}h_4) + \dot{m}_{BD}(h_1 - h_{BD})}{W_{gross}} \qquad (1.2.3\text{-}12)$$

where $\dot{m}_{steam}$ is the steam flow rate in *lb/h*. Note however, for most practical purposes the feedwater flow and steam flow are nearly identical [9].

1.2.4 Influence of Plant Parameters on Heat Rate

All the plant parameters influence efficiency to a certain degree. The most noticeable effects of change in heat rate are seen in the generator output and feedwater flow since there is a one-to-one variation in these parameters and heat rate [9]. To a lesser extent, condenser pressure and final feedwater temperature are next in importance. A listing of some cycle parameters and their impact on heat rate are listed in Table 1.

Table 1 List of Cycle Parameters and Their Influence on Heat Rate[9]

Cycle Parameter	Change in heat rate (+ increase; - decrease)	Per Change in Cycle Parameter
Generator output	varies	
Feedwater flow	1% or greater of reactor power	varies
Condenser pressure	from +12 to – 6 %	varies depending upon exhaust pressure (inches Hg Abs.) and last stage bucket (LSB)
Final feedwater temperature	+0.025 %	1°F decrease
Steam pressure	+0.15 %	10 psi reduction
Steam moisture	+0.25 %	1 % increase
Superheat temperature	+0.3 %	10°F decrease
Turbine moisture	decrease in stage efficiency (0.5-0.9 %)	1 % increase
Moisture separators exhaust moisture content	+0.2 %	1 % increase
Moisture separator reheaters (MSRs)	+0.015 %	1°F increase in TTD
Calorimetric inaccuracies	varies	

1.3 Data Analysis Concerns

A significant number of parameters are readily available through a nuclear plant station's data acquisition system. Typically these range in number from a few hundred to thousands. The problem of selecting a smaller number of these for input modeling is a major concern. Too many make modeling difficult and tend to impose high demands on computer resources, while too few may provide insufficient information in making an accurate prediction. While there are several measures available in selecting the fewest variables possible that contain enough information for modeling the state of the system, two most common are based on correlation and variance.

Variance is discussed first as it relates to sets of random variables. Random variables have the same or equal chances or probability of occurrence for each member of a group. Although inferences about one variable from a set of variables are possible, it is variance and correlation of the combined predictor matrix that one is usually interested in.

1.3.1 Variance

If it is assumed the random variable x represents a characteristic or measure from a set of objects, then a number of these characteristics collected simultaneously form a random vector, $\mathbf{x}$, where $\mathbf{x}^t = [x_1, x_2, ..., x_P]$ and has dimension, P, equal to the number of measured characteristics for the object under consideration. A good example is the collection of object variables for an electric motor. Any number of measurements might include: current, voltage, magnetic flux, etc. Here, a single observation is such that $\mathbf{x}^t = [$*voltage, current, flux*$]$. A large number of observations forms a population and yields a matrix, $\mathbf{X}$, where rows are observations and columns are variables. Assuming normality conditions -- based on theoretical considerations of the central limit theorem and sampling distribution theory -- the vector (or matrix for more than one observation), $\mathbf{x}$, can assume a multivariate normal form. The multivariate density function is expressed in Equation (1.3.1-13) as

$$f(\mathbf{x}) = f(x_1, x_2, \ldots, x_P) = \frac{1}{(2\pi)^{\frac{P}{2}} |\mathbf{V}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})' \mathbf{V}^{-1} (\mathbf{x}-\bar{\mathbf{x}})} \qquad (1.3.1\text{-}13)$$

where $\bar{\mathbf{x}}$ is the vector mean and $\mathbf{V}$ is the covariance matrix, which contains the variances and covariances of the random variables. The term $|\mathbf{V}|^{1/2}$ is an important term and is sometimes referred to as the generalized variance. Similar to the univariant case, the generalized variance is a measure of the clustering of points around the mean value. A small generalized variance therefore implies concentration of points in a small area of P space, whereas a larger value indicates expansion into the outer dimensional regions.

Exemplifying the significance of variance can differ for single and multiple variable cases. The magnitude of a single variable's variance is an indicator of whether or not the variable may be used as a predictor. Normally, variables with a variance falling below a predetermined cutoff value are eliminated as predictors. Although this cutoff value may vary, it is usually set at one. Without significant variation these signals would contribute little or no information as predictors. This includes variables that are constant. It should be noted in mathematical models a constant value is important, but in prediction mappings a constant is immaterial, especially if it is converted to deviate or standardized form where it is removed, i.e., has zero

value. When heat rate is the single criterion value, comparing the variance of different types of model predictions gives some indication of the spread of the prediction. A larger variance from one model prediction over another may indicate poor mapping. Also similar is comparing the variance between the prediction and sensor residual for several types of mappings.

In multiple variable cases, the usefulness of variance is seen in space transformation of scores, such as in principal components or factor analysis, and it is also the major determining criteria in partial least squares. Chapter 3 covers the transformation of scores in more detail. One of the best measures to examine when using principal or factor scores is the amount of information contained in the predictor matrix, which is indicated by percent variance explained. This is plotted in what is called a Scree plot. Partial least squares analysis examines the covariance between the predictor matrix and criterion variable. Rotating the scores maximizes the magnitude of the covariance between the predictors and criterion.

Although variance is one type of measure in data analysis, correlation is another indicator. Correlation can be used to show directionality, and the positive or negative direction of movement relative to a position. In these terms positive means increasing in value while negative means decreasing in value.

1.3.2 Linear Correlation

Correlation between two random variables is simply the expected value of their product. It is an indication of how much these two random variables depend upon one another. Since every pair of random variables can be related by the time separation between them, the correlation becomes a function of time separation. If the two variables are from different random processes the correlation function is called crosscorrelation; if the random process is the same it is called autocorrelation. For two measurements defined at two different time instants (or observation points), this expected value depends upon how rapidly the time function changes. For a small Δt (small time instance) it is expected that the measurements will be highly correlated since the time function is not changing rapidly. Correspondingly, for a large Δt it is expected the correlation will be small.

If the covariance matrix is given by

$$\mathbf{V} = \begin{bmatrix} \sigma_1^2 & \sigma_{12}^2 & \sigma_{13}^2 & \cdots & \sigma_{1P}^2 \\ \sigma_{21}^2 & \sigma_2^2 & & & \\ \sigma_{31}^2 & \sigma_{32}^2 & \sigma_3^2 & & \vdots \\ \vdots & & & \ddots & \\ \sigma_{P1}^2 & & \cdots & & \sigma_P^2 \end{bmatrix} \qquad (1.3.2\text{-}14)$$

where σ_{ij}^2 is the covariance between the two components of the random vector $[x_i, x_j]^t$, then the crosscorrelation, r, is expressed as

$$r = \frac{\sigma_{ij}^2}{\sigma_i \sigma_j} \qquad (1.3.2\text{-}15)$$

Linear correlation values are defined such that r varies between -1 and 1, where

- -1 -- indicates negative correlation,
- 0 -- indicates no linear correlation,
- 1 -- indicates high positive correlation.

It must be emphasized that a value of zero for r does not mean that there is no correlation at all. There might actually be a high-degree of non-linear correlation between the variables. Correlation coefficients measure the goodness of fit of the equation actually assumed to the data. Unless otherwise specified, a linear relationship is assumed. In addition, a high correlation coefficient, i.e., near 1 or -1, does not mean a direct dependence of the variables. There could be a high correlation between the number of bugs in a computer program and the number of frogs eaten by the programmer. Such an example is referred to as nonsense or spurious correlation.

Application of correlation is useful in factorial analysis. The purpose of factor analysis is to account for intercorrelations among variables through a common set of factors, although there will also be a set of unique factors, one for each variable, which do not account for the intercorrelations. When correlation is applied to the predictor matrix, determination of possible linkages between variables is possible. In fact, variable selection criteria for prediction modeling is highly dependent upon the correlation between variables. This is discussed in more detail under Section 1.5.

1.4 Prediction Modeling

Given an initial set of p input variables (see Figure 3), there are several multivariate analysis techniques available in reducing the *input space* or *dimensionality* of the problem. The central approach is to assume a mapping of predictor variables to a set of criterion variable(s) using a black box. The black box represents a general mapping and is not specified. Hence, the objective is to obtain k predictor variables based on achieving the best mapping to criterion variables via the black box. If p-k (p minus k) of these variables

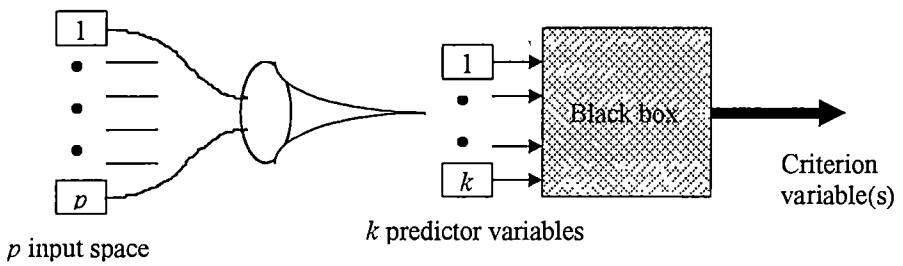

Figure 3 Funnel Diagram

carry little or no information about the characteristics represented by the original p variables, then k variables may be sufficient to represent most of the information contained in the original set of p variables. In some cases the dimension of k may be greater than p; this involves a transformation of the variable space to another type of space (e.g. observation or factor space), however, the higher space *cannot* be filled from a space of lower dimensionality, i.e., the dimensionality of the hypervolume is equal to the maximum number of linearly independent vectors in that space. The range of any linear transformation that takes a system in p dimensions into a system in k dimensions (where $k>p$) cannot exceed p (Green, 1976). It is desired that k is less than p.

If the modeling objective, via the black box, is to include heat rate as the criterion variable, then an overall optimization goal is to identify predictors needed for the desired output. Note: heat rate is proportional to the reciprocal of plant efficiency, and thus optimization of heat rate is improving plant efficiency.

Predictors must contain information that relate to heat rate through some process. They must be changeable, i.e., capable of adjustment so as to provide a related change in the criterion variable. In addition, to provide robustness the model must reflect changing operating conditions, thus predictors must also contain information that reflects the current operating environment. To satisfy the above conditions, its necessary to separate model inputs into two groups:

- state variables – predictors that provide information reflecting the current state of the plant
- setpoint variables – predictors whose value can be adjusted in order to affect process outputs

State variables are normally not controllable, but reflect the status of process outputs.

Having a set of predictor variables and a criterion variable provides the means for development of a process optimization model. Since output values change may depending upon the different types of models employed in the mapping, e.g., neural networks, multivariate methods, etc., model quality should be based on a comparison between the actual plant output and the model's output. Usually a normalized sum of squared error (NSSE) is taken as the evaluation measure. Secondly, and of great importance, is the directional relationships between predictors and criterion variables. Changes in setpoint variables must reflect a proper directional change in the efficiency. Directional changes can be evaluated using correlation between the individual inputs and output. Good correlation is the deciding factor over low NSSE since the goal of process optimization is to provide information about how to alter control setpoints to obtain improved efficiency.

1.5 Variable Selection Criteria

On the basis of the literature survey, there are at least six input selection rules considered in improving plant efficiency, although some of these are highly dependent upon the type of model employed in the mapping than others. These are enumerated and discussed as follows:

1. Enough state variables must be selected for inclusion into the model of sufficient information to allow for an accurate prediction of the heat rate over the plant's operating range. However, a state variable's contribution must not destroy model fidelity nor dampen model output sensitivity from other setpoint variables.
2. Predictors must combine to form a set of interacting process input variables. Predictors that have a one-to-one correlation with heat rate (input-output correlation) provide no process information, and in many cases may destroy sensitivity from other setpoints. In some situations moderately high intercorrelations between predictors may limit the ability of certain setpoint variables to effectively change the heat rate.
3. Setpoint variables must contribute to the system process. At first thought this might seem trivial, however, in data driven models (e.g. neural networks) setpoints considered in the primary loop of a reactor may contribute well to model sensitivity during training. In actual plant conditions the importance of these variables may be diminished and provide no information transfer between the primary-to-secondary loop. In this case it is best to have two stages, setpoints mapped in the primary to some exchange variable, and another mapping of the exchange variable to the secondary loop.
4. Predictors, in general, must contribute to the system process. Although this is idealization of the above, predictors that provide little or no correlation with heat rate contribute nothing to the optimization procedure.
5. Predictors must not have high inner correlations (input-input correlation). There are two reasons for this. First, identical predictors that contain the same information cause co-linearity problems during neural network training; weights may favor one or the other predictor depending upon the

initial random distribution of weights. In short, sensitivity values essentially switch between these inputs. Secondly, redundant variables provide no additional process information.

6. Setpoint variables must be given preference over process variables. Optimization is ineffective unless levels can be changed in order to influence heat rate.

1.6 Organization of the Dissertation

The introduction of thermal performance of nuclear plants along with discussion of a first principle model of heat rate is given in Chapter 1. Data analysis concerns along with prediction modeling and variable selection criteria are also presented in Chapter 1.

A literature review of current and past topics related to nuclear power plant thermal performance is given in Chapter 2. Included in particular are past works dealing with applied artificial intelligence towards heat rate prediction modeling. Chapter 3 continues with the literature review from a mathematics perspective. The basic theory of factor analysis, some history and highlights with important discussions are included. The basic theory of principal components, partial least squares, and multiple regression are also included in Chapter 3.

Chapter 4 sets in motion much of the exploratory work with state variables and different model types. It was the starting stage, and research was at a rudimentary level. Beginning with Chapter 5, a great deal of the background and foundation analysis with setpoint and state mappings finally came into perspective. While Chapter 5 presents results from heat rate prediction modeling using setpoint predictors, it also added several puzzling questions. Summarizing important highlights from earlier attempts at modeling using two separate stages, Chapter 6 examines a single mapping layer model. It also adds the concept of reference directionality – a very powerful method in judging model fidelity. At this point it would seem explanation of all the results were accounted for. Unfortunately, this wasn't the case. Finally, Chapter 7 covers the topic of quasi-static range of a predictor. It provides a basis for understanding perturbation model

directionality and why some models may fail to indicate proper directionality. Chapter 8 concludes with a summary of the important findings.

The computer program codes are given in the Appendix.

Chapter 2

Review of Literature

2.1 Preface

There are three sections in this chapter. Section 2.2 and Section 2.3 deal with selected topics ranging from thermal performance of nuclear power plants to data analysis, and finally, modeling using artificial intelligence. Discussion is primarily limited to the latest developments that are of interest. Many references are given and the reader may review those if more detailed information is needed. The last section, Section 2.4 concludes with background information and a history of factor analysis. It is somewhat disparate from the two earlier sections in that it is not directly connected to nuclear thermal performance, but is a separate unique contribution to this report. Again, numerous references are listed.

Much of the mathematics, e.g., factor analysis, principal components, partial least squares, etc., is an integral part of any literature review. It is purposely left out here. However, beginning with Chapter 3, *Review of Mathematics*, the entire chapter is devoted to mathematical formulas and expressions. Hopefully Chapter 3 presents a sufficient numerical mathematics background along with many references of the forthcoming multivariate data analysis techniques.

2.2 Current Nuclear Plant Thermal Efficiency Perspectives

Nuclear plants differ slightly from fossil plants in that thermal enhancements derive their value by increasing the net electrical energy generated for dollars spent on maintenance, operations, and fixed costs. In a fossil plant thermal performance savings from fuel costs can amount to large sums of money, whereas in a nuclear plant these are usually minimal. Plant thermal performance covers many areas, including but not limited to areas such as monitoring and trending, identifying and investigating megawatt losses, identification of remedial action to correct faulty conditions, and finding potential changes and modifications to recover and enhance electrical output production. The start of a thermal performance

program normally begins with a comprehensive thermodynamic analysis (heat balance) of a power plant's systems, which eventually results in a mathematical model for thermal efficiency, or heat rate. A good example is the Thermal Performance Program (TPP) at the Tennessee Valley Authority (TVA) Watts Bar Nuclear Plant (Wood, 1998). The TPP includes approximately 2300 data points associated with the balance of plant systems and the reactor coolant systems. The entire system is analyzed by using the first and second law of thermodynamics, computing values for heat and work, and connecting component and heat flow paths together. Not only is the heat rate of the plant determined, but information concerning sub-systems is also provided.

Because most models are a type of component focus approach, which concentrate on individual equipment items and then couple them together forming the plant model, the Electric Power Research Institute (EPRI) has published two important documents within the last year that help understand the essentials to monitoring components that have a significant influence on thermal performance. These are Volume I and Volume II of the *Thermal Performance Engineer's Handbook* (EPRI, 1998). Volume II contains a section devoted mostly to individual plant equipment. Each section discussing the equipment contains an explanatory description along with some preliminary mathematical performance calculations that impact heat rate. Another part includes a short listing describing typical problems and associated degradation mechanisms. At the end of each chapter covering the equipment is shown a complete list of references.

Volume I explains the fundamentals typical of any nuclear thermal performance program. It covers the basic elements of baseline and modeling, performance goals, monitoring and trending, search and recovery, and communication. Trending is an important aspect of thermal performance monitoring and is being used more and more in every day nuclear plants. Trending is the monitoring of key variables in the TPP. A plant process or personal computer is used to track and monitor the data. While it might appear thermal performance improvements are limited to equipment design, trending is another approach to increasing the thermal efficiency through optimization of existing parameters. This draws upon several types of information technologies' methods. These include: improving the accuracy and precision of process

measurements, by detecting faults in the process earlier, by increasing operator awareness, and as a decision aid in complex and possibly critical situations (Svein, 1999).

Before applying any new technologies, and just prior to any type of data analysis, the first step begins with a basic understanding of the major cycle parameters affecting heat rate. Performing a first-principles calculation of the steam cycle is helpful. A nuclear plant can be modeled as a whole with the inputs of steam flow and blowdown, outputs of feedwater flow and generator production (Kavaklioglu, 1994). Shown earlier in the introductory part of Chapter 1 are mathematical equations using this method. However, a more detailed model will include the major cycle parameters: main steam pressure, moisture content or superheat of steam, feedwater heater TTD, reheater TTD, change in condenser pressure, LP and HP turbine efficiencies and inlet moisture (EPRI [9]). Yet even with a fairly good model, variations in duplicate measures, inconsistencies in measurements, unexplainable *MW* losses, etc., have a significant impact on model output.

Data validation is also very important. A good thermal performance program rests on a foundation of proper monitoring and collection of data. Included are both the data requirements and data validation. Three parts of any data requirements should include (EPRI [10]) the proper selection of parameters for performance measurements, required accuracy, and sampling frequency. Variables that have a significant impact on heat rate should have a high monetary value. Those with lower impact on heat rate should be examined as to their potential benefit, and should be removed if their uncertainty of measurement is greater than their ability to predict a change in heat rate. The uncertainty is usually stated in terms of precision that includes repeatability and resolution, and bias, which is an accuracy range around the value. Frequent calibration of sensors can reduce bias and usage of multiple sensors is beneficial in detecting deviations of the data. Sometimes data uncertainty will also include a stochastic component, which is the probability the measurements fall within the precision given. The last data requirement is sampling frequency. Data collected at one-day intervals will only be sufficient in forecasting monthly trends. A higher sampling frequency (at least every half-hour) is needed to detect daily changes. An effective sampling frequency

should follow the Nyquist or folding frequency criteria. This states the maximum frequency in a given signal should be less than or equal to one half the reciprocal of the sampling interval (Upadhyaya, [47]).

Finally, validation of the data is necessary. A number of methods to detect and determine where instrument or sensor failures occur include analytical redundancy calculations (also called *reconciliation*), limit checking, calibration, and a variety of statistical methods. Analytical redundancies are basic mass and energy balance calculations. They are applicable in finding decreases in component efficiency and places where *MW* flow losses occur. Limit checking is a procedure that examines data to see if it falls outside of a certain range. This method can spot gross sensor failures. Calibration is the comparison of the measurement to a standard. As mentioned earlier, calibration is useful in reducing sensor bias, except it does not remove sensor drift. Lastly, there are several statistical methods for rejecting a reading. One in particular is given by Chauvent's Criterion (EPRI [10]). It is based on the ratio of maximum acceptable deviation to the sample standard deviation. Many other probabilistic methods of signal analysis and systems analysis use sampling distributions and confidence intervals, hypothesis testing, etc. (Cooper, 1986).

All of the above topics are, and always will be, an important part in any thermodynamic performance program. However, much of the main interest today is centered on simplifying the modeling process.

2.3 New Approaches to Modeling, and Merger of AI

Information technologies' methods are being partially exploited today in nuclear power plants. For example, plant sub-systems are available that calculate and monitor core power distributions, and thermal-hydraulics on-line monitoring (TVA's Watts Bar PEDS) is available from both LAN and WAN connections. In addition, an increasing amount of work in information technologies has been done with artificial intelligence. Many of the AI methods focus on optimization or detection of smaller plant sub-systems or individual components. AI applications include the usage of neural networks, expert or knowledge-based systems, genetic algorithms, etc. Just some of the many practical applications of AI

systems are addressed in detail in many papers and reference journals, such as those by Agogino [1], Black [4], Fowler [13], Frogner [14], Gribok [20], Guo [21, 22], Iida [26], Kavaklioglu [29], Munoz [34], Sherin [37], and Sreedhar [39]. Although less consideration appears to have been paid to the overall optimization of energy produced from a nuclear power plant, much of the present-day AI work attempts to change this direction by applying the technology to the monitoring of plant heat rate.

For any type of model to determine plant thermal efficiency accurately and to do proper *MW* accounting, accurate and reliable data is needed. Unfortunately, many thermal performance programs are susceptible to inadequate instrumentation readings, possible instrument drifts, look-up table interpolation errors, and simply incorrect direct measurements of steam flows. AI methods combined into signal validation software, such as ISCV (Xu, 1999) and PEANO (Fantoni, 1998), are a necessary prerequisite for a successful implementation of a thermal-performance monitoring system. Most of these signal validation systems use the large number of parameters that are readily available through the plant station data acquisition systems as input to an auto-associative neural network. Even with data anomalies, the outputs of the neural networks provide drift-free, best-estimate parameter values.

Neural networks can also provide predictive information. Generally auxiliary steam loads must be known to a reasonable degree of accuracy, because most of these can be a substantial percentage of full-power steam flow (Sherin, 1998). Just as in the case of signal validation, a large number of parameters that are readily available through the plant station data acquisition systems are inputs to an inferential neural network. Here, the neural networks can provide virtual flow measurements based on learned relationships with other data measurement points. With features like fault tolerance that make the network suitable for processing noisy and incomplete data, the neural network outputs are improved substantially over the original measurements. These virtual measurements can be used as inputs to yet another AI system. A good example is feedwater flow. Feedwater flow is a key parameter that is normally suspect in reactor power calibration problems. By excluding its influence in an AI model and substituting its value with virtual flow measurements, the AI model becomes independent of feedwater measurement problems. Much of the inferential sensing work using neural networks in feedwater measurements started with

Kavaklioglu [29] in 1994. However, newer AI methods implementing regularization (Gribok [20]) are not only providing more consistent results, but solving some of the problems associated with ill-posed data matrices consisting of highly correlated sensor values. Regularization is a method that provides a more consistent feedwater flow drift prediction and avoids instability problems associated with noise.

The ultimate goal, however, is to increase plant efficiency. Whether a number of sub-systems are modeled and connected together or several inputs are directly fed into a single model, the idea of the AI model is to capture process information between the inputs and heat rate. Sensitivity studies between input-output relationships (Guo, 1992) can determine predictors that have the most influence on plant efficiency. By selectively reducing the number of predictors to only the most important variables, inducing small perturbations around normal operating levels, changes in the magnitude of efficiency can be evaluated. At Ontario Hydro Nuclear, application of AI technology to thermal performance resulted in as few as 10 predictors being selected (Sherin, 1998). However, this poses many questions as to what are the right input variables to use as predictors. This is an expanding topic of interest, especially in AI. Newer methods are being developed, such factor analysis, to examine variable selection. The use of factorial analysis in the determination of possible linkages between variables is discussed in the next section. It not only represents new insight into the area of variable selection, but the method may contribute as an additional tool in the expanding field of artificial intelligence.

While a proliferation of studies using neural networks for boiler optimization was being established in the 1990s (Fowler [13], Gorzelnik [17], and Munoz[34]), around the same time most of the work in performance monitoring of nuclear power plants using AI began. In 1994 Kavaklioglu ([28]) completed work on estimation of plant heat rate and thermal power using simple neural networks. Data for the analysis was taken from four-loop Westinghouse PWRs. Fifteen signals were input to a neural network with the criterion variable heat rate. Results were shown to have good agreement between the calculated heat rate from the plant model and estimated values from the neural network prediction. At around the same time, Black ([4]) applied the process modeling to plant data from TVA's Kingston Fossil Plant. Neural networks provided the modeling tool in studying the sensitivity of input variable perturbations.

Other AI methods were also being implemented. Some of the earliest efforts using expert systems for thermal performance began with HEATXPRT (Heat Rate Degradation Expert System Advisor) (Agogino, [1]). Again using expert systems technology, EPRI developed a Nuclear Thermal Performance Advisor (NTPA). It is an interactive heat rate diagnostic expert system consisting of a PC program and *Thermal Performance Diagnostic Manual* (EPRI [9]). NTPA provides nuclear engineers with identifying causes of *MW* losses in nine diagnostic areas: condenser, high-pressure feedwater heaters, high-pressure turbines, low-pressure feedwater heaters, low-pressure turbines, main steam system, moisture separator reheaters, nuclear steam supply system, and turbine control valves. In addition, NTPA is a powerful diagnostic tool for analyzing plant data, complete with plotting and report capabilities.

Optimization of thermal performance of a nuclear plant is motivated by economics. While mentioned only briefly here, particular attention paid to maintenance is of high importance because of deregulation of the power industry. Since AI methods applied to data acquisition, analysis, resolution and data management are becoming almost routine, the introduction of AI to maintenance and reliability is expanding. The University of Tennessee conducts a Maintenance and Reliability Engineering Certification Program that incorporates many of the AI methods, such as neural networks and fuzzy logic, into solving maintenance related problems in nuclear power plants. Nuclear plants themselves are establishing permanent facilities on-site dedicated to improving safety and equipment life extension. A good example is the Palo Verde steam generator facility established in 1994 [43]. The facility uses a number of technology advances that aids in improving steam generator inspection and integrity assessment activities.

Like most new technologies, advanced AI techniques and methods developed in-house are not distributed publicly. Many AI enhancements may already exist today in nuclear plant thermal performance programs around the nation. It is hoped some of the concepts have been explained, and that more promising AI work becomes available in the future for public dissemination.

2.4 Factor Analysis Background

Factor analysis is primarily a statistical tool for dealing with observed data. It consists of a number of very simple mathematical techniques for analyzing the relations among a set of random variables. Somewhat subtler, is that factor analysis provides for the resolution of a set of variables linearly in terms of a number of factors. In *classical factor analysis*, this resolution is accomplished by the analysis of the correlations among the variables (Harmon, 1976). If the factors were determined in sequence so that each successive stage would relate to a maximum of the variance, the optimal solution would be called *principal components.* Considering a more general view, factor analysis can be thought of as a model, a very simplistic model of the correlation matrix.

Origins of factor analysis dates back to 1904, when Charles Spearman was beginning work on a "two-factor theory" of general intelligence. It should be noted Spearman did not use the term *factors* explicitly, however, his life-long devotion to the development of factor analysis, regards him as the father of the subject. Many contributed to the theories and mathematical foundation, including Karl Pearson, L. L. Thurstone, Hoteling, Kelly, and a host of other dedicated individuals (Harman, [23]). In the early days, most of the interest in the field centered mainly on common factors. Researchers hoped that by conjecturing a set of *common* factors that were shared amongst all the variables, the intercorrelations between the variables could be accounted for. There will also be *unique* factors, which do not attribute to the intercorrelations, but account for the fact that the intercorrelations are less than unity (Cureton, 1983). They account for the remaining variance and error. It might seem ironic that the first fundamental factor the early researches were seeking was intelligence, especially since a majority of the mathematical tools in this report are based on *artificial intelligence.* However, the common factors are to be interpreted with reference to the original data. Cureton best illustrates this with the following explanation:

> "If the variables are psychological and/or educational tests, the common factors will be interpreted as the principal underlying abilities, such as verbal comprehension, numerical ability, abstract reasoning, visual space manipulatory ability, knowledge of the natural sciences, etc. If the variables are athletic and gymnastic measures, the common factors will be such things as static

strength (as in chinning or push-ups), explosive strength (as in hitting or jumping), speed of movement, eye-hand coordination, endurance, and the like. If the variables are yields per acre of various crops, the common factors will be such things as grains, fruits, vegetables, and the like. The variables may be such things as complex measurements of a collection of boxes, for example:

- Perimeters (of top, side, end)
- Diagonals (of top, side, end)
- Squares (of length, width, height)
- Logarithms (of length, width, height)
- Volume
- Longest diagonal (from northwest top corner to southeast bottom corner)

In this case the common factors will be length, width, and height in relative terms, with a general factor interpretable as size, best represented by volume and longest diagonal but not identical with either."[i]

Since all the variables discussed in this report come from Arizona's Public Service *Palo Verde Nuclear* Generation Station data acquisition system, determining the principal underlying abilities is a formidable task, if not impossible. The first initial concepts imagined a set of common factors that represented states of the data, such as dynamic, steady state, intermittent, binary, cyclic, slow-varying drift, spike, etc. However, this interpretation is too general and is more applicable to a variety of engineering values, not just nuclear data. In most cases the factors are just denoted *hypothetical constructs* because they are random variables that cannot be observed or counted or measured directly. Occasionally the term *latent variables* is given in the literature, however, caution must be exercised with this nomenclature. The definition of latent variables should express the general meaning -- referring to all factors and not only those associated with eigenvalues or characteristic roots from a principal factor solution. Most factors have different meanings depending upon how they are calculated and even though they are presumed to exist in

[i] from Cureton, pages 3-4.

the experimental data, their reality is always open to argument. The later premise, while philosophical, is best stated as either of two propositions of a syllogism, from which the conclusion is drawn. Cureton and D'Agostino [7] best summarize the point by concluding that *if* the factors were real, *then* they would account for the correlations in the sample, however, the converse does not necessarily follow. One can contemplate the point if the converse is not true, then what? Unfortunately, there will always remain some particulars to study and information contemplated by those resolutions. And although effort has been done to study the factors and their classification, most attention is given to the factor loadings -- and sometimes to the factor scores.

Factor loadings, or weights, measure the degree of generalizability found between each variable and each factor. Since the loadings reflect quantitative relationships, the farther the factor is from zero, the higher degree of generalization from that factor to the variable, and vice versa (Gorush, 1983). Information about relationships among the variables is obtained by:

1. comparing loadings of the same variable on several different factors,
2. and, comparing loadings of the same factor on several different variables.

While attempts have been made in the past to estimate factor loadings directly from the observed values (e.g., Holzinger (1944), Saunders (1950), Whittle (1952), Young (1941) [23]), throughout this report the variables are small standardized deviates and the correlation's among them are employed in the calculations to estimate the factor loadings. Once estimates of loadings are known, there are several direct solutions for calculating factor scores. *Factor scores* are defined as measurements of the factors.

Just as standardized scores can be implemented in the modeling process, use of factor scores is another alternative. If the intercorrelations among the variables are accounted for through a common set of factors, choosing predictors with a common set of linkages with heat rate should allow the AI model to capture relevant process information and ultimately, establish the best heat rate. Advantages of using fewer

dimensional factor scores as predictors in modeling are several. There would be a definite reduction in computer resources and calculation times would be much shorter.

Chapter 3

Review of Mathematics

3.1 Mathematics Preliminaries

Model variables are normally classified as either of two types, predictor or criterion. Predictor variables are the independent variables and are represented by X. The dependent variable is the criterion; it is represented by Y. For N observations, the mean of a variable, X, is defined as

$$\overline{X}_j = \frac{\sum X_{ji}}{N} \tag{3.1-16}$$

where j indicates a particular variable, 1 ... n, and the index i is employed to designate individuals or observations, 1 ... N. When the summation has no index, it is assumed to be over all values of the variable.

The deviate is defined as

$$x_{ji} = X_{ji} - \overline{X}_j \tag{3.1-17}$$

and the standard deviate for a variable j and individual i is given by

$$z_{ji} = \frac{x_{ji}}{\sigma_j} \tag{3.1-18}$$

where σ_j is the standard deviation of the variable. Hence, the sample variance of a variable is

$$\sigma_j^2 = \frac{\sum x_{ji}^2}{N} \tag{3.1-19}$$

Defining the small standard deviate as

$$z_{ji}^{*} = \frac{z_{ji}}{\sqrt{N}} \tag{3.1-20}$$

the correlations, r, between any two variables are now simplified as

$$r_{jk} = \sum z_{ji}^{*} z_{ki}^{*} \tag{3.1-21}$$

In matrix form the correlation matrix, $\mathbf{R}$, is just

$$\mathbf{R} = \mathbf{Z}^{*\prime}\mathbf{Z}^{*} \tag{3.1-22}$$

where $\mathbf{Z}^{*}$ is defined as a matrix of small standard deviates such that

$$\mathbf{Z}^{*} \in \Re^{N\times n} \Leftrightarrow \mathbf{Z}^{*} = \left[\mathbf{z}_1^{*} \quad \cdots \quad \mathbf{z}_k^{*} \quad \cdots \quad \mathbf{z}_n^{*}\right];\ \mathbf{z}_k^{*} \in \Re^{N\times 1}$$

3.2 Multiple Regression

The most popular method of predicting a univariate variable with a set of predictors is the linear regression or least squares model, which may be stated as

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e} \tag{3.2-23}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}; \ \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ 1 & x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Nn} \end{bmatrix}; \ \mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}; \ \mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}$$

where $\mathbf{X}$ is the observed $N \times n+1$ matrix of nuclear data, $\mathbf{y}$ is the observed N-vector criterion variable (heat rate), $\mathbf{b}$ is the $n+1$-vector of regression coefficients (including the intercept term), and $\mathbf{e}$ is the N-vector of independent and normally distributed errors. In this particular case, the independent variables are contained within the matrix $\mathbf{X}$ while the dependent variable is the vector $\mathbf{y}$. The matrix $\mathbf{X}$ also includes a column of ones for estimating the intercept b_o. Alternatively, the column partitioning of the matrix $\mathbf{X}$ is a collection of column vectors:

$$\mathbf{X} \in \Re^{N \times n+1} \Leftrightarrow \mathbf{X} = \left[\mathbf{1}, \mathbf{x}_1, \cdots, \mathbf{x}_k, \cdots, \mathbf{x}_n\right]; \ \mathbf{1}, \mathbf{x}_k \in \Re^N$$

The vector $\mathbf{1}$ is a $N \times 1$-unit column vector and the independent variables are the column vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$.

In multiple regression, the set of predicted values $\hat{\mathbf{y}}$ is

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{b} \qquad (3.2\text{-}24)$$

The vector of prediction errors is

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} \qquad (3.2\text{-}25)$$

and using the least-squares criterion that minimizes the sum of the squared errors,

$$
\begin{aligned}
f &= \mathbf{e}'\mathbf{e} \\
&= (\mathbf{y}-\hat{\mathbf{y}})'(\mathbf{y}-\hat{\mathbf{y}}) \\
&= (\mathbf{y}-\mathbf{Xb})'(\mathbf{y}-\mathbf{Xb}) \\
&= \mathbf{y}'\mathbf{y}-\mathbf{b}'\mathbf{X}'\mathbf{y}-\mathbf{y}'\mathbf{Xb}+\mathbf{b}'\mathbf{X}'\mathbf{Xb} \\
&= \mathbf{b}'\mathbf{X}'\mathbf{Xb}-2\mathbf{y}'\mathbf{Xb}+\mathbf{y}'\mathbf{y}
\end{aligned}
\qquad (3.2\text{-}26)
$$

Note that $\mathbf{b}'\mathbf{X}'\mathbf{y} = \mathbf{y}'\mathbf{Xb}$ (it is the same scalar). Minimizing the function f with respect to the vector $\mathbf{b}$,

$$
\frac{\partial f}{\partial \mathbf{b}} = \mathbf{0} \qquad (3.2\text{-}27)
$$

Equation *3.2-27* uses symbolic differentiation of a function with respect to a vector. It involves partial differentiation of the function with respect to each entry of the vector. The notation of whether to use $\partial\mathbf{b}$ or $\partial\mathbf{b}'$ depends upon how the final vector of derivatives is displayed, i.e., whether it is in column or row form, respectively.

Letting $\mathbf{A} = \mathbf{X}'\mathbf{X}$ (symmetric matrix) the column partitioning of $\mathbf{A}$ is

$$
\mathbf{A} \in \Re^{(n\times 1)\times(n+1)} \Leftrightarrow \mathbf{A} = \left[\mathbf{a}_1, \cdots, \mathbf{a}_k, \cdots, \mathbf{a}_{n+1}\right],\ \mathbf{a}_k \in \Re^{n\times 1}
$$

and assuming $\mathbf{w} = \mathbf{b}'\mathbf{Ab}$, this can be written out explicitly as

$$
\begin{aligned}
w = {} & b_0 a_{11} b_0 + b_0 a_{12} b_1 + b_0 a_{13} b_2 + \cdots + b_0 a_{1,n+1} b_n \\
& + b_1 a_{21} b_0 + b_1 a_{22} b_1 + b_1 a_{23} b_2 + \cdots + b_1 a_{2,n+1} b_n \\
& + b_2 a_{31} b_0 + b_2 a_{32} b_1 + b_2 a_{33} b_2 + \cdots + b_2 a_{3,n+1} b_n \\
& + \cdots \\
& + b_n a_{n+1,1} b_0 + b_n a_{n+1,2} b_1 + b_n a_{n+1,3} b_2 + \cdots + b_n a_{n+1,n+1} b_n
\end{aligned}
$$

The partial derivative of w with respect to the first element b_0 is

$$\begin{aligned}\frac{\partial w}{\partial b_0} &= 2a_{11}b_0 + a_{12}b_1 + \cdots + a_{1,n+1}b_n + b_1a_{21} + \cdots + b_na_{n+1,1} \\ &= 2a_{11}b_0 + 2a_{12}b_1 + \cdots + 2a_{1,n+1}b_n \\ &= 2\mathbf{a}_1'\mathbf{b}\end{aligned}$$

The partial derivative of w with respect to the first element b_1 is

$$\begin{aligned}\frac{\partial w}{\partial b_1} &= b_0a_{12} + a_{21}b_0 + \cdots + a_{2,n+1}b_n + b_2a_{32} + \cdots + b_na_{n+1,2} \\ &= 2a_{21}b_0 + 2a_{22}b_1 + \cdots + 2a_{2,n+1}b_n \\ &= 2\mathbf{a}_2'\mathbf{b}\end{aligned}$$

Similarly, the other partial derivatives can be written as

$$\begin{aligned}\frac{\partial w}{\partial b_2} &= 2\mathbf{a}_3'\mathbf{b} \\ &\vdots \\ \frac{\partial w}{\partial b_n} &= 2\mathbf{a}_{n+1}'\mathbf{b}\end{aligned}$$

Hence, the symbolic derivative of w is

$$\frac{\partial w}{\partial \mathbf{b}} = \begin{bmatrix} 2\mathbf{a}_1'\mathbf{b} \\ 2\mathbf{a}_2'\mathbf{b} \\ 2\mathbf{a}_3'\mathbf{b} \\ \vdots \\ 2\mathbf{a}_{n+1}'\mathbf{b} \end{bmatrix} = 2\mathbf{A}'\mathbf{b} = 2\mathbf{A}\mathbf{b} = \mathbf{2X'Xb} \qquad (3.2\text{-}28)$$

By a similar rationale, letting **u** = **y'Xb**, this can be written out explicitly as

$$\begin{aligned} u = {} & y_1 b_0 + y_1 x_{11} b_1 + y_1 x_{12} b_2 + \cdots + y_1 x_{1n} b_n \\ & + y_2 b_0 + y_2 x_{21} b_1 + y_2 x_{22} b_2 + \cdots + y_2 x_{2n} b_n \\ & \qquad + \cdots \\ & + y_N b_0 + y_N x_{N1} b_1 + y_N x_{N2} b_2 + \cdots y_N x_{Nn} b_n \end{aligned}$$

The partial derivatives of u with respect to the first element b_0 and the second element b_1 are

$$\begin{aligned} \frac{\partial u}{\partial b_0} &= y_1 + y_2 + \cdots + y_N \\ &= \mathbf{y'1} \\ \frac{\partial u}{\partial b_1} &= y_1 x_{11} + y_2 x_{21} + \cdots + y_N x_{N1} \\ &= \mathbf{y'x_1} \end{aligned}$$

Similarly, the other partial derivatives can be written as

$$\begin{aligned} \frac{\partial u}{\partial b_2} &= \mathbf{y'x_2} \\ &\vdots \\ \frac{\partial u}{\partial b_n} &= \mathbf{y'x_n} \end{aligned}$$

Thus

$$\frac{\partial u}{\partial \mathbf{b}} = \begin{bmatrix} \mathbf{y}'\mathbf{1} \\ \mathbf{y}'\mathbf{x}_1 \\ \mathbf{y}'\mathbf{x}_2 \\ \vdots \\ \mathbf{y}'\mathbf{x}_n \end{bmatrix} = (\mathbf{y}'\mathbf{X})' = \mathbf{X}'\mathbf{y} \qquad (3.2\text{-}29)$$

The minimizing function simplifies to

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{b}} &= 2\mathbf{X}'\mathbf{X}\mathbf{b} - 2\mathbf{X}'\mathbf{y} = \mathbf{0} \\ &\Rightarrow \mathbf{X}'\mathbf{X}\mathbf{b} = \mathbf{X}'\mathbf{y} \end{aligned} \qquad (3.2\text{-}30)$$

Finally, solving for the regression coefficients that minimizes the function f implies

$$\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \qquad (3.2\text{-}31)$$

The linear estimating equation for predicting values of $\mathbf{y}$ is now

$$\begin{aligned} \hat{y}_i &= b_0 + b_1 x_{i1} + b_2 x_{i2} + \cdots + b_n x_{in} \\ &= b_0 + \sum_{j=1}^{n} b_j x_{ij} \end{aligned} \qquad (3.2\text{-}32)$$

3.2.1 Multiple Regression using Standardized Data

Working with small standard deviates simplifies, but also changes the computation of *3.2-31* slightly. Now the column partitioning of the matrix $\mathbf{Z}^*$ is a collection of column vectors without an $N \times 1$-unit column vector:

$$\mathbf{Z}^* \in \Re^{N \times n} \Leftrightarrow \mathbf{Z}^* = \left[\mathbf{z}_1^*, \cdots, \mathbf{z}_k^*, \cdots, \mathbf{z}_n^*\right] \; \mathbf{z}_k^* \in \Re^N$$

The minor product moment of the correlation matrix, **R**, is

$$\mathbf{R} = \mathbf{Z}_x^{*\prime}\mathbf{Z}_x^* \qquad (3.2.1\text{-}33)$$

where the subscript **x** indicates a matrix of small standard deviates with respect to the predictors.

The vector of regression coefficients (often called beta weights) becomes

$$\mathbf{b}^* = \mathbf{R}^{-1}\mathbf{Z}_x^{*\prime}\mathbf{z}_y^* \qquad (3.2.1\text{-}34)$$

where $\mathbf{z}_y^*$ is the criterion vector of small standard deviates. The product of $\mathbf{Z}_x^{*\prime}\mathbf{z}_y^*$ forms the vector product-moment correlations between the criterion and predictor elements. The original **b** vector as in equation *3.2-31* is found by the conversion equation below:

$$b_j = \frac{b_j^* \sigma_y}{\sigma_{jx}} \qquad (3.2.1\text{-}35)$$

where the subscript *j* indicates a particular variable, 1 ... *n*. Equation *3.2.1-35* involves simple ratios of the standard deviations of the predictor and criterion variables. Next, solving for the intercept term b_0 from

$$b_0 = \overline{Y} - \sum_{j=1}^{n} b_j \overline{X}_j \qquad (3.2.1\text{-}36)$$

And finally, the linear estimating equation is the same as above in Equation *3.2-32*.

3.3 Factor Analysis

Factor analysis was first started with the work of Charles Spearman in 1904. In factor analysis a variable may be represented by a linear combination of factors, or hypothetical constructs. The basic factor model in classical factor analysis (Harman, 1976) is described by

$$z_j = a_{j1}F_1 + a_{j2}F_2 + \cdots + a_{jm}F_m + u_jY_j \qquad (3.3\text{-}37)$$

where each observed variable, z_j, is described in m common factors, F, and a unique factor Y. The a and u coefficients are called loadings. Typically, the loadings are measures of the degree of generalizability found between each factor and variable.

A system of 5 equations, e.g., would be written as

$$\begin{aligned}
z_1 &= a_{11}F_1 + a_{12}F_2 + \cdots a_{1m}F_m + u_1Y_1 \\
z_2 &= a_{21}F_1 + a_{22}F_2 + \cdots a_{2m}F_m + 0 + u_2Y_2 \\
z_3 &= a_{31}F_1 + a_{32}F_2 + \cdots a_{3m}F_m + 0 + 0 + u_3Y_3 \\
z_4 &= a_{41}F_1 + a_{42}F_2 + \cdots a_{4m}F_m + 0 + 0 + 0 + u_4Y_4 \\
z_5 &= a_{51}F_1 + a_{52}F_2 + \cdots a_{5m}F_m + 0 + 0 + 0 + 0 + u_5Y_5
\end{aligned}$$

Note the unique factor, Y, is particular for that variable only while the common factors are shared between variables.

For N observations, the factor model is

$$z_{ji} = \sum_{p=1}^{m} a_{jp}F_{pi} + u_jY_{ji} (i = 1, 2, \cdots, N; j = 1, 2, \cdots, n) \qquad (3.3\text{-}38)$$

Of most importance is the set of m common factors, since they account for the correlations among the variables. This may be thought of as a collective sort of binding, or communality, between variables. Although the underlying factors are called hypothetical constructs, they may have several different names. The names are given depending upon the value of the loadings. For example, if all the variables have a common factor with nonzero loadings, it is termed a general factor. If a factor has nonzero loadings on two or more variables, but fewer than n, it is called a group factor. In cases where the factor has nonzero loadings on only two variables, it is called a doublet.

There are also unique factors in equation *3.3-38* which do not account for the intercorrelations between the variables but makeup the difference when total correlations are less than unity. The unique factors account for the remaining variance. In some cases, the unique variance may be further divided into two parts: variance unique to that variable, and error attributed to the measurement process.

Since the factors are unknown to begin with, it is assumed the *F's* and *Y's* have zero means and unit variances. In addition, the n unique factors are also supposed to be independent of one another and independent of the m common factors. The primary interest is estimating the nm loadings of the common factors. Later on, it will be shown how the loadings are used to reproduce the correlation matrix – satisfying the primary objective of the classical factor model: to maximally reproduce the correlations between the variables. A side point of interest is the resemblance the factor analysis model bears to that of regression analysis insofar as a variable is described as a linear combination of another set of variables plus a residual. In regression analysis the independent variables are observable measurable quantities, while in factor analysis the independent variables are hypothetical constructs which can only be estimated from the observed data. If necessary, factor scores are estimated in a subsequent stage of the analysis.

3.3.1 Principal Components Model vs. Classical Factor Analysis Model

Principal components is another similar method of extracting factors. It was first proposed by Karl Pearson (1901) and later developed by Harold Hotelling (1933). The basic formulation is

$$z_j = a_{j1}F_1 + a_{j2}F_2 + \cdots + a_{jm}F_n (j = 1, 2, \cdots, n) \qquad (3.3.1\text{-}39)$$

where each of the n observed variables is described by n uncorrelated components, F_1, F_2, ..., F_n. Note the similarity of equation *3.3-37* with equation *3.3.1-39*. Except for the unique factors the two are identical. However, the factors are such that each component in the principal factor model makes a maximum contribution to the sum of the variances of the n variables. Hence, a major distinction between the factor analysis model and the principal component model is that the principal component method extracts the maximum variance, while the factor analysis method attempts to "best" reproduce the observed correlations--which will be seen later on. It is also important that all components in equation *3.3.1-39* are required to reproduce the correlations among the variables. Removing minor components, which contribute very little to the overall variance, significantly alters the correlation matrix.

3.3.2 Factor Mathematics

In the following derivations both column and row vectors will be used for clarity. However, care must be taken to prevent confusion whenever variables or observations are discussed, since both use column notation. In contrast, combining sets of variables and observations forms a matrix, which is best described using row notation. The use of row notation alters the calculation of the correlation matrix, which was described earlier by equation *3.1-22*.

A single set of measurements of n variables can be represented by a column vector (Harman, 1976), **z**, such that

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_n \end{bmatrix} \qquad (3.3.2\text{-}40)$$

where the subscript designates a specific variable. In addition, a number of measurements or N observations of a particular variable, z_j, in vector notation using small standard deviates is

$$\mathbf{z}_j^* = \begin{bmatrix} z_1^* \\ z_2^* \\ z_3^* \\ \vdots \\ z_N^* \end{bmatrix}, \; or \; \mathbf{z}_j^{*t} = \begin{bmatrix} z_1^* & z_2^* & z_3^* & \cdots & z_N^* \end{bmatrix} \qquad (3.3.2\text{-}41)$$

where the notation " ^{t} " or " $'$ " implies transpose, and the subscript now designates a specific point of interest or observation. Hence, for a set of n variables and N observations the notation is

$$\mathbf{Z} = \begin{bmatrix} z_{11}^* & z_{12}^* & \cdots & z_{1N}^* \\ z_{21}^* & z_{22}^* & \cdots & z_{2N}^* \\ \vdots & \cdots & \ddots & \vdots \\ z_{n1}^* & z_{n2}^* & \cdots & z_{nN}^* \end{bmatrix} \qquad (3.3.2\text{-}42)$$

where $\mathbf{Z}$ is the observed $n \times N$ matrix of small standard deviates (nuclear data). Alternatively, the row partitioning of the matrix $\mathbf{Z}$ is a collection of row vectors:

$$\mathbf{Z} \in \Re^{n \times N} \Leftrightarrow \mathbf{Z} = \begin{bmatrix} \mathbf{z}_1^{*t} \\ \vdots \\ \mathbf{z}_k^{*t} \\ \vdots \\ \mathbf{z}_n^{*t} \end{bmatrix}; \ \mathbf{z}_k^{*t} \in \Re^{1 \times N}$$

where the dependent variables are now the row vectors $\mathbf{z}_1^{*t}, \ldots, \mathbf{z}_n^{*t}$. It should be noted the matrix $\mathbf{Z}$ may include an assemblage of variables, both predictors and criterions.

Similarly, common factors are represented by the vector $\mathbf{f}$ and unique factors by the vector $\mathbf{y}$:

$$\mathbf{f} = \begin{bmatrix} F_1 \\ \vdots \\ F_m \end{bmatrix} \tag{3.3.2-43}$$

$$\mathbf{y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} \tag{3.3.2-44}$$

For a set of N observations the common and unique factor matrices, respectively, are

$$\mathbf{F} = \begin{bmatrix} F_{11} & \cdots & F_{1N} \\ \vdots & \ddots & \vdots \\ F_{m1} & \cdots & F_{mN} \end{bmatrix} \tag{3.3.2-45}$$

$$\mathbf{Y} = \begin{bmatrix} Y_{11} & \cdots & Y_{1N} \\ \vdots & \ddots & \vdots \\ Y_{n1} & \cdots & Y_{nN} \end{bmatrix} \tag{3.3.2-46}$$

Whereas the loadings are represented by two separate matrices, **A** and **U**, it is sometimes simpler to combine them into one matrix, **M**. Hence,

$$(\mathbf{A} \mid \mathbf{U}) = \mathbf{M} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} & u_1 & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & a_{2m} & 0 & u_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} & 0 & 0 & \cdots & u_n \end{bmatrix} \qquad (3.3.2\text{-}47)$$

A is commonly referred to as the pattern matrix, while **M** is called the total pattern matrix. Thus a set of variable equations forms a factor pattern:

$$\begin{aligned} z_1 &= a_{11}F_1 + a_{12}F_2 + \cdots + a_{1m}F_m + u_1Y_1 \\ z_2 &= a_{21}F_1 + a_{22}F_2 + \cdots + a_{2m}F_m + 0 + u_2Y_2 \\ &\vdots \\ z_n &= a_{n1}F_1 + a_{n2}F_2 + \cdots + a_{nm}F_m + 0 + 0 + \cdots + u_nY_n \end{aligned}$$

which in vector/matrix form is simply

$$\mathbf{z} = \mathbf{Af} + \mathbf{Uy} \qquad (3.3.2\text{-}48)$$

or for a collection of N observations, the matrix notation is defined as

$$\mathbf{Z} = \mathbf{AF} + \mathbf{UY} \qquad (3.3.2\text{-}49)$$

In addition to a factor pattern, another important part is the factor structure. A table or matrix of a variable's correlations with a set of factors is called a factor structure. Normally, both a factor structure and pattern are necessary in order to furnish a complete explanation of the problem. However, in some cases the factor structure becomes identical to the factor pattern. This is when the variables are in standardized format and the intercorrelations among the common factors are zero.

Letting a particular variable's correlations with a common factor designated by the symbol s_{jp}, where

$$s_{jp} = r_{z_j F_p} (j = 1, 2, ..., n; p = 1, 2, ..., m)$$

and the subscript j indicates the variable number, subscript p indicates the factor, and $r_{z_j F_p}$ is the correlation between the two, the factor structure may now be represented in matrix form as

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1m} \\ s_{21} & s_{22} & \cdots & s_{21} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nm} \end{bmatrix} \qquad (3.3.2\text{-}50)$$

3.3.2.1 Relationships between Factor Pattern and Structure

From equation *3.3.2-49*, the variables are related to the loadings and common factors by

$$\mathbf{Z} = \mathbf{AF} \qquad (3.3.2.1\text{-}51)$$

Post-multiplying each side by the transpose of **F** yields

$$\mathbf{ZF'} = \mathbf{AFF'} \qquad (3.3.2.1\text{-}52)$$

Now, the factor structure is defined to be equivalent to the left-hand member of equation *3.3.2.1-52*, or

$$\mathbf{S} = \mathbf{ZF'} \qquad (3.3.2.1\text{-}53)$$

Hence,

$$\mathbf{S} = \mathbf{ZF}' = \mathbf{A}(\mathbf{FF}') \qquad (3.3.2.1\text{-}54)$$

The symmetric matrix formed by **FF'** is also the correlation matrix of common factors, defined as

$$\Theta = \mathbf{FF}' = \begin{bmatrix} 1 & r_{F_1F_2} & \cdots & r_{F_1F_m} \\ r_{F_2F_1} & 1 & \cdots & r_{F_2F_m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{F_mF_1} & r_{F_mF_2} & \cdots & 1 \end{bmatrix} \qquad (3.3.2.1\text{-}55)$$

For uncorrelated factors

$$\Theta = \mathrm{I} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \qquad (3.3.2.1\text{-}56)$$

or **FF'** is the equal to the identity matrix. Thus the factor structure and pattern are equivalent, i.e.,

$$\mathbf{S} = \mathbf{AI} = \mathbf{A} \qquad (3.3.2.1\text{-}57)$$

3.3.2.2 Relationships between Observed and Reproduced Correlation Matrices

The matrix of observed correlations is given by

$$\mathbf{R} = \mathbf{ZZ}' \qquad (3.3.2.2\text{-}58)$$

where **Z** is defined earlier as a collection of row vectors of small standard deviates. Expanding equation *3.3.2.2-58* yields:

$$\begin{aligned}
\mathbf{R} &= \mathbf{ZZ'} = (\mathbf{AF}+\mathbf{UY})(\mathbf{AF}+\mathbf{UY})' \\
&= (\mathbf{AF}+\mathbf{UY})((\mathbf{AF})'+(\mathbf{UY})') \\
&= \mathbf{AFF'A'}+\mathbf{AFY'U}+\mathbf{UYF'A'}+\mathbf{UYY'U'} \\
&= \mathbf{A\Theta A'}+\mathbf{U}^2 \text{ or} \\
\mathbf{R} &= \mathbf{ZZ'} = \left(\mathbf{M}\begin{bmatrix}\mathbf{F}\\ \mathbf{Y}\end{bmatrix}\right)\left(\mathbf{M}\begin{bmatrix}\mathbf{F}\\ \mathbf{Y}\end{bmatrix}\right)^{t} = \mathbf{M}\begin{bmatrix}\mathbf{F}\\ \mathbf{Y}\end{bmatrix}\begin{bmatrix}\mathbf{F'} & \mathbf{Y'}\end{bmatrix}\mathbf{M'} \qquad (3.3.2.2\text{-}59) \\
&= \mathbf{M}\begin{pmatrix}\mathbf{FF'} & \mathbf{FY'}\\ \mathbf{YF'} & \mathbf{YY'}\end{pmatrix}\mathbf{M'} \\
&= \mathbf{M}\begin{pmatrix}\Theta & 0\\ 0 & \mathbf{I}\end{pmatrix}\mathbf{M'} = \begin{bmatrix}\mathbf{A} & \mathbf{U}\end{bmatrix}\begin{bmatrix}\Theta & 0\\ 0 & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{A'}\\ \mathbf{U}\end{bmatrix} = \begin{bmatrix}\mathbf{A\Theta} & \mathbf{UI}\end{bmatrix}\begin{bmatrix}\mathbf{A'}\\ \mathbf{U}\end{bmatrix} \\
&= \mathbf{A\Theta A'}+\mathbf{U}^2
\end{aligned}$$

The observed correlation matrix consists of two parts. The first part is a function of the common factor loadings and the second is a function of the unique loadings. Off-diagonal terms of the correlation matrix can only be reproduced by the common loadings since the unique loadings define a square matrix in which all off-diagonal entries are zero. As a result, the division of the correlation matrix into two additional matrices yields

$$\mathbf{R} = \begin{bmatrix} 1-u_1^2 & r_{12} & \cdots & r_{1n} \\ r_{21} & 1-u_2^2 & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & 1-u_n^2 \end{bmatrix} + \begin{bmatrix} u_1^2 & 0 & \cdots & 0 \\ 0 & u_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_n^2 \end{bmatrix} \qquad (3.3.2.2\text{-}60)$$

where the first matrix is equivalent to $\mathbf{A\Theta A'}$. The terms $1\text{-}u_1^2$, $1\text{-}u_2^2$, etc., are called communalites of variables z_j, and are designated h_1^2, h_2^2, etc., respectively. A while later, in the section on Variance Analysis, communality is determined to be the part of the total variance attributable to the common factors.

Replacing communalities in the main diagonal of the correlation matrix produces

$$\mathbf{R} = \begin{bmatrix} h_1^2 & r_{12} & \cdots & r_{1n} \\ r_{21} & h_2^2 & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & h_n^2 \end{bmatrix} + \begin{bmatrix} u_1^2 & 0 & \cdots & 0 \\ 0 & u_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_n^2 \end{bmatrix} \qquad (3.3.2.2\text{-}61)$$

By letting $\hat{\mathbf{R}}$ represent the matrix of reproduced correlations (with communalites in the main diagonal),

$$\mathbf{R} = \hat{\mathbf{R}} + \mathbf{U}^2 \qquad (3.3.2.2\text{-}62)$$

or when the factors are uncorrelated (as shown in equation *3.3.2.1-56*),

$$\begin{aligned} \hat{\mathbf{R}} &= \mathbf{A}\Theta\mathbf{A}' \\ &= \mathbf{A}\mathbf{A}' \end{aligned} \qquad (3.3.2.2\text{-}63)$$

Therefore the observed correlation matrix is just

$$\mathbf{R} = \mathbf{A}\mathbf{A}' + \mathbf{U}^2 \qquad (3.3.2.2\text{-}64)$$

Equation *3.3.2.2-63* is commonly referred to as the fundamental theorem of factor analysis. The linkage between the measured data and the common factor loadings is given by equation *3.3.2.2-64*, via. observed and reproduced correlation matrices. It is important to note that the observed correlation matrix, equation *3.3.2.2-64*, may be obtained by using only common-factor loadings, or a combination of either common

and unique loadings. In the case where the observed correlation is fitted by both, the minimization of the residuals[ii]

$$\left\|\mathbf{R}-(\hat{\mathbf{R}}+\mathbf{U}^2)\right\|_F^2$$

reduces to the principal components method.

When communalites are used in the main diagonal of **R**, minimization of

$$\left\|(\mathbf{R}-\mathbf{U}^2)-\mathbf{A}\mathbf{A}'\right\|_F^2$$

leads to principal factors.

If only the off-diagonal elements (*(n(n-1)/2)* off-diagonal residual correlations) are considered, the minimization[iii] of

$$\left\|(\mathbf{R}-\mathbf{I})-(\hat{\mathbf{R}}-(\mathbf{I}-\mathbf{U}^2))\right\|_F^2$$

leads to the method called "minres" or "minimum residuals". Here the diagonal elements of the correlation matrix are not known, i.e., are to be determined. Hence, it should be clear that the values put in the diagonal of the observed correlation matrix determine:

1. the type of factors that are produced, and

[ii] The Frobenius norm measure may be considered an overestimation since the residual matrix is symmetric.

[iii] Since increasing the number of variables affects the value of the minimization function, normally the root-mean-square deviation is used, hence, the minimization function is independent of the order of the correlation matrix:

$$\text{root - mean - square} = \sqrt{\frac{2\left\|\text{minimization function}\right\|}{n(n-1)}}$$

2. the portions of the unit variances that the common factors constitute.

Assuming that only off-diagonal terms of the observed correlation matrix are minimized, then both communality and uniqueness are ascertainable. This is important and necessary when decisions regarding selection of variables are made.

3.3.3 Number of Common Factors

The total number of common factors should be less than the number of variables. It is the smallest number of factors that reconstructs the correlations that will determine the size of the factor space. The following explanation provides a logical basis for this statement.

A mathematical theorem (Harman 1976, p. 64) states *the rank of the product of a matrix by its transpose is equal to the rank of the matrix.* The correlation matrix is the product of the small standard deviate matrix, **Z**, by its transpose, **Z'** (equation *3.3.2.2-58*). Since the number of observations, N, in **Z** is usually much larger than the number of variables, the maximum rank of the data could reside in an N-dimensional space. However, the observed correlation matrix with ones in the main diagonal usually has rank n, the number of variables. This smaller n-dimensional space is imbedded in the larger N-space. Hence, all the variables would then be describable in not less than n possible factors.

Now, if the factor pattern, **A**, is used to reproduce the correlation matrix, $\hat{\mathbf{R}}$ (with communality's in the main diagonal), then the total number of linearly independent factors is m. This is called the common-factor space of m-dimensions or flat m-space. Figure 4 Collapsing of N-space to m-space shows the reduction to smaller spaces, all which happens from the reformation of

$$\mathbf{Z} \Rightarrow \mathbf{R} \Rightarrow \hat{\mathbf{R}}$$

At this point the dimension of the common-factor space is equivalent to the rank of the reproduced correlation matrix. The problem of communality decides the rank of the reproduced correlation matrix, which determines the smallest number of common factors necessary to account for the intercorrelations.

Two approaches are possible. One, is to approximate communality estimates, which will determine the number of common factors. This is the principal factor method. The other approach is to fix the dimension of the common-factor space, eventually computing communalities. The later follows the classical factor model.

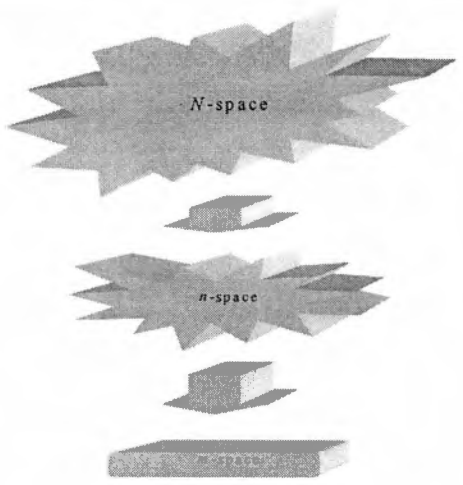

Figure 4 Collapsing of N-space to m-space

Another issue is that of indeterminacy of the factor solution. Since the factor loadings reside in *m*-space, a question arises as to the uniqueness of the factor pattern coefficients. Many different types of orthogonal and/or oblique rotations or translations of factors in the *m*-space are possible, while still achieving a factor pattern that has consistency with the observed correlations. Normally a unique factor space is provided by the factor model, but not the common factor loadings. Since the factor analysis solution does not provide a connected basis or frame of reference to either the *N*-space or *n*-space, a physical interpretation is impossible. However, any solution can be put into a canonical form, which is merely designating a coordinate system in *m*-space that satisfies certain properties. One such canonical form that can be used is so that successive factors account for the maximum possible variance in the *m*-space provided by the original solution. Note: this is not the same as extracting the maximum variance from the variables! The implications of using a canonical form allow equivalence of two solutions, if both solutions are truly equivalent, then in canonical form both solutions would be identical.

3.3.4 Variance Analysis

The relationships between variables and factor loadings are best realized when performing a variance analysis on the factor model (equation *3.3-37*).

The standard variance of a variable, z_j, is expressed by

$$\sigma_{z_j}^2 = \frac{\sum_{i=1}^{N} z_{ji}^2}{N} = \sum_{i=1}^{N} \left(z_{ji}^*\right)^2 = 1 \text{ or}$$
$$\sigma_{\mathbf{j}}^2 = \frac{\mathbf{z}_{\mathbf{j}}^{\mathbf{t}}\mathbf{z}_{\mathbf{j}}}{N} = \mathbf{z}_{\mathbf{j}}^{*\mathbf{t}}\mathbf{z}_{\mathbf{j}}^{*} = 1 \qquad (3.3.4\text{-}65)$$

Inserting the mathematical definition of the factor model (equation *3.3-38*) into equation *3.3.4-65* simplifies to

$$\sigma^2 = \begin{bmatrix} \mathbf{z}_1^{*t}\mathbf{z}_1^{*} \\ \mathbf{z}_2^{*t}\mathbf{z}_2^{*} \\ \mathbf{z}_3^{*t}\mathbf{z}_3^{*} \\ \\ \mathbf{z}_n^{*t}\mathbf{z}_n^{*} \end{bmatrix} = \mathbf{1} \qquad (3.3.4\text{-}66)$$

The vector **1** is a $n \times 1$-unit column vector. Expanding equation *3.3.4-65* using common and unique loadings and factors yields

$$\begin{aligned}
\sigma_{\mathbf{j}}^2 &= \mathbf{z}_{\mathbf{j}}^{*t}\mathbf{z}_{\mathbf{j}}^{*} \\
&= \left(\mathbf{A}(j,:)\mathbf{F} + \mathbf{U}(j,:)\mathbf{Y}\right)\left(\mathbf{A}(j,:)\mathbf{F} + \mathbf{U}(j,:)\mathbf{Y}\right)^{t} \\
&= \left(\mathbf{A}(j,:)\mathbf{F} + \mathbf{U}(j,:)\mathbf{Y}\right)\left((\mathbf{A}(j,:)\mathbf{F})' + (\mathbf{U}(j,:)\mathbf{Y})'\right) \\
&= \left(\mathbf{A}(j,:)\mathbf{F} + \mathbf{U}(j,:)\mathbf{Y}\right)\left(\mathbf{F}'\mathbf{A}(j,:)' + \mathbf{Y}'\mathbf{U}(j,:)'\right) \\
&= \mathbf{A}(j,:)\mathbf{F}\mathbf{F}'\mathbf{A}(j,:)' + \mathbf{A}(j,:)\mathbf{F}\mathbf{Y}'\mathbf{U}(j,:)' + \mathbf{U}(j,:)\mathbf{Y}\mathbf{F}'\mathbf{A}(j,:)' + \mathbf{U}(j,:)\mathbf{Y}\mathbf{Y}'\mathbf{U}(j,:)'
\end{aligned}$$

This can be broken down into parts where

$$\mathbf{A}(j,:)\mathbf{F}\mathbf{F}'\mathbf{A}(j,:)' = \sum_{p=1}^{m} a_{jp}^2 \left(\sum_{i=1}^{N} F_{pi} / N \right) + 2 \sum_{p<q=1}^{m} a_{jp} a_{jq} \left(\sum F_{pi} F_{qi} / N \right)$$

$$\mathbf{A}(j,:)\mathbf{F}\mathbf{Y}'\mathbf{U}(j,:)' + \mathbf{U}(j,:)\mathbf{Y}\mathbf{F}'\mathbf{A}(j,:)' = 2u_j \sum_{p=1}^{m} a_{jp} \left(\sum F_{pi} Y_{ji} / N \right)$$

$$\mathbf{U}(j,:)\mathbf{Y}\mathbf{Y}'\mathbf{U}(j,:)' = u_j^2 \sum Y_{ji}^2 / N$$

or simply

$$\sigma_j^2 = \sum_{p=1}^{m} a_{jp}^2 + u_j^2 + 2 \sum_{p<q=1}^{m} a_{jp} a_{jq} r_{F_p F_q} + 2u_j \sum_{p=1}^{m} a_{jp} r_{F_p Y_j} \qquad (3.3.4\text{-}67)$$

By imposing the restrictions that unique factors are always uncorrelated with the common factors, and if the common factors are uncorrelated, then equation *3.3.4-67* reduces to

$$\begin{aligned} \sigma_j^2 &= a_{j1}^2 + a_{j2}^2 + \cdots + a_{jm}^2 + u_j^2 \\ &= \mathbf{A}(j,:)\mathbf{A}(j,:)' + \mathbf{U}(j,:)\mathbf{U}(j,:)' \\ &= \left\| \mathbf{A}(j,:) \right\|_2^2 + \left\| \mathbf{U}(j,:) \right\|_2^2 \end{aligned} \qquad (3.3.4\text{-}68)$$

Equation *3.3.4-68* expresses variance as function of common and unique factor loadings. Therefore, the total standard variance in vector form is just a combination of 2-norms of the common and unique loadings, i.e.,

$$\boldsymbol{\sigma}^2 = \begin{bmatrix} \left\| \mathbf{A}(1,:) \right\|_2^2 + \left\| \mathbf{U}(1,:) \right\|_2^2 \\ \left\| \mathbf{A}(2,:) \right\|_2^2 + \left\| \mathbf{U}(2,:) \right\|_2^2 \\ \left\| \mathbf{A}(3,:) \right\|_2^2 + \left\| \mathbf{U}(3,:) \right\|_2^2 \\ \vdots \\ \left\| \mathbf{A}(n,:) \right\|_2^2 + \left\| \mathbf{U}(n,:) \right\|_2^2 \end{bmatrix} = \mathbf{1} \qquad (3.3.4\text{-}69)$$

This important result provides two important concepts in factor analysis:

1. Communality of a variable z_j is given by the sum of the squares of the common-factor coefficients, via., $h_j^2 = \sum a_{jp}^2 = a_{j1}^2 + a_{j2}^2 + \cdots + a_{jm}^2 (j = 1,2,\cdots,n) = 1 - u_j^2$.
2. Uniqueness – which is the variance contribution of a unique factor model specific to a variable.

Although uniqueness is that portion which the common factors fail to account for, it may be specific to the unique factor only or contributable to measurement error. Hence, it is normally desirable to divide uniqueness into two separate portions of variance – that due to the particular selection of variables in the study (specificity) and that due to unreliability of measurement (unreliability). Specificity is the portion of variance contributable to a particular variable. Unreliability or error variance is random and is assumed to belong to neither the common factor nor specific factor domains. It is usually the remaining portion due to imperfections of measurement.

When the composition of variance of a variable is to be found, the following procedure is used:

First, in order to determine the values of specificity and unreliability it is normal to first calculate communality estimates from the common-factor loadings.

- Estimate communality: h_j^2

  Since the total variance is defined by $\sigma_j^2 = 1 = h_j^2 + u_j^2$, uniqueness can be determined next.

  - Determine uniqueness: $u_j^2 = \sigma_j^2 - h_j^2 = 1 - h_j^2$
  - Determine a reliability coefficient.

A measure of a variable's variance that is nonrandom is given by a reliability coefficient, r_{zz}, which is an estimate of the true variance. The complement of reliability is error, or the random part. For example, if there are two independent measurements of the same variable, denoted as z_{jA} for the first representation and z_{jB} as the second representation, then their correlation, r_{jAjB}, is the measure of reliability of the variable.

However, this last statement needs quantifying because several different types of reliability coefficients may be used. The reliability coefficients designation are based upon the type of error.

First, one may have a stability coefficient of reliably when error is defined as nongeneralizability over the period of time between the initial measurement and a second measurement using the same procedure (Gorsuch, 1983). This is when one would measure and then re-measure the same system at a later time. This normally is not the case in sensor measurements, since the sensor is usually measuring a different process some time later, i.e., the process is assumed to be changing as a function of time. One might argue in quasi-static systems that some internal processes are not changing, however, it is virtually impossible to assure that a point in time and at a particular position of the process is the same at a later time. Any small perturbation would essentially negate such an assumption. And for dynamic processes, the sensor would essentially have to be moved from one position to another, which would invalidate the assumption of using the same procedure. In the second approach to reliability, error is defined as nongeneralizability across various ways of measuring the process within the same limited timeframe. Essentially two different types of sensors would measure the same process point. Here there is a homogeneity, or internal consistency, of the reliability coefficient. One could also consider a single sensor measuring a static or dynamic process over a constant period of time. For the static case, deviations from a designated reliable value would indicate error. If the process were dynamic, it would be difficult to determine a reliability coefficient with a single measurement unless some prior knowledge was known about the process itself. Lastly, there are two forms of parallel reliability, where error takes on a different form, depending upon the case. One is if parallel measurements are collected

within a very limited time span. Several measurements taken with different sensors at the same localized position within a period of time would qualify as such. This parallel form of reliability is similar to the internal consistency reliability approach. Secondly, if the parallel measurements are separated in time, then the error is defined as both nongeneralizability over time and nongeneralizability from one way of measuring the variable to another similar way of measuring it. This is normally not the case when analyzing sensor values.

Since the variables selected for factor analysis usually comprise a definite period of time, reliability coefficients are based on internal consistency or parallel form. The exact expression to use is a function of the kinds of errors that normally affect the intercorrelations between the variables.

Under the assumption that some form of reliability is known or estimated for a variable, the error variance may be obtained by means of the equation

- Error variance: $e_j^2 = 1 - r_{zz}$

Specificity, s_j^2, can now be calculated.

- Determine specificity: $s_j^2 = \sigma_j^2 - h_j^2 - e_j^2 = u_j^2 - e_j^2$

Now, by including specificity and unreliability in the classical factor model (equation *3.3-37*), the formula is

$$z_j = a_{j1}F_1 + a_{j2}F_2 + \cdots + a_{jm}F_m + s_j S_j + e_j E_j (j = 1,2,\cdots,n) \qquad (3.3.4\text{-}70)$$

where the "*e*" and "*s*" and S_j and E_j, are specificity and error loadings and factors, respectively.

Not only is a variable's variance defined by the above formulas, but additional equations may be derived to find the contribution of the common factors to the variance. For example, the value of a_{j3}^2 is the

contribution of the common factor, F_3, to the variance of variable z_j. The total contribution of a common factor to the variance of all variables is

$$\sigma_p^2 = \sum_{j=1}^{n} a_{jp}^2 \qquad (3.3.4\text{-}71)$$

Furthermore, the contribution of all common factors to the variance of all variables is

$$\Omega = \sum_{p=1}^{m} \sigma_p^2 = \sum_{p=1}^{m} \sum_{j=1}^{n} a_{jp}^2 \qquad (3.3.4\text{-}72)$$

Other formulas and or descriptions, such as completeness of factorization, complexity, etc. yield valuable information about the factorization of the data.

3.3.5 Summary

Finally, a complete analysis of the communality, specificity and unreliability, along with other measurement criteria, provide insight into the question of how a set of factors explain the relationships among the variables. Since the primary objective of factor analysis is to fit a set of data with a reproduced correlation matrix, if near equivalence is achieved from the observed correlations and those derived from the model, the model is good representation of the data. Although some allowance must be made for sampling and measurement errors.

3.4 Principal Components Regression

Principal components regression is a two-step procedure. In the first step the data is transformed into principal components space via a weight matrix. The new representation of the data are called scores. Following conversion of the data into scores, multiple regression is applied to the score matrix against one or several dependent variables.

Understanding PCR involves linear transformations, principal component analysis, and multiple regression. The principal components space also provide some interesting aspects, since it forms an orthonormal basis.

3.4.1 Linear Transformations

A linear composite or linear transformation of several variables (quantity p) essentially takes a vector, $\mathbf{x}$, where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \tag{3.4.1-73}$$

and transforms it into a single scalar, y, via the mapping $y = b_1x_1 + b_2x_2 + \ldots + b_px_p$. This is essentially a point transformation of the column vector $\mathbf{x}$ by a set of direction cosines or weights onto y in some space of interest. A set of linear equations is obtained by mapping this vector into a k dimensional vector, i.e., $\mathbf{y}' = [y_1, y_2, \ldots, y_k]$. Hence,

$$\mathbf{y} = \mathbf{B}\mathbf{x} \tag{3.4.1-74}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}, \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & & b_{2p} \\ \vdots & & \ddots & \vdots \\ b_{k1} & b_{k2} & \cdots & b_{kp} \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$$

Here, the column partitioning of the matrix $\mathbf{B}$ is a collection of column vectors:

$$\mathbf{B} \in \Re^{k \times p} \Leftrightarrow \mathbf{B} = \begin{bmatrix} \mathbf{b}_1 & \cdots & \mathbf{b}_j & \cdots & \mathbf{b}_k \end{bmatrix}, \ \mathbf{b}_j \in \Re^k$$

For a data matrix the scores are:

$$\mathbf{Y}_{\text{scores}} = \mathbf{XB}$$

3.4.2 Principal Component Analysis (PCA)

Instead of a single observation, we now assume we have a set of data containing p variables and N observations. The data is given in matrix and column vector notation by

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & & x_{2p} \\ \vdots & & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Np} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_j & \cdots & \mathbf{x}_N \end{bmatrix}, \ \mathbf{x}_j \in \Re^N$$

Again a mapping of the data is considered as in equation *3.4.1-74*, however, this time a set of direction cosines or weights defines the vector of linear composites, i.e., multiple regression:

$$\mathbf{y} \cong \mathbf{Xb} \qquad (3.4.2\text{-}75)$$

where **y** and **b** are column vectors of dimension $N\times1$ and $p\times1$, respectively. Multiplying each side by the transpose of **y** yields a point transformation

$$\begin{aligned} \mathbf{y}'\mathbf{y} &= (\mathbf{Xb})'\mathbf{Xb} \\ &= \mathbf{b}'\mathbf{X}'\mathbf{Xb} \\ &= \mathbf{b}'\mathbf{Cb} \end{aligned} \qquad (3.4.2\text{-}76)$$

where the matrix **C** is the minor product moment of **X**. If small standard deviates are used in place of raw data points, then from equation **Error! Cannot open file.** the minor product moment of $\mathbf{Z}^*$ ($\mathbf{Z}^{*\prime}\mathbf{Z}^*$) simplifies equation *3.4.2-76* such that

$$\mathbf{y}'\mathbf{y} = \mathbf{b}'\mathbf{R}\mathbf{b} \qquad (3.4.2\text{-}77)$$

where **R** is the observed correlation matrix. **R** is also called the matrix of the quadratic form that finds the squared length of **b** under the linear transformation of **X**, i.e., via equation *3.4.2-75*.

The principal components problem is to maximize equation *3.4.2-77* subject to the constraint the vector norm of **b'b** is equal to one. This limits the maximum value of the weights; without it the weights could essentially approach infinity. Using LaGrange multipliers equation *3.4.2-77* can be re-written as

$$\mathbf{y}'\mathbf{y} = \mathbf{b}'\mathbf{R}\mathbf{b} - \lambda(\mathbf{b}'\mathbf{b} - 1) \qquad (3.4.2\text{-}78)$$

where **b'b** –1 is the constraint equation. Using the symbolic derivative as in defined in *3.2-27*, one obtains

$$\begin{aligned} \frac{\partial(\mathbf{y}'\mathbf{y})}{\partial\mathbf{b}} &= \frac{\partial}{\partial\mathbf{b}}\left[\mathbf{b}'\mathbf{R}\mathbf{b} - \lambda(\mathbf{b}'\mathbf{b} - 1)\right] = \mathbf{0} \\ &= 2\mathbf{R}\mathbf{b} - 2\lambda\mathbf{b} = \mathbf{0} \\ &\Rightarrow (\mathbf{R} - \lambda\mathbf{I})\mathbf{b} = \mathbf{0} \end{aligned} \qquad (3.4.2\text{-}79)$$

Thus the solution to the above reduces to an eigenvalue/eigenvector problem.

Explained in other terms, principal components analysis finds the first weight vector, $\mathbf{b}_1$, where

$$\mathbf{b}_1 = \begin{bmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{p1} \end{bmatrix}$$

such that the variance of $\mathbf{Xb}_1 = \mathbf{b}_1^t\mathbf{Cb}_1$ ($\mathbf{C}$ is the covariance or correlation matrix) is maximum given that vector norm squared of $\mathbf{b}_1^t\mathbf{b}_1 = 1$. Then it finds the second weight vector, $\mathbf{b}_2$, where

$$\mathbf{b}_2 = \begin{bmatrix} b_{12} \\ b_{22} \\ \vdots \\ b_{p2} \end{bmatrix}$$

such that the variance of $\mathbf{Xb}_2 = \mathbf{b}_2^t\mathbf{Cb}_2$ is maximum given that $\mathbf{b}_2^t\mathbf{b}_2 = 1$ and $\mathbf{b}_2^t\mathbf{Cb}_1 = 0$. The later condition implies the two weight vectors are uncorrelated, i.e., orthogonality. Next, the third weight vector, $\mathbf{b}_3$, is found such that the variance of $\mathbf{Xb}_3^t = \mathbf{b}_3^t\mathbf{Cb}_3$ is maximum given that $\mathbf{b}_3^t\mathbf{b}_3 = 1$, $\mathbf{b}_3^t\mathbf{Cb}_1 = 0$, and $\mathbf{b}_3^t\mathbf{Cb}_2 = 0$. In each successive computed principal component the variance has less variance associated with it then the previous principal component. For arbitrary units of the variables it is best to use the correlation matrix for $\mathbf{C}$, otherwise the principal components weights become more a function of the units of measurement than of the underlying structure of observations.

Since the sum of the variances of the principal components is equal to the sum of the variances of the original variables, one can determine if most of the information lying in the original variables is captured by the first few principal components. If the first principal component has the largest variance relative to the sum of all principal components, then it captures most of the information, and so forth for the next principal components.

Combining the weight vectors into one matrix yields

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pp} \end{bmatrix} \qquad (3.4.2\text{-}80)$$

where

$$\mathbf{B} \in \Re^{p \times p} \Leftrightarrow \mathbf{B} = \begin{bmatrix} \mathbf{b}_1 & \cdots & \mathbf{b}_j & \cdots & \mathbf{b}_p \end{bmatrix}, \mathbf{b}_j \in \Re^p$$

The weight matrix, **B**, is equivalent to all the eigenvectors found in equation *3.4.2-79*. Principal component scores, $\mathbf{P}_{\text{scores}}$, can simply be calculated by

$$\mathbf{P}_{\text{scores}} = \mathbf{Z}^*\mathbf{B} \qquad (3.4.2\text{-}81)$$

3.4.3 Relating Principal Component Scores to Factor Scores

The scores calculated in equation *3.4.2-81* are the same as those derived if performing a factor analysis and minimizing residuals of the observed correlation matrix with ones in the main diagonal (see Relationships between Observed and Reproduced Correlation Matrices). To associate the two different techniques, the factor and PCA scores will be related to one another. Keep in mind that matrices from either method are arranged differently, i.e., variables and observations may use different row or column ordering.

Repeated here, from equation *3.4.2-81* the PCA scores are given by

$$\mathbf{P}_{\text{scores}} = \mathbf{Z}^*\mathbf{B}$$

Post-multiplying each side by $\mathbf{B}^t$ yields

$$\begin{aligned}\mathbf{P}_{\text{scores}}\mathbf{B}^t &= \mathbf{Z}^*\mathbf{B}\mathbf{B}^t \\ &= \mathbf{Z}^*\mathbf{I} \\ &= \mathbf{Z}^*\end{aligned} \qquad (3.4.3\text{-}82)$$

where

$$\mathbf{P}_{\text{scores}} \in \Re^{N\times p}$$
$$\mathbf{B} \in \Re^{p\times p}$$
$$\mathbf{Z}^* \in \Re^{N\times p}$$

Repeated here, from equation *3.3.2.1-51*,

$$\mathbf{Z} = \mathbf{A}\mathbf{F} \qquad (3.4.3\text{-}83)$$

where

$$\mathbf{Z} \in \Re^{n\times N}$$
$$\mathbf{A} \in \Re^{n\times m}$$
$$\mathbf{F} \in \Re^{m\times N}$$

The number of observations, N, is the same from either equation *3.4.3-82* or *3.4.3-83*. However, the number of variables are p and n, respectively; both these should be equal. In addition, when searching for principal components the number of factor scores, m, should be equal to the number of variables, n. Therefore,

$$\mathbf{Z} \in \Re^{p \times N}$$
$$\mathbf{A} \in \Re^{p \times p}$$
$$\mathbf{F} \in \Re^{p \times N}$$

The small standard deviate matrices should be equivalent, $\mathbf{Z}^{*t} = \mathbf{Z}$. Upon taking the transpose of each side of equation *3.4.3-82* results in

$$\begin{aligned} (\mathbf{P}_{scores}\mathbf{B}^{t})^{t} &= \mathbf{Z}^{*t} \\ \mathbf{B}\mathbf{P}_{scores}^{t} &= \mathbf{Z}^{*t} \end{aligned} \qquad (3.4.3\text{-}84)$$

Now, the matrices in equation *3.4.3-84* are related to equation *3.4.3-83* by

$$\begin{aligned} &\mathbf{B}\mathbf{P}_{scores}^{t} = \mathbf{Z}^{*t} \Leftrightarrow \mathbf{Z} = \mathbf{A}\mathbf{F} \\ &\Rightarrow \mathbf{Z}^{*t} = \mathbf{Z} \\ &\Rightarrow \mathbf{B} = \mathbf{A} \\ &\Rightarrow \mathbf{P}_{scores}^{t} = \mathbf{F} \end{aligned}$$

These latest results serve as a check for self-consistency when comparing the eigenfunction or factor analysis techniques of finding principal components.

3.4.4 Principal Component Regression

Once the principal component scores are calculated, the next step is predicting a univariate variable with them. Using the multiple regression equations set forth in Multiple Regression, a new set of predictor equations may be stated as

$$\mathbf{y} = \mathbf{P}_{scores}\mathbf{b} + \mathbf{e} \qquad (3.4.4\text{-}85)$$

The scores matrix, $\mathbf{P}_{\text{scores}}$, may be a reduced matrix such that

$$\mathbf{P}_{\text{scores}} \in \Re^{N \times m}, \text{where } m \leq p$$

Using formula *3.2-31*, the solution to equation *3.4.4-85* may be stated as

$$\hat{\mathbf{b}} = (\mathbf{P}_{\text{scores}}^{\mathbf{t}} \mathbf{P}_{\text{scores}})^{-1} \mathbf{P}_{\text{scores}}^{\mathbf{t}} \mathbf{y} \qquad (3.4.4\text{-}86)$$

It has been noted in several papers that the inversion of $\mathbf{P}_{\text{scores}}^{\mathbf{t}} \mathbf{P}_{\text{scores}}$ does not present a problem because of the mutual orthogonality of the scores.

3.4.5 PCA Dimensionality Issues

Standardized space allows for a geometric interpretation of the variables. However, there are other dimensions that can be used for analysis. These spaces are observation, variable, and *pca*. For p-linearly independent variables in N-dimensional observation space, there is an p-dimensional subspace of the observation space that will describe all the variables. Since the correlation matrix in standardized space is the cosine of the angle between the variable vectors, analyzing the correlation matrix is a direct analysis of this smaller variable space. Unfortunately, this variable space contains p-correlated dimensions. Using p-uncorrelated dimensions removes the necessity to account for the relationships among the variables. An advantage of using principal components analysis of standardized variables is that it generates uncorrelated *pca* dimensions. In addition, analysis of uncorrelated composites versus correlated variables is that the composites will be of unequal size and importance, via standardization. Thorndike (1978) concludes that, "it is not possible to say that any one of them is more important or more pervasive in its influence than another". In *pca* space, the composites vary in the amount of variance, which allows for some interpretation of the components, especially the minor component.

3.5 Partial Least Squares

Whereas principal components regression is a two-step method, partial least squares regression is a step-wise procedure that utilizes information between the independent and dependent variables. In the first step a score vector is found using a weight vector that maximizes the covariance between the predictors and criterion. The contribution to the data matrix by the scores vector is determined via regression, and consequently the original data matrix is reduced by subtracting this part away. Similarly, the dependent data matrix is also adjusted. The above steps are repeated until the scores contribute very little to the original data matrix.

3.5.1 PLS Regression

Assume a set of principal component scores, **T**, can be calculated as follows (see *3.4.2-81*)

$$\mathbf{T}_{\text{scores}} = \mathbf{Z}^{*}\mathbf{W} \tag{3.5.1-87}$$

where the matrix $\mathbf{Z}^{*}$ is a set of data containing p independent variables and N observations, forming a collection of column vectors:

$$\mathbf{Z}^{*} \in \Re^{N\times p} \Leftrightarrow \mathbf{Z}^{*} = \left[\mathbf{z}_1^{*},\cdots,\mathbf{z}_k^{*},\cdots,\mathbf{z}_p^{*}\right],\ \mathbf{z}_k^{*} \in \Re^{N}$$

and the weight matrix, **W**, is such that

$$\mathbf{W} \in \Re^{p\times p} \Leftrightarrow \mathbf{W} = \left[\mathbf{w}_1 \quad \cdots \quad \mathbf{w}_j \quad \cdots \quad \mathbf{w}_p\right],\ \mathbf{w}_j \in \Re^{p\times 1}$$

Additionally, another score matrix, **U**, is

$$\mathbf{U}_{\text{scores}} = \mathbf{Y}^{*}\mathbf{Q} \tag{3.5.1-88}$$

where $\mathbf{Y}^*$ is a set of data containing m dependent variables and N observations and $\mathbf{Q}$ is a weight matrix of dimension $m \times m$.

The weight matrices are orthogonal, i.e., $\mathbf{W}^t\mathbf{W}=\mathbf{W}\mathbf{W}^t=\mathbf{I}_p$ and $\mathbf{Q}^t\mathbf{Q}=\mathbf{Q}\mathbf{Q}^t=\mathbf{I}_m$. Thus, solving for both data sets leads to

$$\begin{aligned} \mathbf{Z}^* &= \mathbf{T}\mathbf{W}^t \\ \mathbf{Y}^* &= \mathbf{U}\mathbf{Q}^t \end{aligned} \qquad (3.5.1\text{-}89)$$

Traditionally, error matrices, $\mathbf{E}$ and $\mathbf{F}$[iv], are included in the above equations to account for any source of inexactness. Including these results yields

$$\begin{aligned} \mathbf{Z}^* &= \mathbf{T}\mathbf{W}^t + \mathbf{E} \\ \mathbf{Y}^* &= \mathbf{U}\mathbf{Q}^t + \mathbf{F} \end{aligned} \qquad (3.5.1\text{-}90)$$

It is convenient to write the outer product formulation of *3.5.1-90* as

$$\mathbf{Z}^* = \sum_{k=1}^{p} \mathbf{t}_k \mathbf{w}_k^t + \mathbf{E} \qquad (3.5.1\text{-}91)$$

where

[iv] The matrix F is an error matrix, and not to be confused with a factor matrix.

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1 & \cdots & \mathbf{t}_p \end{bmatrix} \text{with } \mathbf{t}_k \in \Re^N$$

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1^t \\ \vdots \\ \mathbf{w}_p^t \end{bmatrix} \text{with } \mathbf{w}_k \in \Re^p$$

$$\mathbf{E} \in \Re^{N \times p}$$

and

$$\mathbf{Y}^* = \sum_{k=1}^{m} \mathbf{u}_k \mathbf{q}_k^t + \mathbf{F} \qquad (3.5.1\text{-}92)$$

where

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_m \end{bmatrix} \text{with } \mathbf{u}_k \in \Re^N$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1^t \\ \vdots \\ \mathbf{q}_m^t \end{bmatrix} \text{with } \mathbf{q}_k \in \Re^m$$

$$\mathbf{F} \in \Re^{N \times m}$$

Equations *3.5.1-91* and *3.5.1-92* can describe all the components when the summation is taken from 1 to p or 1 to m, respectively. In this case the error matrices are equivalent to $\mathbf{0}$, a zero-field matrix of dimension $N \times p$ or $N \times m$.

To connect $\mathbf{Y}^*$ to $\mathbf{Z}^*$, an inner relation is needed. Sometimes a linear fit is used, where

$$\mathbf{u}_k = b\mathbf{t}_k \qquad (3.5.1\text{-}93)$$

The coefficient, b, is a regression coefficient found by minimizing the residuals.

While a linear fit is acceptable, controversy arises over the way information is exchanged. That is, information is calculated separately for the two scores, hence, knowledge is not directly shared.

The idea behind PLS regression is to exchange information. This can be performed between $\mathbf{Y}^*$ and an arbitrary score vector $\mathbf{t}$; $\mathbf{t}$ is not a principal component score as before, but is yet to be determined. Naturally, one measure worth optimizing is the covariance between the two vectors ($\mathbf{Y}^*$ is assumed to be univariate). Since covariance may have sign information, plus or minus, the magnitude is usually taken. Hence, the goal is to maximize the covariance squared, or

$$\left|\mathbf{Y}^{*t}\mathbf{t}\right|^2$$

Expanding the above in terms of the weight vector, $\mathbf{w}$, leads to

$$\left(\mathbf{Y}^{*t}\mathbf{t}\right)^t\left(\mathbf{Y}^{*t}\mathbf{t}\right)=\mathbf{w}^t\mathbf{Z}^{*t}\mathbf{Y}^*\mathbf{Y}^{*t}\mathbf{Z}^*\mathbf{w} \qquad (3.5.1\text{-}94)$$

Similar to the concept behind principal components analysis, the maximum covariance is obtained when

$$\begin{aligned}
&\frac{\partial}{\partial\mathbf{w}}\left[\mathbf{w}^t\mathbf{Z}^{*t}\mathbf{Y}^*\mathbf{Y}^{*t}\mathbf{Z}^*\mathbf{w}-\lambda(\mathbf{w}^t\mathbf{w}-1)\right]=0\\
&=2\mathbf{Z}^{*t}\mathbf{Y}^*\mathbf{Y}^{*t}\mathbf{Z}^*\mathbf{w}-2\lambda\mathbf{w}=0\\
&\Rightarrow(\mathbf{Z}^{*t}\mathbf{Y}^*\mathbf{Y}^{*t}\mathbf{Z}^*-\lambda\mathbf{I})\mathbf{w}=0
\end{aligned}$$

Thus the solution to the above reduces to an eigenvalue/eigenvector problem.

However, the regression analysis is carried out in steps. This is the main difference separating the PLS and PCR techniques. The stepwise computation allows for exchanging of information. Note: in some older derivations regression is carried out between the two score vectors, **t** and **ú**.

Thus, only the largest eigenvalue determines the first eigenvector, **w**. Although $\mathbf{Z}^{*t}\mathbf{Y}^{*}\mathbf{Y}^{*t}\mathbf{Z}^{*}$ is a matrix, the remaining latent roots should be zero since it reduces to a rank 1 structure. Once **w** is found the scores vector, **t**, is calculated using

$$\mathbf{t} = \mathbf{Z}^{*}\mathbf{w} \tag{3.5.1-95}$$

Because only one score vector was determined, its contribution to $\mathbf{Z}^{*}$ must be found. Equation *3.5.1-91* lists all principal component score and weight vectors, hence, a loading vector is needed to find the contribution of **t** to $\mathbf{Z}^{*}$. If **p** is the loading vector, then

$$\begin{aligned} \mathbf{Z}^{*} &\cong \mathbf{t}\mathbf{p}' \\ \mathbf{t}'\mathbf{Z}^{*} &= \mathbf{t}'\mathbf{t}\mathbf{p}' \\ \mathbf{p}' &= \frac{\mathbf{t}'\mathbf{Z}^{*}}{\mathbf{t}'\mathbf{t}} = \frac{\mathbf{t}'\mathbf{Z}^{*}}{\|\mathbf{t}\|^{2}} \end{aligned} \tag{3.5.1-96}$$

A similar step is used to find the regression coefficient **q** (a 1 dimensional loading vector) on $\mathbf{Y}^{*}$, where

$$\begin{aligned} \mathbf{Y}^{*} &\cong \mathbf{t}\mathbf{q}' \\ \mathbf{t}'\mathbf{Y}^{*} &= \mathbf{t}'\mathbf{t}\mathbf{q}' \\ \mathbf{q}' &= \frac{\mathbf{t}'\mathbf{Y}^{*}}{\mathbf{t}'\mathbf{t}} = \frac{\mathbf{t}'\mathbf{Y}^{*}}{\|\mathbf{t}\|^{2}} \end{aligned} \tag{3.5.1-97}$$

All that is left is to subtract the contribution of the score vector **t** to each of the matrices, $\mathbf{Z}^{*}$ and $\mathbf{Y}^{*}$. The necessary steps are

$$\mathbf{Z}^*_{new} = \mathbf{Z}^* - \mathbf{tp}'$$
$$\mathbf{Y}^*_{new} = \mathbf{Y}^* - \mathbf{tq}' \qquad (3.5.1\text{-}98)$$

This completes one step of the regression algorithm. Equations *3.5.1-95* through *3.5.1-98* are repeated until no information is left in the residual matrices, $\mathbf{Z}^*_{new} - \mathbf{Z}^*$ and $\mathbf{Y}^*_{new} - \mathbf{Y}^*$.

The number of steps performed is sometimes referred to as the number of factors (Qin,1992). If too many factors are used, the model tends to overfit the data. For proper generalization of the data and to avoid over-fitting, cross-validation methods are normally applied.

3.5.2 Selection of Principal Variables

PLS can also be used to select the most important independent variables (Höskuldsson, 1996). This method requires using equation *3.5.1-94* and setting the weight vector w for a single co-ordinate that is one. For example,

$$\mathbf{w} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

will select a column of $\mathbf{Z}^*$ such that the maximum covariance of $\mathbf{Z}^{*t}\mathbf{Y}^*\mathbf{Y}^{*t}\mathbf{Z}^*$ is chosen.

Chapter 4

State Variable Analysis

4.1 Preface

This chapter is concerned with selecting state variables and the mapping of these to heat rate. It is a preliminary stage necessary to preprocess data, refine computer codes, examine model fidelity, determine output response, etc. It is also necessary to define the number of common factors required as predictors and identify outliers in the observation space. To quantify the heat rate, the state variables are input to three types of efficiency models implementing partial least squares, neural networks, and multiple regression. Based on state variable analysis alone, anyone of the pre-determined models may have poor performance characteristics.

Initially, 48 state variables were selected out of a total number of 293 variables. Data was preprocessed with computer code, File 11 preprocessv2.m, listed on page 286. The set was further reduced to 43 state variables because five of the 48 variables were considered criterion outputs, hence, they were eliminated as predictors. The remaining 43 state variables formed an input set consisting of data scaled as raw scores, deviates or small deviates, or in standardized or small-standardized form. The input set was also converted to common factor and principal component scores. Table A-3 lists the names of the 43 state variables.

4.2 Computing Factor Scores

Variables $\mathbf{Z}$ are related to the loadings $\mathbf{A}$ and common factors $\mathbf{F}$ by

$$\mathbf{Z} = \mathbf{AF} \tag{4.2-99}$$

If the observable variables are used in estimating the factor measurements, then computation of the factor scores is

$$\mathbf{F} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{Z} \qquad (4.2\text{-}100)$$

Equation *4.2-100* is a reasonable approximation when the observed variables are considered "ideal" variables. Hence, the method is otherwise known as estimation of ideal variables. All common factor measurements were calculated using Equation *4.2-100*, with the exception of a few used for comparison. Some of the other ways to calculate factor scores are given by Harman [23], they are:

1. Factor scores computed by principal components
2. Common factor scores computed by rotation of principal components (e.g., varimax form)
3. Common factor scores computed by estimation of regression,

$$\mathbf{F} = \mathbf{A}'\mathbf{R}^{-1}\mathbf{Z} \qquad (4.2\text{-}101)$$

4. Common factor scores computed by theoretical model,

$$\mathbf{F} = (\mathbf{A}'\mathbf{U}^{-2}\mathbf{A} + \Theta^{-1})^{-1}\mathbf{A}'\mathbf{U}^{-2}\mathbf{Z} \qquad (4.2\text{-}102)$$

5. Common factor scores computed by minimizing unique factors

$$\mathbf{F} = (\mathbf{A}'\mathbf{U}^{-2}\mathbf{R}\mathbf{U}^{-2}\mathbf{A})^{-\frac{1}{2}}\mathbf{A}'\mathbf{U}^{-2}\mathbf{Z} \qquad (4.2\text{-}103)$$

While it has been assumed throughout this report that the observed variables are linearly related to one another, i.e., the variables are linear combinations of the factors, it could be possible for the data to be non-linear. Under these circumstance new equations would have to be derived to fit the data.

4.2.1 Addressing Indeterminacy of Factor and Pattern Results

Pattern matrices were computed for 1, 2, 3, 4, 5, 6, 7, and 10 common factors using the *minres* method (see page 50), with initiatory pattern matrices consisting of the first eigenvector components. Factor scores were also estimated using Equation *4.2-100*. The factor matrices were not rotated to canonical form.

Naturally, the problem of indeterminacy is of utmost importance. The correlations between a set of variables will yield an infinite set of coefficients for the factor pattern. It is because of the indeterminacy in the model that the pattern loadings are not unique, and so the estimation of factor scores also differ. However, the following results show factor scores retain similarity and prediction.

Although no in-depth study of matching or classifying factors was made, Figure A-1 confirms visually the relationship between factors. The graph shows plots of 4 factor scores. Four separate pattern matrices were computed with *minres*; however, random data was used instead of eigenvectors for the initial patterns. In each case, the patterns resulted in a different ordering of five factor scores. Thus, pattern #1 produced five factor scores, labeled F1, F2, F3, F4, and F5. Pattern #2 also had five factor scores, labeled F1, F2, F3, F4, and F5. And so on, for patterns #3 and #4, respectively.

Three of the four scores in Figure A-1 are offset by 5 ordinate units to allow for better inspection. The first line (signal 1) was computed with pattern #1, and represents the first factors score (F1). The second line (signal 2) was computed with pattern #2, and also represents the first factors score (F1). While the two signals are not exactly identical, the shape is similar and it appears one is a reflection of the other. The third line (signal 3) was computed with pattern #3, and represents the fifth factors score (F5). With the exception of random fluctuations, or noise, it is congruent to signal one and is reflexive with signal two. The last signal, pattern #4, factor score one (F1), is congruent to signal 2 and reflexive with signals 1 and 3.

All four factor scores were obtained with four different patterns; yet, all share either congruency or reflectivity with one another. It is postulated that the preservation of correlation structure,

$$\mathbf{R} = \mathbf{A} * \mathbf{A}' \qquad (4.2.1\text{-}104)$$

also retains similarity between factor scores. If this true, then combinations of similar factor scores should provide the same prediction with little or no variation.

4.2.2 Factor Score Prediction

Another calculation was performed using six sets consisting of five factor scores each. The factor scores were inputs to a multiple regression model, with the criterion variable being heat rate.

The factor scores were estimated in the usual way, by way of Equation *4.2-100*. Patterns were computed using the *minres* method with random data for the initial pattern matrices. Correlation structure was preserved according to Equation *4.2.1-104*. The mean squared error using the six input sets is shown in Table 2.

Table 2 MSE values for factor score prediction

Pattern #1 Factors F1,F2,F3, F4,F5	127.88
Pattern #2 Factors F1,F2,F3, F4,F5	127.89
Pattern #3 Factors F1,F2,F3, F4,F5	127.86
Pattern #4 Factors F1,F2,F3, F4,F5	127.91
Pattern #5 Factors F1,F2,F3, F4,F5	127.89
Pattern #7 Factors F1,F2,F3, F4,F5	127.93
Average	127.89
Variance	.00040

The variation over the entire range of values is extremely small, reinforcing the idea that combinations of like factor scores provide the same prediction with little or no variation.

An even more important conclusion follows when the same numbers of factor scores are used and the structure is preserved, inference estimates are unequivocal. Hence, the ordering and small similarity difference of factor scores doesn't bias the prediction result.

4.3 PLS Estimates

Figure A-2, Figure A-3, Figure A-4, and Figure A-5 show the PLS model estimate against the criterion heat rate. The inputs are raw scores, 43 state variables (400 observations) with no pre-scaling. As the number of stages increase, MSE declines. Figure A-6 shows the negative exponential tapering of MSE levels. An asymptotic value of approximately 67 is reached as the number of stages approaches 10-12. The use of

standardized scaling improves the PLS estimate for fewer number of stages, see Figure A-7. PLS yields an MSE value of approximately 46 when all 43 stages are implemented and scores are standardized. This limiting value agrees with MR estimates given in Figure A-29 and Figure A-31. MR responses are equivalent when using either standardized or raw scores. By contrast, PLS models using raw scores achieve higher MSE values.

In Figure A-8 predictors for the PLS model estimate are factor scores (5 dimensional), with SD pre-scaling. It is interesting to note that only 1-2 stages are necessary to reach a minimum MSE, see Figure A-9. Essentially all the information is mapped to heat rate within one stage. Figure A-10 shows the heat rate prediction when using the first major principal component scores (5 dimensional). Corresponding with factor inputs, only a few stages are necessary to reach a limiting MSE value, see Figure A-11.

It should be noted that the nature of the error term is not examined, nor are its components: measurement and stochastic error. The purpose here is to make conclusions with a high degree of generalizabilty. As such, it can be safely said that prior standardization of inputs is necessary when implementing PLS models. Another advantage of PLS over MR or NN models is one can limit the number of stages. This can have a pronounced affect on the introduction and/or elimination of stochastic error.

4.4 Deciding Upon the Number of Common Factors or Principal Components

Interpreting how many dimensions to retain is difficult. Many experimenters make a plot of the eigenvalues, see Figure A-12. A cutoff is determined when the eigenvalue magnitude falls below one. For the 43 state variables, this is about 5-6 factors. Mol [33] presents a more rigorous method for deciding the preliminary number of essential factors for a given data matrix. It utilizes both empirical and random eigenvalues. He concludes that, "rotational considerations must be in the end the most important considerations (besides the statistical ones) in deciding the number of factors... In the case too many factors are extracted, it might lead to the conclusion that superfluous error variance must have stealt in."

Based upon when error variance appears, a somewhat simplified statistical approach for resolving the number of common factors is given later on in Section 4.7.

Figure A-13 shows a Scree plot of the major principal components. The percent total variance is tallied for increasing numbers of principal factors. For the 43 state variables, about 5-6 principal components should be kept. This occurs when about 90% of the total explained variation is accounted for.

4.5 Neural Network Estimates

Unless otherwise noted, the neural network object consisted of a 3-layer NN (1 input, 1 middle or hidden, 1 output) with 2 hidden nodes (tansig activation functions) in the middle layer. The default training algorithm was Levenberg-Marquardt. Validation vectors were used for improving generalization; a technique also called *early stopping*. After a specified number of iterations after the validation error begins to rise above the training set error, training is stopped, and the weights and biases at the minimum of the validation error are returned. Emphasis was placed on using validation vectors for all neural network training, except in a few circumstances. In cases where over-training was desired to examine how directionality changed, identical training and validation vectors would be used. The purpose intended here was to achieve a sort of localized *early stopping*. Separate test sets were not used in the state variable analysis because the major emphasis is placed on comparison of different types of predictors, which is best performed by having the training and test set equivalent; however, they are implemented in the chapter on efficiency prediction using set-point variables. The motivation for test set error is used to compare different models (NN, PLS, and MR) and also indicate poor division of the data set. Since it is difficult to determine how much the estimate error is when the model is misspecifed, this problem is partially alleviated by using a separate test set.

Figure A-14, Figure A-15, Figure A-16, Figure A-17, Figure A-18, and Figure A-19 display neural network predictions for heat rate. Predictors are factor scores (5 dimensional) with SD pre-scaling. The results shown in Figure A-14 were obtained with validation vectors every fourth position, except observations 216-289 (period from July 5th, 1998 to September 16th, 1998). The lack of sufficient training in the summer

region shows exceptionally poor results. When all the points in the summer region are included as both training and validation vectors, see Figure A-15, the NN prediction is good. An MSE value of 83 is obtained. The MSE does not improve for validation vectors taken over the entire observation period at every 4 intervals, see Figure A-16. The shape of the NN signal appears choppy in the summer region and also tends to spread out at both ends of the observation period. This divergence at the endpoints is also present in all other NN graphs. In Figure A-17 validation vectors were assigned at every 4th interval, however, the validation vectors were not used during training of the neural network. Only the training set was implemented, with points taken every first, second, and third positions. The match between the NN output and sensor signal is very good. However, there is the possibility of overfitting. Figure A-18 and Figure A-19 show NN predictions using localized early stopping. A small amount of random noise was added to the validation predictor and target values. Both graphs achieve a good fit, fairly low MSE, and fast training times. As in the case of Figure A-17, the chance of over fitting exists. Adding noise to input data has been favorable in creating a more robust NN model, [Wrest, 52]. However, whether or not corrupting the validation set achieves a localized robustness is still questionable.

Mean squared error values for other NN simulations, comprising various combinations of middle layer nodes, validation intervals, and noise addition, are listed in Table A-1. It is very difficult in assessing which method is better. The highest MSE, which is about 234, provides a relative upper bound on the NN prediction ability, or roughly 0.15%. Allowing for additional nodes in the middle layer tends to reduce the MSE. However, too few middle nodes are probably not advisable since the predictor data has already been compressed by a factor of 88%. Four to five middle layer nodes with validation vectors manifests an ideal combination. Figure A-20 and Figure A-21 show results using a 4-layer middle neural network. Heat rate prediction appears well defined in the summer region and a low MSE is obtainable. Not only is MSE a crucial factor, but NN training times are also important. Excessive training times are not favorable when they lead to little or no improvement of the MSE over faster methods. The use of validation vectors improves training times significantly, in addition to the benefits already inherent to the neural network, i.e., improved generalizability.

4.6 Multiple Regression Estimates

A short note about multiple regression is needed because of its importance in prediction analysis. The regression equation is a mathematical model of the mapping of predictors to criterion. It differs from other analysis techniques, such as correlation analysis, in that regression analysis shows *how* variables are related, whereas correlation analysis shows the *degree* to which variables are related. Compared to other linear modeling techniques, e.g., PLS, the multiple regression estimate has the minimum variance. This follows from the Gauss-Markov Theorem. The Gauss-Markov Theorem [Wonnacott, 50] states that, "within the class of linear unbiased estimators of beta or alpha, the least squares estimator has minimum variance." Hence it follows that biased or nonlinear estimators may be better. This implies that in the limiting case (as the number of stages approaches the rank of the matrix **X'YY'X**), linear PLS estimation should achieve multiple regression MSE values. In addition, pure linear NN estimates should be equivalent while non-linear NN estimation an improvement.

Figure A-22 plots the MR estimate against heat rate using five factor scores as predictors. An MSE of 127 was calculated. This number is of course equivalent with that determined in the limiting case of PLS, see Figure A-8. Beta coefficients along with error bar plots of 95% confidence intervals about residuals are shown in Figures A-23 and A-24. Two important observations are drawn. First, the mapping at both ends of the entire observation period is partially divergent. In the same way, NN estimates in Figures A-14 - A-21 show the separation occurring at the end points. Because NN estimates using factor scores predicted lower overall MSE values, this tends to imply partial non-linear mapping during training of the neural network. However, multiple regression is a pure linear technique, and because the separation occurring at the end points is common to both models, it implies the fault lies in the factor scores themselves. Several reasons are likely explanations. It is possible that during transformation of scores from the input space to the factor space information was simply lost. Or, since reduced dimensional factorization provides enhanced communality and at the same time removes specificity and error variance from the input space, it is likely that the mapping is correct and the removal of information was noise. The latter assumption is supported by the experimental predictions in Figure A-25 and Figure A-26. A pure linear NN mapping is shown in Figure A-25. As expected, the MSE is equivalent to the MR model. This result serves to validate

the linear mapping between the two neural networks and multiple regression models. If noise is primarily resident in the minor principal component scores, then using five of the major principal components as predictors should indicate a similar trend at the endpoints. This is verified by the prediction results in Figure A-26. Hence, it is assumed some degree of stochastic error was removed from the factor variables. Secondly, most of the outliers exist in the summer region defined by observations 216-289. Because the summer region is wildly chaotic, heat rate changes rapidly from day-to-day, the functional dependence appears noisy. An increase in MSE using factor scores would be accounted for in this region. A similar error is to be expected when using principal components.

Furthermore, Figure A-26 points out a striking observation. Artifacts or sharp spikes, which up to now have not been mentioned, appear at certain times. The largest one occurs at observation number 257, however, it is not assumed a spike because of the overall movement of predictor values fluctuating in the region. However, an artifact appears at observation number 160. Several predictors in the excel spreadsheet had missing data points. These variables were flagged as having bad indices, but were not corrected in this part of the analysis. One of the reasons was to determine if outliers influence the factor/principal space, and secondly, to see if prediction results would be affected.

Figures A-27 and A-28 show the beta coefficients along with error bar plots of 95% confidence intervals about residuals for principal component predictors. Although an outlier is indicated in Figure A-28 at observation number 257, because of the reason given above, it is not considered a corrupted value. Still, a majority of outliers appear in the summer region, see Figure A-24 and Figure A-28.

If predictors are 43 state variables, SD pre-scaling, the MR MSE is the lowest attainable, having a value around 46, see Figure A-29. In the general case, neither NN or PLS models arrive at the same value, except in two instances. One is in the limiting case of PLS, i.e., as the number of stages approaches maximum, and the second is if the NN model has all linear activation functions. Either one validate the linear mapping situation with multiple regression. The beta coefficients shown in Figure A-30 are higher weighted to variables 39, 40, and 41. These correspond to HTR DRN PMP A EFF DEV, HTR DRN PMP

B EFF, HTR DRN PMP B EFF DESIGN. When no pre-scaling is implemented, the MR prediction results are identical, see Figure A-31. This is contrary to prior results obtained with PLS. Beta coefficients are plotted in Figure A-32. Notice the first coefficient refers to a negative quantity. This means the regressors are invalid outside of their domain. Error bar plots of 95% confidence intervals about residuals are shown in Figure A-33.

Interactive fitting and visualization of the response surface for linear multiple regression estimates may be done using MATLAB. Figure A-34 is a snapshot of the MR an example when predictors are factor scores (5 dimensional, sorted), SD pre-scaling. Prediction results are much more sensitive to the first two factors than others.

Finding guidelines that indicate a few specific regressors are appropriate are usually nonexistent. In practice, most experimenters pick up the regressors that most highly correlate with the criterion. The MR prediction is very good using five of the highest correlated state variables with heat rate, see Figure A-35. All of them should be used in MR, rather then test one at a time with any sort of stepwise approach. A stepwise method is basically starting from scratch and adding one significant regressor at a time. The stepwise approach is valuable in limiting the number of regressors. However, it has a clear disadvantage in that it tends to discriminate against regressors tested last, even if it is correctly applied; and if incorrectly applied, it sets apart the last regressors even more. The situation is most pronounced with collinear regressors, since it is difficult to establish statistical significance. Fortunately, one of the advantages of implementing factor or principal component scores is that the problem of multicollinearity is not present. The problem of multicollinearity occurs when the independent variables are collinear, or nearly so. Then it is difficult to predict for values of the independent variable (regressors or predictors) removed from their line of collinearity, and the relation of the criterion to the independent variables cannot be sensibly investigated. However, with factor or principal component scores the structural questions can be answered, i.e., the relation of criterion to predictors. While the problem of multicollinearity appears to be solved, unfortunately another issue arises, i.e., the relationship of the score to a physical quantity. In the principal component case, the scores represent a region of total variation. For factor scores, it is simply hypothetical constructs. Neither of these have a true physical meaning as in the sense of the original independent

variables. Hence, a sensitivity analysis or interactive fitting as in Figure A-34, will provide information on hypothetical quantities. Whether or not confidence intervals are applied for regressors in this situation, as seen in Figure A-24 and Figure A-28, is debatable.

In conclusion, interpreting an estimated regression requires good judgement and prior theoretical understanding of the model being tested, along with an understanding of the assumptions and limitations of the statistical techniques.

4.7 Multiple Regression Estimates using Factor and PCA Scores

Table A-2 lists multiple values of MSE for different factor scores. A 33% improvement in MSE is noticed when five factor scores are applied instead of four; this is a considerable difference. Only a 2% change is accounted for when using six factor scores instead of five. One explanation possible for the huge difference is that at least five factors are necessary to capture most of the information between all predictors. As seen by the R^2 values, there is almost a 10% increase in variability in the observations accounted for by the MR model when using five factors instead of four. Very little benefit is seen with the use of six or more factors; the delta R^2 improvement amounts to a change in variability of only 1%, see Figure A-36. MSE results from some of the other ways to calculate factor scores are also shown in Table A-2. These are in good agreement with previous calculations if the uniqueness matrix is determined from the patterns.

Not surprisingly, all 43 principal component scores are necessary to achieve the lowest obtainable MSE. This is equivalent to earlier MR estimates, which were using all 43 state variables. A vast improvement over working with just the five highest major PCA components is to employ those that correlate highest with heat rate. In this situation the MSE will drop down from 171 to 111, a decrease of 35%. Unfortunately, two of these components are below the constant one sigma line.

Chapter 5

Set Point Variable Analysis

5.1 Preface

This chapter is concerned with the mapping of set point variables to heat rate. This is accomplished first by mapping the set point variables to state variables. Because the mapping of state variables to heat rate was covered earlier in Chapter 4, combining the two layers provides a functional mapping from set point variables to heat rate, see Figure 5. The middle stage is necessary to examine the temperature and pressure limits of the system (represented by state variables) when small changes are applied to the set point variables (control variables).

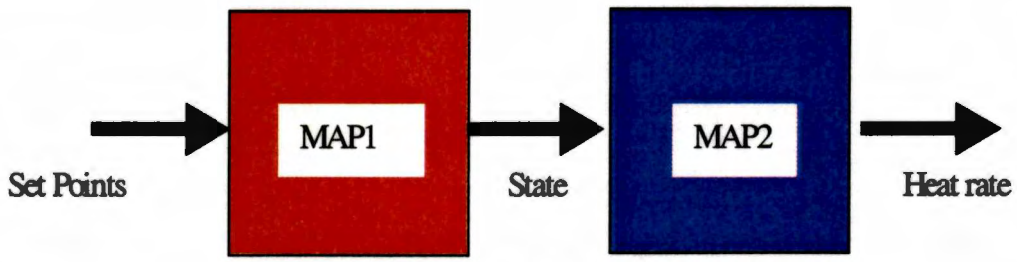

Figure 5 Modeling Layout

Out of a total number of 244 variables designated as set points by Arizona Palo Verde personnel, 76 were selected. During preprocessing, 168 of these fell below a predetermined cutoff variance that was set at a value of one. Without significant variation these signals would contribute little or no information, and hence, were eliminated. The set was further reduced to 55 set point variables -- 21 of the variables were essentially duplicate signals having a correlation > .95 with other signals; they were eliminated as predictors in order to prevent progressive degenerate regression (or multicollinearity) -- this topic is discussed in detail later. The remaining 55 set point variables formed an input set consisting of data scaled as raw scores, deviates or small deviates, or in standardized or small-standardized form. The input set was not converted to common factor or principal component scores. The descriptions of the 55 set point

variables are shown at the end of this chapter in Table A-11 on page 202; Table A-13 lists the 21 signals that contained identical information and were subsequently removed.

5.2 Predictor Perturbations

Predictors in the first mapping were normally increased in value by either +5% or +.5%. The magnitude of the increase was slowly ramped up from observation 100 (0% change) to observation 110 and beyond (100% change or maximum increase). A smaller perturbation would keep most values within the predictor's domain, see Figure A-38. For some signals the size of the perturbation's magnitude made no difference since values fell outside the domain anyway, see Figure A-39. This didn't present a problem when linear mapping modes were used. A corresponding change in heat rate for a set point variable is reflected in Figure A-40 for a 5% perturbation.

5.2.1 PET and MPET Units

Units are given in PET, which is adopted from Stark [40]. PET is the percentage error defined as

$$Error_{PET} = Error_{relative} \times 100 \qquad (5.2.1\text{-}105)$$

where relative error is defined as

$$Error_{relative} = \frac{(\text{Calculated value}) - (\text{True value})}{(\text{True value})} \qquad (5.2.1\text{-}106)$$

Sometime later MPET was adopted. The term MPET follows the same mathematical definition as above except it is the mean of PET values over a particular observation range. For most calculations this would occur at 100% maximum increase, or observations 110 to 400.

5.2.2 Predictor Perturbations Specific to Neural Networks

Neural networks were extremely sensitive to data that was outside the training region's domain by providing varying results. The change in heat rate in Figure A-40 is shown again in Figure A-41 using a neural network mapping for MAP2. Figure A-42 shows the heat rate estimation using PCA components as predictors. Both graphs were generated using a 5% input perturbation. A .5% change resulted in another set of estimations. There was a general lack of consistency for different number of epochs, hidden nodes, cross-validation points, etc. Specific MATLAB neural network training routines, such as: *trainlm*, *trainoss*, *trainrp*, *trainscg*, *trainwb*, and *traincgf*, displayed mixed outcomes.

The affects of increasing an input signal outside the neural networks training domain are seen in Figure A-43. Comparing this with Figure A-39 one notices failure of prediction in the region. It is interesting to note that linear mapping techniques did not exhibit any anomalous transition in the same vicinity, the lack of prediction was solely attributable to the use of neural networks. The overall mapping, however, was extremely good with neural networks, see residual plots of heat rate in Figure A-44. Better results were obtained with fewer PCA components, see Figure A-45; with 6, 7, 8, or more PCA components the neural network would revert back to a mix of heat rate estimations. Unfortunately, this is an adverse problem believed to be contributive to the higher frequency PCA component modes. Directionality was not preserved between FACTOR and PCA scores, as indicated in Figure A-46 and Figure A-47. This was puzzling.

In the case of negative perturbations, the inverse mapping was fairly consistent except near the boundaries, see Figure A-48 and Figure A-49 -- the problem of offset mapping near the boundary was also noticed during the state variable analysis. Aside, the definition of inverse mapping is used here in the sense of producing a reverse or inverted image, not a true inverse where inputs and outputs are exchanged. Calculated MPET values were 0.10501 and -0.10443, respectively. The difference in magnitude is plotted for all 55 set point variables in Figure A-50. Variables FW PMP A IN ENTH, FWPT B HP SV CHEST TEMP and S/G PRESSURE 1 provided the poorest inverse mapping and had the largest MPET variation. An examination of the set point signals, before and after perturbation (see Figure A-51, Figure A-52, and

Figure A-53), revealed the smaller 0.5% increase was outside the domain for most signals. Reducing the magnitude of the perturbations to 0.05% helped, still a noticeable -- but reduced -- difference in MPET values remained for inverse mappings, see Figure A-54. The overall heat rate predictions appeared better, see Figure A-55 and Figure A-57. Comparing Figure A-57 to Figure A-55, most of the spikes seemed to occur at points where the signal rapidly changed directionality. This phenomenon was confusing since the mapping using simple neural networks was in MAP2.

What occurred when overtraining the data is shown in Figure A-57 and Figure A-58. Although spikes tend to appear in these plots, the underlying general features approach those for MR-MR mappings. For these situations the use of a median PET over MPET serves as a better indicator of heat rate performance. Simple to use, the median PET value tends to evaluate to the constant step characteristic of the increase and ignores the alternating high and low points.

It is concluded that because of the high number of sharp directionality changes that exist not only in the state variable domain, but also in the set point domain, an insufficient amount of training data is one of the primary causes for the opposite behavior of the neural network predictions. Compensation for the lack of training points can be partially overcome by implementation of cross-validation, however, for best results the number of observations per day should be increased at least threefold to better represent the morning, afternoon, and evening operational loads.

5.3 MR-MR, PLS-PLS Mappings

Multiple regression mappings produced consistent results. The consequence of perturbations falling outside the input domain end up showing a linear change in heat rate. Because of this, the heat rate would not change rapidly and remained fairly constant from full onset of perturbation to the end. Changes in heat rate for different perturbation magnitudes could be scaled appropriately. For example, an MPET value of 0.7 at +5% perturbation would have a value of 0.07 at +.5%, and so forth. Negative perturbations involved

only a change in sign. In spite of all this, unusual observances were noted with MR mappings in MAP2. It is of primary importance and is discussed next.

5.3.1 Independence of the Type of State Variables

Heat rate predictions should be independent of the type of state variables represented in MAP2. Whether Factor, PCA, or SD scores are used should make little difference. Figure A-59 shows the change in heat rate for a +5% perturbation in each set point variable using 5 dimensional FACTOR scores. Likewise, Figure A-61 and Figure A-63 show the change in heat rate for 5 dimensional PCA and 43 SD scores, respectively. Sensitivity plots are shown in Figure A-60, Figure A-62, and Figure A-64.

Ten of the set point variables that exhibited the highest change in heat rate are listed in Table A-4 along with MPET values. While there appears to be some agreement between PCA and FACTOR scores, this is not true with SD scores. Two other types of predictors -- five variables that correlated highest with heat rate and five principal variables -- are essentially the same variables and yield identical results; these were disregarded in future analyses because of their high collinearity with heat rate and high multicollinearity (see 5.3.2). This leaves the remaining three scores: FACTOR, PCA, and SD.

Aside from any logical explanation, which is nonexistent, why using SD scores gave different results was somewhat puzzling. The next step was to determine if any changes were noticed using PLS. Table A-5 lists the ten most important variables for FACTOR, PCA, and SD scores with partial least squares mappings. Instead of using all stages for both MAP1 and MAP2, the number of stages was determined by the squared magnitude of the maximum covariance between X and Y deviations. Mathematically,

$$\left(s_{XY}^2\right)=\left(\left(\frac{\sum xy}{n}\right)^2\right)$$

$$\left(\frac{\mathbf{x}'\mathbf{y}\mathbf{y}'\mathbf{x}}{n^2}\right)=\begin{bmatrix} s_{x_1y}^2 & s_{x_1y}s_{x_2y} & s_{x_1y}s_{x_3y} \\ s_{x_1y}s_{x_2y} & s_{x_2y}^2 & \vdots \\ s_{x_1y}s_{x_3y} & \cdots & \ddots \end{bmatrix} \qquad (5.3.1\text{-}107)$$

$$Cutoff \approx \text{maximum eigenvalue of matrix } \mathbf{x}'\mathbf{y}\mathbf{y}'\mathbf{x}/n^2$$

The answers listed in the top row of Table A-5 are for a *cutoff* value less than or equal to one. Although this value is changeable, for most problems the mapping with a *cutoff* less than this provides no added benefit. Other possibilities include looking at the change in the *cutoff* from one stage to the next, which is examined later on. Note that the results shown in the second row use a fixed number of stages. When all stages were used for each PLS mapping, the results were identical to those given for MR-MR mappings. Obviously, from inspection of the results in Table A-5, it is apparent that a considerable discrepancy exists for SD scores.

In the top row of Table A-5 all three types of scores produced comparable answers, which is hypothesized true – "heat rate predictions should be independent of the type of state variables represented in MAP2." Using more stages (2nd row) should contribute little towards the mapping since the *cutoff* is less than one. As expected, the PCA and FACTOR scores are more or less independent of the number of stages. However, the case is not the same for SD scores, especially when additional stages are added. A possible explanation for this behavior is simply more information is contained in the predictor matrix. In fact, examining the results by varying the number of PCA and FACTOR dimensions, see Table A-6 and Table A-7, show that as more information is added, the predictions change. Another measurement reinforcing this point is when all PCA components are used as inputs the results are identical with those of SD scores under MR-MR mappings. The likely conclusion is FACTOR and PCA scores remove information from the predictor matrix that has a direct relationship on not only the prediction of heat rate, but also the degree of influence those inputs have on the prediction. Unfortunately, if this statement is true then the data shown in the top row of Table A-5 is incorrect. Two of the three types of scores (PCA and FACTOR) may have like

answers because of their similarity in technique, but the SD results should differ because more information is contained in the scores. Naturally, this would explain the behavior seen earlier under MR-MR mappings.

5.3.2 Multicollinearity

When the predictor matrix variables are collinear, it is called the problem of multicollinearity. In this case, for prediction purposes, it does not hurt provided there is no attempt to estimate heat rate for predictor values removed from their line of collinearity. However, underlying structural questions cannot be answered, i.e., the degree of relationship of the state variables to that of heat rate. For example, in simple regression the vectors **x** and **y** are:

X values	Y values
1	5.9
2	6
3	6.3

The X and Y pairs are spread apart and the effect of X on Y can be determined. When the regressors show little or no variation, as seen in the next example, then the effect of X on Y cannot be investigated. Now,

X values	Y values
1.95	5.9
2.0	6
2.05	6.3

This is a limiting case, in such that a mean value for Y can still be predicted from $\overline{X}$, however, the dependence of X on $\overline{Y}$ is impossible to determine because any number of differently sloped lines may pass through the point $(\overline{X}, \overline{Y})$ with the sum of squared deviations the same. The name for this phenomenon is called degenerate regression. In multiple regression the same thing happens except now there exists two or more independent variables. A numerical example for two regressors is

X values	Z values	Y values
1.95	1.96	5.9
2.0	2.01	6
2.05	2.05	6.3

The X and Z values are now bunched up on a line.

In simple regression the best fit for Y was not a line, but a point, whereas in multiple regression the best fit is not a plane, but a line. In both cases, collinearity makes one lose a dimension. As noted before, the prediction of heat rate is still possible, however, the relation of heat rate to the predictor variables cannot be examined.

5.3.3 High Collinearity of State Variables

It is imperative the degree of relationships be preserved in both MAP1 and MAP2. To check for multicollinearity in the state matrix a preprocessing technique was used to search for duplicate signals having a correlation greater or equal to 0.95. Out of the original 43 state variables, 22 matches were found. The duplicate signals are listed in Table A-13. The remaining set of 21 variables formed a new state variable matrix.

Because 22 duplicate signals out of 43 were found, the puzzling issues raised in Section 5.3.1 are believed to be the result of multicollinearity, the degree and extent to which are unknown. Pure multiple regression techniques are sensitive to multicollinearity due to the fact the inverse matrix, **X'X**, needs to be calculated. And while a simple rank estimation may indicate full rank, columns of the predictor matrix may be almost linearly dependent. It was hoped that by using PCA or FACTOR scores the issue of multicollinearity would be avoided, i.e., the regressors would be statistically uncorrelated. At least this would provide more precise confidence intervals and a greater certainty about which regressors are relevant. Unfortunately, removing dimensions removes information -- no matter whether the data matrix contains duplicate variables that measure nearly the same event or only a moderate degree of correlation exists. This is best illustrated by examining the change in heat rate while varying the number of PCA components. Figure A-65 shows the perturbation of two setpoint variables swap importance at the 14th component.

In contrast to what has been said up to this point and in complete disagreement with the idea of reducing the dimensionality of PCA or FACTOR scores, the use of fewer dimensions is not necessarily bad.

However, this applies only in specific situations. Case in point is the original 43 state variables. While it is difficult to relate a matrix factorization of higher and lower dimensions, a comparison of the major PCA components for the 43 and 21 state variable matrices shows information is driven into the first major component, see Figure A-66. It is possible for highly collinear data to be represented by the first major component. In such circumstances, the potential for pure prediction mappings is encouraging. Unfortunately, using a smaller number of dimensions from data matrices that have only a moderate degree of collinearity will remove criterion-predictor dependence information. In a standardized space this is what determines which variable is more or less important in influencing the prediction.

5.4 Final Results with Noncollinear state matrix

Table A-8 shows results when the state variable matrix contains 21 variables. The set point variable 'FW PMP A IN ENTH' in Table A-8 is significant for a number of major PCA components; this is not true as the number of PCA components increase. In Figure A-65 'FW PMP A IN ENTH' is compared to 'TRB HEAT RATE GROSS UNCORR'. Remember, the 21 state variables form a non-collinear data matrix and using fewer than 21 PCA components increases the probability that valuable information is lost. The information gain is most apparent at PCA component number 14, when the two variables switch position in importance. It is interesting to note that for SD scores, see Table A-8, the most important variables are similar to those determined in the multicollinear case implementing PCA and FACTOR scores. PLS results using two different methods of stopping criteria are shown in Table A-9. The SD scores with MR mappings are repeated in the last column of the first row of Table A-9. There is good agreement using SD scores for both MR and PLS mappings. If SD scores are not exploited, applying all the PCA components in MAP2 provides similar answers, see second row, first column of Table A-9.

So far, most of the computations involved working with a highly collinear data matrix (43 state variables) in MAP2 and then switching over to a reduced data matrix of 21 state variables to avoid the problem of multicollinearity. If this type of methodology in reducing collinearity were carried out further, an ongoing outcome would essentially see the absence of data. An experiment was performed to determine the most

important variables by selecting 12 arbitrary state variables out of the original 21. The results are shown in the second row, second column, of Table A-9. Although it is not rigorous, hopefully, this situation simulates a reduced collinearity situation in MAP2. Notwithstanding the possible errors, the highest prediction results are in agreement with PLS or MR cases.

Results for neural networks in MAP2 are shown in Table A-10. The top row lists variables for a +0.005 and +0.0005 perturbation change, respectively, while the bottom row shows variables sorted by median PET values. It was noted earlier that the median value tends to approximate the constant segment and should be a better indicator when implementing neural networks. In this case the most important variables differ substantially than those listed in Table A-9, and while it is difficult to judge, it appears some of these are more closely related to those obtained in a heat rate model based on a mathematical, first-principles type of approach.

5.5 Considerations for Future Work

There are at least four important topics disclosed in this report that should be designated for further research. The first two are considered difficult to investigate, the third is highly probable, while the fourth topic is achievable, but statistical in nature. These are:

1. *The collinearity information transfer associated with space transformations.* It was observed that data matrices having a high degree of multicollinearity can be reduced to those with a moderate or slight amount of collinearity by simply removing duplicate variables. In either case, a space transformation of the data matrix, such as FACTOR or PCA, will result in non-correlated variables. However, the original collinearity inherit to the original data is not removed. What happens to the collinearity and where it is transferred to in the new space may be of interest and is suggested as a topic of further research.
2. *The influence of collinearity on MR or PLS mappings.* Data matrices with a moderate or slight degree of collinearity provide fairly consistent predictions. However, the degree of

influence standardized variables have with the criterion(s) is especially important in determining those having the greatest importance. How collinearity induces an effect amongst the variables is of interest and is suggested as a topic of further research.

3. *Neural network inverse mappings and effects.* It was assumed neural network overtraining at individual observation points helped in stabilizing the results (approaching a linear condition). This was verified experimentally, and for certain perturbations of the input, overtraining provided partial areas or segments in observation space where the heat rate MPET values remained constant. However, spikes or erratic results were common. PLS and MR mappings provided an identical inverse image for a corresponding sign change in the perturbation. Naturally, this reflects a symmetrical mapping. However, the combination of regular (positive perturbation) and inverse (negative perturbation) neural network mappings did not yield a zero residual. Hence, it can be assumed directionality biases existed among neural network weights and are a reason for spikes appearing. At particular and forthcoming observations the neural network mapping might be heavily inclined in the wrong direction. Perturbations at these points could account for the appearance of spikes. Although these phenomena are hypothesized, a more in-depth study of inverse neural network mappings, within the observation space, is suggested as a topic for further investigation.
4. *PLS criterion in determining the number of stages.* Section 5.3.1 alluded to the initial development of a stopping criteria for PLS mappings based upon the maximum covariance between predictors and criterion(s). Since linear PLS mathematics readily allow calculating this value just prior to transforming the predictor data, it can be used in a number of ways. The first is as a maximum cutoff value in deciding how many PLS stages to implement. However, the change in magnitude between stages might also be desirable. This research is left as a topic for further investigation.

Chapter 6

Results

6.1 Preface

There are five main sections in this chapter. Section 6.2 briefly discusses modeling of the data. Section 6.3 covers important lessons learned in prior data analyses. Topics range from pre-scaling of data to model selection, and finally, implementation of stages for partial least squares. Discussion is primarily limited to concluding developments found earlier. The reader may review Chapter 4, State Variable Analysis, and Chapter 5, Set Point Variable Analysis, for more information. Section 6.4 introduces important findings and recommendations for the linear and nonlinear models using a single mapping layer. Afterwards, some interesting concerns about optimization are mentioned in Section 6.5. Lastly, Section 6.6 provides a summary of the most important results.

6.2 Methodology and Data Limitations

Fifty-five variables were designated as set points, forming the predictor matrix. A list of the setpoint variable descriptions can be found in the Appendix, located on page 202. Heat rate was the criterion variable. To quantify the heat rate, the predictor matrix was input to three types of models, implementing partial least squares (PLS), neural networks (NN), and multiple regression (MR).

Data came from Arizona's Public Service *Palo Verde Nuclear* Generation. Originally the data consisted of approximately 400 observation points at one-day intervals. It was further separated into four seasonal groups: Winter (11/09/97 to 2/28/98), Spring (3/01/98 to 6/30/98), Summer (7/1/98 to 9/30/98), and Fall (10/1/98 to 1/8/99). Seasonal variations include the periodic variation in cooling water inlet temperature or atmospheric effects on cooling towers and ponds. It mostly affects condenser pressure, although not used in a direct calculation of heat rate, condenser pressure can have a major influence on electrical output. Differences in heat rate may be noticed for the four groups, and normally are attributable to seasonal effects

provided there wasn't any correctable deficiencies or component degradation occurring during the time period.

Although it was initially expected that 400 observation points were sufficient to compare different model predictions, the small number of points restricted prediction to the training points themselves. Implications due to lack of sufficient data are many, and can impose severe limitations on prediction of heat rate within the training interval, the applicability of separate test sets for data validation, statistical tolerance, model prediction accuracy, etc. The fewer number of points used for seasonal training sets made this even worse. It should be noted that the amount of data examined is less than optimal conditions.

6.3 Lessons Learned

All data was scaled in standardized form. It was hoped that by using PCA or FACTOR scores the issue of multicollinearity would be avoided, i.e., the regressors would be statistically uncorrelated. At least this would provide more precise confidence intervals and a greater certainty about which regressors are relevant. Unfortunately, removing dimensions removes information -- no matter whether the data matrix contains duplicate variables that measure nearly the same event or only a moderate degree of correlation exists.

A single stage model was implemented over a two-stage model. Initially, the mapping of set point variables to heat rate was accomplished by mapping first to state variables, and then to heat rate. Combining the two layers provided a functional mapping from set point variables to heat rate. The middle stage was necessary to examine the temperature and pressure limits of the system (represented by state variables) when small changes were applied to the set point variables (control variables). Unfortunately, error produced in the first stage propagates into successive stages. Several workarounds were hypothesized to overcome the problem, however, the idea of a second stage was later dropped.

MR responses are equivalent to PLS estimates when using standardized scores. One possible advantage of PLS over MR or NN models is one can limit the number of stages. This can have a pronounced affect on the introduction and/or elimination of stochastic error. Instead of using all stages in the mappings, the number of stages was determined with a *cutoff* value (see 5.3.1 for more information), which is related to the squared magnitude of the maximum covariance between predictors and criterion. Although this value is changeable, for most problems the mapping with a *cutoff* value less than one provides no added benefit. Other possibilities include looking at the change in the *cutoff* from one stage to the next. All PLS results obtained in this chapter used the change in cutoff, which was set at a small value of 0.1 E-06.

6.4 Findings and Recommendations

Seasonal plots for heat rate estimates using a linear neural network with cross-validation, multiple regression, and partial least squares models are given in Figure A-68, Figure A-69, Figure A-70, and Figure A-71. The plot surfaces were drawn with highlights from a light source, where shading was based on a combination of diffuse, specular and ambient lighting models. Places where there are sharp contrasts in color normally indicate the surface topside and underneath layer -- indicated by blue/orange color regions. Results are in good agreement for all models; however, the summer region heat rate prediction shows a little more disparity between models. Linear neural networks can provide identical results with MR and PLS models if the training region encompasses the same observations. Linear NN, MR, and PLS predictions are shown in Figure A-72 using the same training points. Results are identical. Standard deviations are calculated for the three types of models and are shown in Figure A-68 through Figure A-72. The values provide a crude measure of the variability of the data within each observation region.

Seasonal heat rate modeling using a nonlinear neural network is compared to MR and PLS in Figure A-73, Figure A-74, Figure A-75, and Figure A-76. Residual standard deviations are calculated and shown for all plots. The residual is determined as the difference between one of the three model estimates and heat rate (criterion variable) calculated from the thermal model. A lower residual standard deviation implies less overall modeling error. All three model types yield good heat rate predictions.

Fifty-seven perturbations were made of the predictor matrix, one for every set point variable. In Matlab notation the perturbations spanned in percent from [[-0.05:.005:-.015][-0.01:.0005:+0.01][0.015:.005:.05]]. A number of –0.05 means holding all other set point variables constant and decreasing a particular set points' value by 5%. Similarly, a number of 0.005 implies increasing the set points' value by 0.5%. All predictor values were perturbed for each observation point in the seasonal group. Hence, heat rate estimates were calculated at every observation. Figure A-77 shows a three-dimensional winter plot of heat rate estimates using a NN perturbation model for the set point variable TOTAL FEEDWATER FLOW. Figure A-78 is the same graph excepted rotated to view the observation axis. Heat rate estimates using a linear perturbation model are shown in Figure A-79 and Figure A-80.

It should be emphasized the perturbation method above evaluates heat rate based upon instantaneous changes, i.e., increasing/decreasing a certain percent of the predictor value at each observation point. If the percent perturbation is taken relative to an average quantity then the perturbation magnitude over all observations reduces to a constant. Given a constant change in input with a neural network model, Figure A-81, Figure A-82, Figure A-83, and Figure A-84, show seasonal heat rate estimates for perturbation of the set point variable TOTAL FEEDWATER FLOW. Outcome using the same method applied to a MR model is shown in Figure A-85. When averaged over the seasonal group, heat rate estimates using either percent method, instantaneous or relative to a mean, should be essentially the same. Only in extreme cases, such as in highly nonlinear mappings, will the results differ. There does appear to be a limit to the amount the heat rate can change in a season. Using data from Figure A-81 through Figure A-84 the difference between maximum and minimum heat rate estimates for the four seasonal periods results in 41.3, 36.8, 97.4, and 40.0 *Btu/kW-hr*, respectively. The difference between the maximum and minimum non-perturbed heat rate prediction for the same four periods is 35.4, 33.7, 76.7, and 39.7 *Btu/kW-hr*, respectively. The summer and winter seasons show greater variability, but whether or not this implies they are better candidates for optimization is questionable. Average heat rate estimates must be made for all setpoint variables and examined over the four different seasons.

Figure A-86 and Figure A-87 plot heat rate estimates for all 55 setpoint variables using nonlinear and linear mapping models, respectively; here the "instantaneous" values are averaged over the seasonal observation range. A *dead zone* appears in each figure, occurring at the zero perturbation line. The dead zone represents the fit of the predictors to the criterion variable; modeling errors will, of course, change the heat rate prediction value. Depending upon the magnitude of the perturbation, the height above/below the dead zone is different for many of the setpoint variables. A larger height implies the corresponding variable maintains a greater influence, or sensitivity, on heat rate. Table A-15 lists the importance ranking of setpoint variables for fixed perturbations of –0.5% and 0.5%. Variable symmetry is maintained for both the neural network and linear perturbation models. However, individual rankings for many of the variables in the linear model are inconsistent, e.g., FEED PUMP(A/B) SPEED is both important and not important. Red and green lines in Figure A-88 and Figure A-89 indicate averaged heat rate estimates using linear perturbation models (MR and PLS); the blue line shows the heat rate prediction using a nonlinear neural network perturbation model. Both linear models indicate opposite "directionality" for the two feedwater pumps. Directionality is defined as either the positive or negative movement of the heat rate given a predetermined increase/decrease in predictor value, or input perturbation. It might be considered probable that because the observation period does not reflect the sampling frequency of these variables, heat rate estimations are erroneous. This is not assumed to be true. For the winter season, FEED PUMP(A) has a speed of 4,422 RPM while FEED PUMP(B) speed is 4,491 RPM. Both traverse approximately the same input range of ± 10 RPM. It is assumed RPM values are averaged quantities taken over a proper sampling interval. Visual inspection of the data does not indicate why preference is given to one pump over the other when using a linear perturbation model.

Figure A-77 through Figure A-85 showed the heat rate perturbation model is a time varying dynamic system. A time varying dynamic system can be described by differential equations (criterion) which have coefficients that are functions of the independent variables (predictors). The independent variables vary as a function of time. In essence, the AI model implies heat rate is a function of set point variables, each of which in turn is a function of time. There are two interesting questions that need answering. First, is this system as a whole linear or nonlinear? Why this is important is because linear techniques applied to a

nonlinear system may produce erratic and false results. Secondly, does the perturbed system behave in a linear or nonlinear fashion? The answer here has interest in sensitivity studies and in thermal performance optimization.

The daily behavior of the NN perturbation model at large perturbations clearly indicates it is nonlinear. In fact, any system driven hard enough will fail with characteristic nonlinear behavior. However, most large input perturbations are outside the training domain of neural networks. Figure A-90 shows heat rate estimates for several different perturbations of the set point variable TOTAL FEEDWATER FLOW. Heat rate values are averaged over all observations. The two dashed red lines define a confidence interval for heat rate prediction. The lines are determined by computing the difference between the minimum/maximum predictor value and the average predictor value. Input domain values are considered to lie within the NN training region if they are within this confidence interval. Figure A-91 shows an expanded view of the graph points of Figure A-90 within the confidence interval. Other seasonal graphs using the same set point variable are shown in Figure A-92, Figure A-93, Figure A-94, Figure A-95, Figure A-96, and Figure A-97. Figure A-98, Figure A-99, Figure A-100, Figure A-102 show the essentially the same information, however, a constant change in input was used. The different directional changes and wild fluctuations seen at zero perturbation are explained shortly. Daily perturbation plots, as seen in Figure A-103, Figure A-104, Figure A-105, Figure A-106, and Figure A-107 are somewhat misleading because of insufficient data. However, trending of seasonal averages should be fairly accurate.

For small perturbations the predicted seasonal heat rate averaged over all observations usually proceeds in a linear manner. This range varies depending upon the set point variable and magnitude of perturbation. Some exceptions to the linear behavior are seen in Figure A-108 and Figure A-109. Data modeling for the curves was determined using the nonlinear perturbation model. In absence of any perturbations, the small gray diamond indicates the "true" heat rate averaged over time. Because it acts only as a reference or baseline, it is shown the same for all perturbations. At zero perturbation, NN, MR, or PLS predictions may be biased away from the baseline. This indicates modeling error. Biasing error exists for all the predictions. For example, Figure A-96/Figure A-102 appear to have a considerate amount of bias for the

NN prediction. In addition, predictions may also have reversed directionality. The interpretation of opposite directionality between NN and MR estimates is not an easy task.

Typical of nonlinear models, once outside the confidence interval range a number of nonlinear estimates are possible. But because neural networks are data driven, it is expected the directionality of the perturbations be preserved if within this range. Figure A-90 and Figure A-92 indicate proper directionality of the NN heat rate prediction. A corresponding decrease in heat rate should occur for increases in feedwater flow. Both winter and spring linear models indicate otherwise. Summer and fall plots achieve proper directionality for all the model types.

However, there are exceptions to perfect directionality. It's been assumed that using multiple regression or partial least squares techniques on a nonlinear system are the major reason directionality is not preserved. This assumption's validity has been experimentally verified. Yet, nonlinear techniques may also produce false directionality. For example, Figure A-110 shows the summer perturbation plot of the AVG MAIN STEAM PRESSURE. Figure A-111 shows the spring perturbation plot of the same variable. Using data from EPRI's Thermal Performance handbook, the effect of inlet steam pressure produces a heat rate change of approximately +0.15% per 10 psi reduction in inlet pressure. The average pressure reading for the summer and spring are approximately 953.92 and 953.58 psia, respectively. The maximum pressure is 954.74 psia for the summer and 954.97 psia for the spring. Minimum pressures for the summer and spring are 952.42 and 952.12 psia, respectively. Variability of the data is considerable for this predictor, although its range is very small – approximately 3 psia. Nevertheless, directionality is preserved as seen in Figure A-110, i.e., a reduction in pressure shows a corresponding rise in heat rate. If heat rate values within the confidence region are extrapolated for a 10 psi reduction in inlet pressure, the approximate change in heat rate magnitude is also fairly accurate. However, directionality is reversed in Figure A-111. As seen in the daily perturbation plots, see Figure A-112, Figure A-113, Figure A-114, and Figure A-115, there is a considerable amount of modeling error occurring during the spring time period. Different neural network training routines, how well training occurs, cross-validation, etc., all impact model performance. In addition, the shear number of observation points is very important. With fewer points for training the

prediction ability of the model may be only slightly impaired, yet it has a large effect on directionality. Directionality is enhanced when numerous data points are available for training and cross-validation. A good "rule of thumb" is to use more data points. The number of observations per day should be increased to better represent the daily shifts in operational loads.

Known, theoretical directionality of several important plant parameters includes condenser pressure, final feedwater temperature, steam pressure, and feedwater flow. These are reviewed below:

- Condensers operate at a low pressure to enable maximum expansion of the LP turbine exhaust steam. The operating range is at a very low pressure, measured in inches of mercury absolute. An increase in condenser pressure gives rise to a higher condensing temperature, and hence, the greater the amount of heat rejected. The heat rejected is the area under the curve from points 2-3 of Figure 2, this corresponds to Q_{out} *or* $q_{2\text{-}3}$ in Section 1.2.1 of Chapter 1. Note the area beneath the path of a reversible process represents *heat transfer per pound of fluid undergoing that process;* if comparing areas of dissimilar curves care must be taken to account for mass variations between the different paths [27]. This shouldn't be problem, as mentioned earlier, nuclear plants operating under steady-state conditions have essentially equivalent steam and feedwater flows. Since more heat rejected provides less available energy for expansion through the turbine and the heat added is not changed, the subsequent result is an increase in heat rate due to a reduction in thermodynamic efficiency of the cycle. Correspondingly, a lower condenser pressure will decrease the heat rate.

- Feedwater heaters provide a process for transferring heat energy to the water flowing to the steam generator, thereby increasing the thermal efficiency of the Rankine cycle. Steam is extracted from the turbine at various points and pumped to a number of open and closed feedwater heaters. In an open feedwater heater the bled steam and condensate pumped from the condenser are mixed. In a closed feedwater heater the two fluid streams do not mix. The goal is to increase the final feedwater temperature. All of this is accomplished by trying to keep the processes reversible. Because reversible processes of the working fluid provide more work than irreversible processes, a cycle composed

entirely of reversible processes achieves optimal efficiency. Unfortunately the necessity for reversible heat transfer involves small temperature differences. This can be accomplished with large surface areas. Ignoring the trade-off costs between heat exchangers and thermal efficiency, two measures are defined to measure feedwater heater effectiveness, TTD and DCA (see LIST OF ABBREVIATIONS for definitions). Increasing either will increase the heat rate because less heat is transferred to the feedwater and more heat is rejected to the condenser via the cascading drains [9]. Correspondingly, a lower TTD or DCA will decrease the heat rate. The result is a higher final feedwater temperature will lower the heat rate, and vice versa.

- Steam pressure determines the amount of available thermal energy per unit mass for conversion into electrical power by the turbine-generators. Steam pressure is measured in psi or psia. An increase in steam pressure corresponds to a greater amount of heat added. The heat added is the area under the curve from points 4-1 of Figure 2, this corresponds to Q_{in} *or* q_{4-1} in Section 1.2.1 of Chapter 1. More heat added provides more available energy for expansion through the turbine. Because the heat rejected is not increased/decreased, the subsequent result is a decrease in heat rate due to an increase in thermodynamic efficiency of the cycle. Correspondingly, a lower steam pressure will increase the heat rate.

- Feedwater flow is the mass amount of feedwater fluid crossing a boundary of the system per unit of time. It is measured in *klb/hr*. Because the feedwater flow and steam flow is almost identical, the amount of flow work done on/by the system is greater than before when the feedwater flow is increased. An increase in feedwater flow corresponds to more power generated. A subsequent result is a decrease in heat rate due to a higher thermodynamic efficiency of the cycle. Correspondingly, a lower feedwater flow will increase the heat rate.

These four plant parameters define a set of reference directionality's, i.e., given a positive or negative change in magnitude of one variable, the corresponding direction of increase or decrease in heat rate is

assumed to be true. In addition, turbine heat rate is also included as part of the set, directionality is defined such that an increase in turbine heat rate results in a higher heat rate, and vice versa.

Ten of the setpoint predictors with established reference directionalities are listed in Table A-17. Neural networks provided correct directionality for many of the variables. Exceptions to this occurred when a considerable amount of modeling error was present in daily perturbation plots. The reference directionality for the predictor variable TOTAL STEAM FLOW was assumed. Even though modeling error is present when using linear models, multiple regression is a relatively straightforward mathematical technique (see Chapter 3.2). Failure of the linear methods to determine proper directionality during the spring season for setpoint predictor TRB HEAT RATE NET CORR is a gross and significant failure.

Despite the ordinary ability for MR and PLS models to predict heat rate under zero perturbation conditions, severe doubt is cast upon both the MR and PLS perturbation models to indicate proper directionality. In general, most all systems are nonlinear. And while averaged values should not be indicative of daily or weekly tendencies, there is the possibility the system as a whole is operating under nonlinear conditions. Using a linear model may destroy proper directionality -- as seen above. In fact, for a constant perturbation of a set point variable over its entire observation interval, the heat rate prediction remained constant; this was seen with the linear models in the section Set Point Variable Analysis. Nonlinear neural network models with cross-validation provided predictions in heat rate varying as a function of time, which is expected for a time varying nonlinear dynamic system. Destruction of proper directionality occurring under linear MR and PLS perturbation models is puzzling and is recommended as a future topic of research.

6.5 Afterthoughts

Up until now, the issue of optimization of heat rate has been circumvented. It has been demonstrated that neural networks provide the best means of modeling plant or design parameters to heat rate. Even in sensitivity studies, directionality, not magnitude of the heat rate increase, is the most important factor in

perturbation modeling. There are several reasons for this assumption. First, it is absolutely absurd to improve the efficiency of plant performance with a construct if under operating conditions it provides the wrong effect. Just from an economic perspective, this can be very costly. Second, directionality is an indicator of your model. A proper model should capture the directional relationships between setpoint changes and process output responses. While it might appear that data validation of the model prediction to the actual plant output is important, and while it is a concerning factor, *optimization of heat rate can only be explored if the correct directionality is imposed by the predictors.* With this in mind, the experimental results based on nuclear data from the Palo Verde Nuclear Generation Station show a neural network-based perturbation model is most promising for optimization studies.

Unfortunately, optimization of heat rate is not independent of costs or system limitations. A sensitivity analysis of the predictors may yield parameters that are more influential over others. However, a monetary value is associated with the improvement of those parameters, and it might be more cost effective to make improvements to less influential parameters. System limitations normally include material and structural requirements that, if exceeded, result in damage and failure of plant operation. One of the earlier approaches (see Chapter 4) was to study state variables using a two-stage model. The present mapping model problems prevented exploring this more fully. However, it is probable of linking cost information and state variables as additional target outputs to the neural network. A neural network-based perturbation model would then provide not only heat rate information, but also cost and parameter limitations at the same time. This research is left as a possible topic for further investigation.

6.6 Summary

In this chapter several reference directionalities were defined for variables that affect heat rate the most. Reference directionality indicates the direction heat rate should change given a small perturbation in the input variable. What was observed was neural network models provided better agreement with the set of reference directionalities. Both linear models sometimes provided correct results, and at other times failed.

In particular was the significant failure of linear models to provide the correct heat rate direction for an increase in the predictor TRB HEAT RATE NET CORR during the spring season.

Since the concept of linearizing a nonlinear function around an operating point is a well-known technique in engineering, there should be no reason why the directionality from a linear estimation perturbation model is different than the nonlinear type model – unless there are many different operating points. Then it is possible for a number of linear estimators to exist. While individual linear estimators should provide for correct modeling, the average of the estimators over a season might not. A possible explanation considered earlier was modeling error. With non-linearity's present in the system, a nonlinear model has a definite advantage. In any event, comparison of perturbation model directionalities to reference directionalities is a necessary step in identifying proper model choices.

Notwithstanding recommendations made earlier, the major concern is directionality failure of the models. Several reasons are possible, including lack of sufficient data, too large time intervals, weight instabilities, small dynamic range of the inputs, etc. These are explored in the next chapter.

Chapter 7

Quasi-static Assumptions

7.1 Preface

There are two sections in this short chapter. Both attempt to solve the problem of reversed directionalities with linear models. Section 7.2 takes a brief look at regularization. Regularization has been applied in feedwater flow rate evaluation of nuclear power plants, primarily to solve ill-posed problems. Section 7.3, which is the subject of the chapter heading, discusses the assumption of quasi-static behavior. Plots in Section 7.3 use data from *Palo Verde Nuclear* Generation Station, Unit 2. Data was taken every 15 minutes apart during February 3rd, 2000 to February 10th, 2000. In total there was 670 observations.

7.2 Regularization

Figure A-116, Figure A-117, Figure A-118, and Figure A-119 plot seasonal heat rate estimates for the three model types used in Chapter 6. The original data from Chapter 6 was kept. However, another modeling technique implementing ridge regression was included. Ridge regression provides an improvement in directionality for the winter data. Unfortunately, there is no added benefit for spring, summer and fall data plots, except for some changes in magnitude. While helpful, application of ridge regression for perturbation models is not a definitive cure for wrong directionality. It's primarily intended for working with highly collinear data matrices and cases where instabilities of weights occur. Not having enough data for the four seasons may have contributed to some weight unpredictability.

7.3 Derivation and Meaning of Quasi-Static Range of a Variable

The state of a system is specified by the values of its properties. Identical states occur when a system has the same values of its properties at two different times. If no changes in the system state are noticed, the system is in equilibrium, or static condition. A dynamic state would naturally imply a system that is not in

equilibrium. When a system is in equilibrium, and small changes are applied, if equilibrium is allowed to reestablish itself before another change occurs, the process is called a quasi-static process. Since a transformation of a system from one state to another is called a process, this state of the system is commonly refereed to as being quasi-static.

When data is recorded and one attempts to model the state of the system using the data, if the model captures sufficient process information, then the model is a reflection of the system state. As mentioned before, the system state is determined by its properties, however, the model state is determined by the data. Equivalency between the model and system states occurs if, and only if, the model captures system process information. Hence, it would be a severe misjudgment to say the model state is established by the current operating state of the system. In the following theory on quasi-static range, which will be defined shortly, the model state is determined by the data, which then defines the current system state. It takes a reverse approach in that required modeling tools are defined by the data, not whatever the original system condition was in.

The definition of the term *quasi* means: in some sense or degree. Quasi-static, as applied to data, would naturally imply a system that is slowly changing, and hence, the data is slowly varying over time. In many instances, like data in Chapter 4 that was separated at one-day intervals, the quasi-static assumption might be valid. But there is uncertainty in that assumption, keeping this in mind, the following paragraphs presents an overview of fundamental quasi-static concepts.

Assume that several points exist in time, but for now, only two are examined. One point is designated $Y(t_1)$, and the other $Y(t_2)$, where t_1 and t_2 represent a time measurement, with $t_2 > t_1$. Both are criterion points, i.e., they are model targets. Just as one would plot two individual data points, $Y(t_1)$ & $Y(t_2)$ are shown in Figure 6.

The next step is to model three predictors to the criterion points using a single mapping layer. The setup is similar to the one described in Section 6.2, Methodology, on page 95. Predictors are designated A, B, and

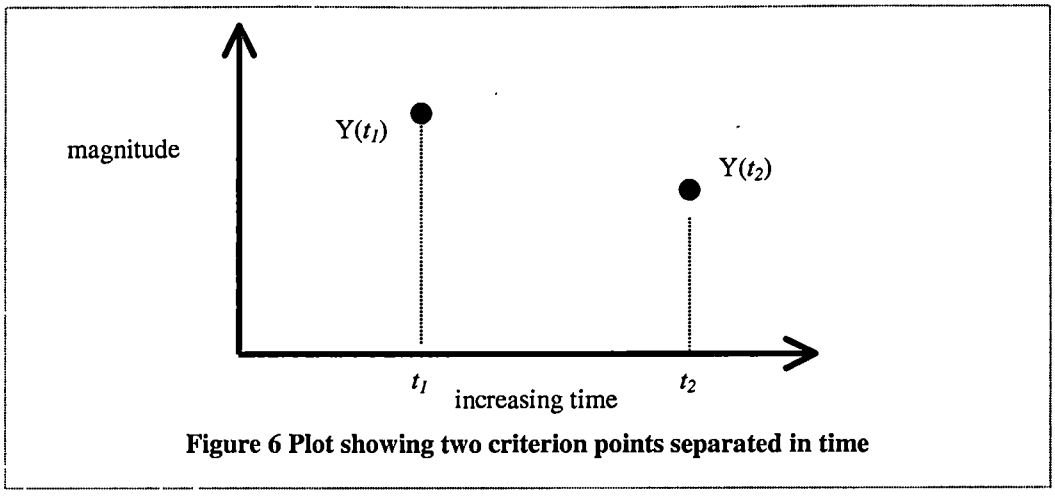

Figure 6 Plot showing two criterion points separated in time

C. The model simply provides a functional mapping of the three predictors (A, B, and C) to the two criterion points, Y(t_1) and Y(t_2).

Assume at an earlier time, t'_C, where $t_1 < t'_C < t_2$, the predictor C(t'_C) changed magnitude. Both A and B were constant. Although C has changed value, there exists a time period, $t_2 - t'_C$, for system equilibrium to occur. This period is defined as the *quasi-static range* (QSR) of predictor C. It is the equilibrium time necessary for predictor C to influence the value of the criterion point at Y(t_2). The quasi-static range may differ for all predictors, and it is especially dependent upon the criterion variable applied in the mapping. If predictors A and B are system process variables, like temperature and pressure, they may also fluctuate in magnitude during C's quasi-static range. Otherwise, A and B will not be affected. Several hypothetical

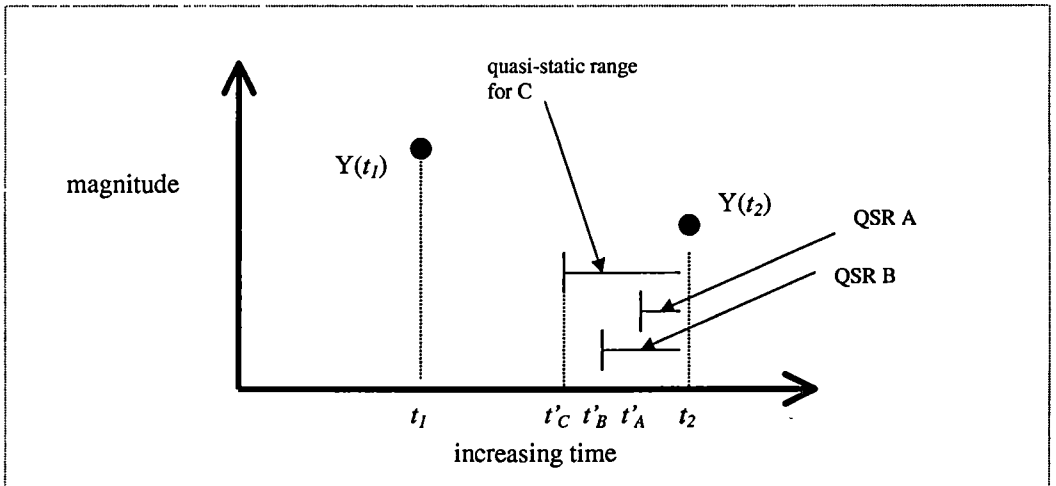

Figure 7 Plot showing quasi-static range of predictors A, B, and C

quasi-static ranges are plotted in Figure 7 for predictors A, B, and C.

The quasi-static range of the predictors is important in determining the state of the system. Normally, systems are classified as dynamic, quasi-static, or static. If the QSR for all predictors' collapses to zero, i.e., the interaction time of predictor influence on the criterion is very small, the problem reduces to a static problem. The data may still vary rapidly between time separations, indicating a dynamic process; however, a zero QSR implies independence between data points. To properly capture the relationships between predictors and criterion nonlinear models might be necessary if the mapping is highly nonlinear; otherwise linear techniques should suffice. When the time separation between points (t_2 - t_1) is greater than the maximum QSR, the problem may be considered quasi-static. If the difference between (t_2 - t_1) and the maximum QSR is negative, the system exhibits dynamic behavior. The former case (static system) may be solved with any number of linear and nonlinear models, while the later (dynamic system) must be solved with temporal models. Using the two points, $Y(t_1)$ and $Y(t_2)$, and QSR's for the predictors, the three system classifications are shown visually in Figure 8.

In prediction-type studies, where only the criterion magnitude is relevant, it is normally safe to use static modeling tools in quasi-static systems. However, this does not apply to perturbation models. The reason is not magnitude related, but dependent upon the predictor QSR. Looking back at the two-point model in Figure 7, if modeling error is sufficiently small then the criterion value at $Y(t_2)$ is a function of all 3

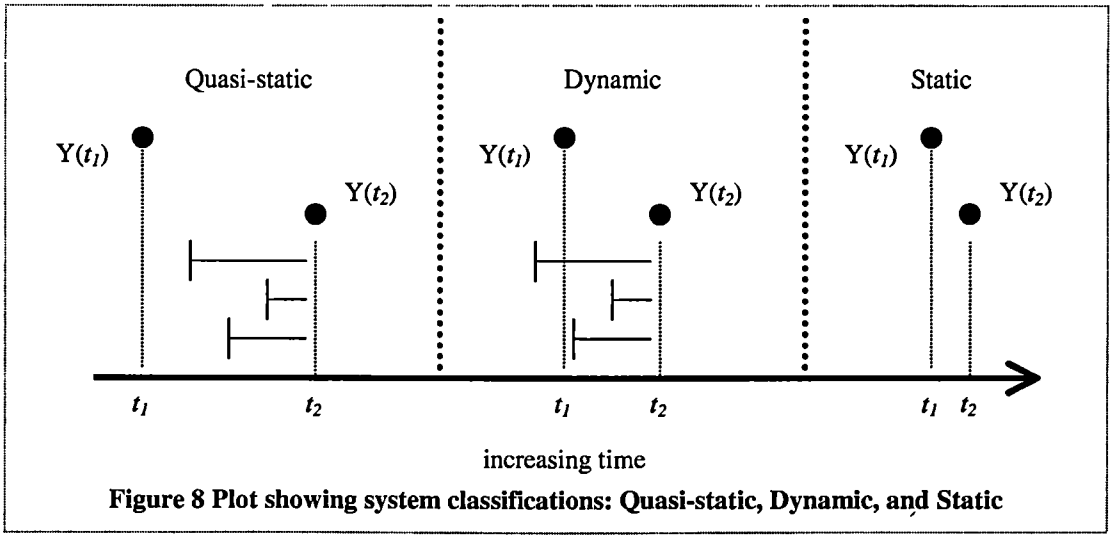

Figure 8 Plot showing system classifications: Quasi-static, Dynamic, and Static

predictors at time t_2. The predictor value at $C(t'_C)$ must be the same at $C(t_2)$ since C's magnitude didn't change. Unfortunately, the modeling process cannot determine if the change in magnitude at $C(t'_C)$ was due to an increase or decrease at time t'_C. The only prior directional information the model has for predictor C is at $Y(t_1)$, and this may or may not be the same – this is especially true if the model bases directionality on correlation of the data points. In ideal cases, the directional information needed by the model is contained at the start of the predictors' QSR, which, for large time separations between data points $(t_2 - t_1)$, is not available.

In situations working with raw data, if a setpoint predictor is kept constant for a considerable time, the directional information might be lost due just to the data sampling interval that is chosen; another ill effect is the QSR of the predictor is extended. There is also the chance data may be taken within the QSR of the predictor, thus providing no directional information. Figure A-118 shows a plot of the variable feedwater flow. Even though the variable magnitude is changing slowly over time, directional changes are occurring quite rapidly. Note: in standardized space the magnitude may even be more significant, since its importance relative to other predictors may be greater than before. Most models filter directional change information by removing so called "noise", working with magnitude information only. What appears to be noise is actually important. Perturbation models are far less concerned with magnitude, since directionality is the major concern. Of utmost importance are perturbation models capturing directional information at the start of the predictors' QSR – not an entire sampling interval before.

Models, like multiple regression, partial least squares, and neural networks, exhibit different characteristics when it comes to determining proper directionality. It was observed in Chapter 6 that linear models sometimes provided correct directionality, and that neural networks were consistently agreeing with the pre-defined reference directionalities. In cases where linear and nonlinear models conflicted on directionality, perturbation magnitude was almost equivalent. This was somewhat expected since all the models use the same data. Using more data should provide for a better prediction and less modeling error; but not necessarily correct directionality. With this in mind, and because feedwater flow has an unusual QSR, it is examined next with data separated at 15-minute intervals.

7.3.1 The QSR of Predictor Feedwater Flow for Turbine Heat Rate

Heat rate was mathematically defined in Section 1.2.3 as

$$heat\,rate = \frac{\dot{m}_{fw}(h_1 - h_4)}{electrical\,output}\left(\frac{Btu}{kW - hr}\right) \qquad (7.3.1\text{-}108)$$

The QSR for feedwater flow mapped to turbine heat rate is considerable. It would appear that an initial perturbation of $\dot{m}_{fw}$ in the numerator of Equation *7.3.1-108* would lead to a direct increase in heat rate -- not so. Reactor power is controlled on the secondary side. Changes in feedwater flow propagate back to the reactor power control with a one-to-one ratio. A 1 % high feedwater flow reading on the secondary side implies plant thermal power on the primary side is 1 % high, resulting in a subsequent increase in *MW* electrical output generated. Plant cycle efficiency is increased and heat rate decreases. This delayed feedback/heating process does take time though, which means the QSR is considerably longer than for most predictors taken from the secondary side.

7.3.2 U2 Data Results

Figure A-119 shows the nonlinear and linear model heat rate estimates for the U2 data. The U2 data contained 670 observations, same 55 setpoints as in Section 6.2, but mapped to turbine heat rate. In the lower right-hand corner of Figure A-119 are standard deviation values for each signal. There is very good agreement between the model predictions and thermodynamic model heat rate values. Figure A-120, Figure A-121, Figure A-122, and Figure A-123 are perturbation plots for 15, 30, and 45-minute time separations between data points.

In Figure A-120 heat rate appears to be increasing for a positive perturbation in feedwater flow. This is a significant error, since it indicates wrong directionally. With the same data interval spacing, application of

ridge regression, see Figure A-121, provides no improvement. Figure A-122 also shows heat rate is increasing for a positive perturbation in feedwater flow. Comparison of the confidence intervals for Figure A-120 and Figure A-122 are different due to the wider interval sampling range. Again, Figure A-123 shows another perturbation plot; however, data was taken every 45 minutes. Directionality is correct for the neural network model; assumptions about the system state being quasi-static must hold for data taken at least every 45 minutes apart. Yet, linear models fail to predict the correct directionality in all plots and the neural network prediction fails when the data is separated 15 and 30 minutes apart. If the QSR for feedwater flow is greater than 30 minutes, this would account for the wrong directionality in Figure A-120 and Figure A-122; the state of the system is dynamic, hence, a temporal model is required. In Figure A-124 and Figure A-125 an Elman neural network was used. The Elman network consisted of a regular inferential network with an added time delay and a few more nodes in the middle layer to compensate for the extra information. Prediction of the NN, MR, and PLS estimates are good, see Figure A-124. Comparison of Figure A-124 with Figure A-119 shows both models provide similar heat rate predictions. However, in Figure A-125 the neural network directionality agrees with the reference directionality. Ordinary MR and PLS linear techniques cannot capture the proper directionality of system process information in either the quasi-static or dynamic state. They fail because directional information needed by the models is contained in the QSR of the predictors, not in the training points.

7.3.3 Consistency of Directionality and Magnitude

To provide a proper perspective of perturbation results, two measures, reliability and uniformity define consistency of directionality and magnitude. Reliability is defined as the extent to which the measuring procedure yields the same results on repeated trials. Uniformity is defined as the consistency of the results obtained under different model configurations.

When using linear models the predicted perturbation magnitude and direction was always reliable. This was expected since nothing changes during the calculation. However, altering the number of variables or observations in the predictor matrix affected directionality, and consequently magnitude. Linear models

provided extremely poor uniformity. The end result was linear models provided highly reliable, but inconsistent, perturbation magnitudes and directions.

On the other hand, neural networks showed good uniformity of directionality provided that a low MSE was obtained. Reliability was generally poor when using a large number of variables, but improved, as the number of variables in the predictor matrix became smaller. Different neural network configurations were very important. The number of nodes in the hidden layer made a significant contribution to both reliability and uniformity. Too few nodes would restrict the solution to a linear estimate, while an abundance of nodes provided a plethora of all types of solutions. A method to force network pruning of nodes is highly desirable. Overall, the proper training of neural networks, when properly structured, ascertains the reliability and uniformity of perturbation magnitudes and directions.

7.4 Summary

The benefits of closely spaced data are twofold. First, data available in the QSR region of the predictors permits the right perturbation models to attain proper directionality. It also allows for increased accuracy of the model prediction and lowers stochastic error. This is especially true if Elman neural networks are exploited. When the time separation is such that the system state is quasi-static, regular inferential neural networks provide good agreement with reference directionalities. However, accounting of correct directionality within the predictors' QSR is believed to be an inherent trait within the model design of the neural network. It is highly recommend that data is spaced at least 10-15 minute intervals and a temporal model is used, such as an Elman neural network. Closer data spacing increases the accuracy of predictor spatial relationships during perturbation, i.e., magnitude of the changes. Multiple regression and partial least squares methods should not be used in perturbation models because they may fail to account for correct directionality.

Chapter 8

Conclusions

8.1 Summation of Main Points

In performing a perturbation or sensitivity analysis, one is primarily interested in the sensitivity coefficient between heat rate and some other quantity, i.e., a quantitative indication of how the heat rate changes with arbitrary changes in some other input variable. The sensitivity coefficient has magnitude and a sign or direction. When an input variable changes, a positive coefficient means that the heat rate changes in the same direction, and a negative coefficient means that it changes in the opposite direction. It is imperative that the directionality of the change be correct, i.e., the change in heat rate obtained from the models due to an input change agrees in direction with the first principles of thermodynamics. While the criterion prediction is still important, it is directionality that determines whether a model is capturing proper changes in system process information. Without the correct direction imposed by the model during input perturbation, heat rate magnitudes are of little concern. This fact was not apparent at first, since most effort in the beginning (Chapters 4 & 5) concentrated on comparing heat rate predictions using the different models, i.e., two linear models (multiple regression and PLS) and a nonlinear model (neural network).

Especially oblivious to the concerns of directionality were the initial steps taken in preprocessing of data. While preprocessing of data is a necessary step for prediction modeling, *in perturbation studies* methods should be limited to standardization of the data and removing collinear variables. Standardized scaling provides equal importance to inputs, which is necessary for comparing perturbation of input variables or examining sensitivity results; it also lessens the number of PLS stages required to attain a satisfactory MSE for the criterion and shortens training times with neural networks. The procedure used in this report for tackling collinear inputs was simply to remove those that had a high-degree of correlation with another input. Other ways to address multicollinearity are to use some type of averaging of the inputs or even possibly applying principal component analysis in finding the first major component or factor analysis to

find a common factor. PCA or factor analysis should not be applied to conditioned data matrices because information is removed -- only if all the inputs are collinear, is PCA or factor analysis recommended

Since all data was initially filtered by way of median filtering of input vectors, data smoothing provided for better modeling and quicker training. However, filtering of data should not be used except to remove outliers. The reason for this was not apparent until understanding QSR theory (Chapter 7), but it is suffice to say for now that filtering changes not only the magnitude of the input, but also changes the system process information the model is trying to capture.

As the work progressed it was obvious something was astray. Change in heat rate for large percent input changes resulted in bizarre neural network predictions. Linear models were not affected, and tended to globally extend heat rate predictions in a linear manner regardless of the magnitude of the input perturbation. While for small perturbations the predicted seasonal heat rate averaged over all observations should proceed in a linear manner, the unbounded linear prediction behavior of multiple regression and PLS models was misleading. Extension of the system model output should approach a nonlinear condition for larger perturbations. Examination of the original and perturbed setpoint signals showed most data were outside the training range of the neural networks. Input perturbations were reduced in magnitude, yet, neural network predictions still showed the heat rate changing as a function of time, whereas linear models provided for a constant change in heat rate. At this point, attempts were made to overtrain the neural networks, add combinations of noise to training and validation vectors, etc. All this was carried out for the simple reason that some experimental results showed using these methods provided for what appeared to be, at times, regimes or areas where heat rate was constant given a constant change in input perturbation. This condition approached that provided by linear models, and it was incorrectly assumed at the time that linear models were yielding correct results and neural networks were not.

In Chapter 6, a new approach was to model heat rate using a simplified, single-stage, direct setpoint-criterion mapping layer. Data analysis continued while keeping in mind the assumption that the heat rate was changing as a function of time. Three-dimensional graphs showed the failure (nonlinear behavior) of

the perturbation models at increased, perturbed-input magnitudes. Comparison of seasonal directionality for the same perturbed inputs for both linear and nonlinear models provided mixed results. Sometimes the two types of model directionalities were congruent, but for some seasons the directionalities were reversed. This was most frustrating, since usually the point of comparing two different types of models is to address statistical significance between the predictions. It was here that the concept of directionality became a more important measure of model fidelity. In fact, determining the proper model directionality led to the development of defining a set of reference directionality's, i.e., given a positive or negative change in magnitude of one variable, the correct direction (in a first principles sense) of increase or decrease in heat rate is assumed to be true. Theoretical directionality of several important nuclear plant parameters were defined for condenser pressure, final feedwater temperature, steam pressure, and feedwater flow. The results from Chapter 6 showed neural network based perturbation models provided better agreement with reference directionalities than did linear models. This was a significant fact, since it implied linear models couldn't capture the correct input-to-output variations when a system changes state.

The shear number of observation points was considered important. With fewer points for training the prediction ability of the model might be only slightly impaired but it might have a large effect on directionality. It was assumed model prediction was enhanced when numerous data points were available for training and cross-validation. Additional data was gathered and the number of observations per day was increased to better represent the daily shifts in operational loads of the nuclear power plant. Chapter 7 examines the closely spaced data. Results showed both neural networks and linear models failed important directionality tests. The reason why both models exhibited wrong directionality is explained by QSR theory (also in Chapter 7), which accounts for equivalency between the model and system states. If one attempts to model the state of the system using the data, the model captures sufficient process information, then the model is a reflection of the system state. The system state is determined by its properties, and the model state is determined by the data. Since directionality is the major concern, it was found that directional information needed to be captured at the start of a predictors' QSR – not an entire sampling interval before. Elman neural networks were necessary for the model to attain proper directionality. The Elman network consisted of a regular inferential network with an added input having time delays.

It was assumed at the start of the project that the secondary-side of a nuclear plant met conditions for a steady-flow system, even though the properties at various points within the system or at the system boundary vary cyclically with time, periodically returning to the same values. The necessity of using temporal information for the model was not expected.

8.2 Overall Conclusion and Recommendations for Future Study

Modeling of the data is governed by quasi-static range theory, which states data must be closely spaced (in time) and prior temporal information is necessary. The conclusion reached is the perturbation model of a nuclear plant is a time-dependent dynamic system, albeit slow varying, and the data show it is also nonlinear.

While individual chapters made specific recommendations for future work relative to the topics presented within, there are at least four important topics that significantly affect the conclusions of this report that should be designated for further research. The first three are achievable using basic engineering or mathematics, while the last is considered more difficult. These are:

1. *Investigation of a predictor's quasi-static range.* The QSR of a predictor is the equilibrium time necessary for the predictor to influence the value of the criterion point (heat rate). It may differ for all predictors, and because it is important in determining not only the state of the system but also the number of time-delays necessary in the perturbation model, it is suggested as a topic for further research.
2. *Optimal time sampling of data.* One of the important conclusions of this report was the need for closely spaced data. However, it is believed data spaced too close together might provide for misleading results. Because the nuclear plant steam cycle is a very large thermodynamic system, perturbations introduced into the system will take a certain amount of time to have an affect upon the system. A time-delayed perturbation model with predictor data separated every second may not

capture important feedback information. Possibly the time spacing taken every minute is all that is necessary. In any event, a more in-depth study of optimal data spacing is suggested as a topic for further investigation.

3. *Number of time-delays.* The optimal number of time delays necessary in the Elman model should be investigated. Models that have a mixed number of predictor QSR equilibrium times need to have several time delays. Especially important is capturing feedback information that originates from the secondary side and traverses back to the primary loop and back again to the secondary side. In this situation a number of time-delays may be necessary, spaced anywhere from one to fifteen minutes or more apart. This research is left as a topic for further investigation.
4. *Stability of nonlinear neural network predictions.* Pure linear estimators were shown to provide a minimum sum-squared error lower than that of nonlinear model predictions. However, it was concluded linear models were not ideally suited to perturbation analysis; indeed, they were not valid because they failed many directionality tests. Unfortunately, in a nonlinear neural network the model configuration might also provide false outcomes with minimum squared-error but incorrect directionality. Naturally this will effect the consistency of the neural network to produce answers with proper directionality. It is not known at this time if data limitations, such as over sampling or under sampling, impose additional restrictions on neural network stability. In fact, many of the items discussed above may also have an impact. This research is left as a topic for further investigation.

It should be noted many items, including cost estimations and system limitations, need to be examined. However, the ability of the model to properly capture system process information is of utmost importance, and hopefully, research into the above future recommendations will provide better insight into how to achieve a better perturbation model representation of the nuclear plant steam cycle.

References

References

1. Agogino, A.M.; Kim, Y.J.; Sopocy, D.M.; Henry, R.E.; Frogner, B.; Blanco, M.A.; Divakaruni, S.M.; *Development of an On-line Expert System*: HEATXPRT (Heat Rate Degradation Expert System Advisor), Proceedings: 1989 Conference on Advanced Computer Technology for the Power Industry. Volume 1, Computer Technology, Publication Date: Jul 1990 p11.1-11.3.

2. Angelo, Joseph A., Jr., Ph.D. and David Buden, SPACE NUCLEAR POWER, Orbit Book Company, Inc., Malabar, Florida, © 1985.

3. Beale, Mark, and Demuth, Howard, Neural Network Toolbox User's Guide, The Mathworks, Inc., Natick, MA, © 1992.

4. Black, Christopher L., Artificial Neural Network Sensitivity Analysis and Process Modeling with Application to Kingston Fossil Plant Net Unit Heat Rate Optimization, Thesis, University of Tennessee, Knoxville, 1994.

5. Cooley, William W. and Lohnes, Paul R., Multivariate Analysis, John Wiley & Sons, Inc., New York, NY, © 1971.

6. Cooper, George R., and McGillern, Clare D., Probabilistic Methods of Signal and System Analysis, 2nd Edition, Oxford University Press, New York, NY, © 1986.

7. Cureton, Edward E., and D'Agostino, Ralph B., FACTOR ANALYSIS, An Applied Approach, Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, © 1983.

8. Dunteman, George H., Introduction to Multivariate Analysis, Sage Publications, Beverly Hills, CA, © 1984.

9. EPRI, Thermal Performance Engineer's Handbook, Volume I, Introduction to Thermal Performance, TR-107422-V1, EPRI, Palo Alto, California, March 1998.

10. EPRI, Thermal Performance Engineering Handbook, Volume II, Advanced Concepts in Thermal Performance, TR-107422-V2, EPRI, Palo Alto, California, October 1998.

11. *Factor Analysis*, Tripos Inc., © 1998.

12. Fantoni, P.F., Racz, A. and Figedy, S., *PEANO, a Toolbox for Real-Time Process Signal Validation and Estimation*, HWR-515, OECD Halden Reactor Project, Halden, 1998.

13. Fowler, N.M.; Klett, D.E.; Moreno, J.B.; Heerman, P.D.; *Using Artificial Neural Networks to Predict the Performance of a Liquid Sodium Reflux Pool Boiler Solar Receiver*, Sandia National Labs., Albuquerque, NM, Report Number(s): SAND-97-0366C, CONF-970441-2; Publication Date: 1997.

14. Frogner, B.; Dourandish, R.; *Application of Knowledge Engineering to Power Plant Heat Rate* Performance Modeling, American Society of Mechanical Engineers, Report Number(s): CONF-861211, Publication Date: 1987 p.8.

15. Geladi, Paul, and Kowalski, Bruce R., *Partial Least-Squares Regression: A Tutorial, Analytica Chimica Acta*, 185 (1986) 1-17, © 1986 by Elsevier Science Publishers B.V.

16. Golub, Gene H., and Van Loan, Charles F., Matrix Computations THIRD EDITION, The Johns Hopkins University Press, Baltimore, Maryland, © 1996.

17. Gorzelnik, E.F.; *Boiler Optimization – Utilities Look for ways to Run Boilers Smarter*, Electr. World (United States) v 199:11, Publication Date: Nov 1985 p49-56.

18. Gorsuch, Richard L., Factor Analysis, 2nd Edition, Laurence Erlbaum Associates, Publishers, Hillsdate, New Jersey, © 1983.

19. Green, Paul E.; Carrol, J. Douglas, MATHEMATICAL TOOLS FOR APPLIED MULTIVARIATE ANALYSIS, Academic Press, New York, NY, © 1976.

20. Gribok, Andrei V.; Attieh, Ibrahim; Hines, J. Wesley; Uhrig, Robert E.; *REGULARIZATION OF FEEDWATER FLOW RATE EVALUATION FOR VENTURI METER FOULING PROBLEM IN NUCLEAR POWER PLANTS*, Ninth International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-9), San Francisco, California, October 3-8, 1999.

21. Guo, Zhichao, Nuclear Power Plant Fault Diagnostics and Thermal Performance Studies Using Neural Networks and Genetic Algorithms, Thesis, University of Tennessee, Knoxville, 1992.

22. Guo., Z.; Uhrig, R.E.; *Nuclear Power Plant Performance Study Using Neural Networks*, Conference Title: 1991 Institute of Electrical and Electronic Engineers (IEEE) Nuclear Science Symposium and Medical Imaging Conference, Report Number(s): CONF-911106-98, Publication Date: 1991.

23. Harman, Harry H., Modern Factor Analysis, Third Edition Revised, The University of Chicago Press, Chicago, Illinois, © 1976.

24. Höskuldsson, Agnar, PLS Regression Methods, Journal of Chemometrics, Vol. 2, 211-228 (1988), © 1988 by John Wiley & Sons, Ltd.

25. Höskuldsson, Agnar, *Experimental Design and Priority PLS Regression*, Journal of Chemometrics, Vol. 10, 637-668 (1996), © 1996 by John Wiley & Sons, Ltd.

26. Iida, Osamu; Ushijima, Yuichi; Sawada, Toshiro; *Application of AI Techniques to Blast Furnace Operations*, Iron and Steel Engineer v 72:10.; Publication Date: Oct 1995 p24-28.

27. Jones, James B., Hawkins, George A., ENGINEERING THERMODYNAMICS, *An Introductory Textbook*, John Wiley & Sons, New York, N.Y., © 1960.

28. Kavaklioglu, K.; Uhrig, R.E.; Upadhyaya, B.R.; *Performance Monitoring And Control Design Of Nuclear Power Plants*, Final Report-Volume 1, TVA-84395V, prepared for TVA Research Group, Chattanooga, Tennessee, September 1994.

29. Kavaklioglu, K.; Upadhyaya, B.R.; *Monitoring Feedwater Flow Rate and Component Thermal Performance of Pressurized Water Reactors by Means of Artificial Neural Networks*, Nuclear Technology v 107:1; Publication Date: Jul, 1994 p 112-123.

30. Lindgren, Fredrik, and Geladi, Paul, and Rännar, Stefan, and Wold, Svante, Interactive Variable Selection (IVS) for PLS. Part 1: Theory and Algorithms, Journal of Chemometrics, Vol. 8, 349-363, © 1994 by John Wiley & Sons, Ltd.

31. Maines, P.; Williams, S. Levy, E.; *Application of Boiler Op for Combustion Optimization at PEPCO*, Proceedings of the American Power Conference: Volume 59-1, Publication Date: 1997 p336-339.

32. Mochizuki, Y.; *Way to Save Energy Recommended by Group Work*, Nippon Enerugi Gakkaishi (Journal of the Japan Institute of Energy) v 72:6, Publication Date: 20 Jun 1993 p493-495.

33. Mol, Ir., J., Factor Analysis in Research, University of Groningen, © 1980.

34. Munoz, A.; Villar J.; Sanz-Bobi, M.A.; *Neural Network Approach to the Diagnosis of the Boiler Combustion in a Coal Power Plant*, Power-Gen Europe '95. Conference papers. Book III. Volumes 6-7; Publication Date: 1995 p379-402.

35. Qin, S.J., and McAvoy, T. J., *Nonlinear PLS Modeling using Neural Networks, Computers chem. Engng*, Vol. 16, No. 4. pp. 379-391, 1992, © 1992 by Pergamon Press Ltd.

36. Rännar, Stefan, and Lindgren, Fredrik, and Geladi, Paul, and Wold, Svante, *A PLS Kernel Algorithm for Data Sets With Many Variables and Fewer Objects. Part 1: Theory and Algorithm*, Journal of Chemometrics, Vol. 8, 111-125, © 1994 by John Wiley & Sons, Ltd.

37. Sherin, J. R., *Neural Network Techniques Applied to Thermal Performance Monitoring, Reactor Performance Monitoring Section*, Corporate Nuclear Engineering Division, Ontario Hydro Nuclear, Toronto, Ontario (1997-1998).

38. Smilde, Age K., COMMENTS ON MULTILINEAR PLS, Journal of Chemometrics, Vol. 11, 367-377 (1997).

39. Sreedhar, Rajiv, Fault Diagnosis and Control of a Thermal Power Plant (Neural Networks, Artificial Intelligence), The University of Texas at Austin, 1995.

40. Stark, Peter A., Introduction to Numerical Methods, The Macmillan Company, New York, NY, © 1970.

41. Stuart, Alan, The Idea's of Sampling, MacMillian Publishing Co., NY, NY, © Charles Griffin & Co., Ltd., 1984.

42. Sunde, Svein, and Berg, Øivind, *Status for the development of an optimisation system for thermal-performance monitoring*, OECD Halden Reactor Project, Session C2, Paper No. 7, May-1999.

43. The Nuclear Professional, Volume 14, Number 4, Fourth Quarter 1999, Atlanta, GA.

44. Thorndike, Robert M., Correlational Procedures for Research, Gardner Press, Inc., New York, NY, © 1978.

45. Tipler, Paul A., Physics, 2nd Edition, Worth Publishers, Inc., New York, NY, ., © 1982

46. Tsoukalas, Lefteri H., Uhrig, Robert E., Fuzzy *and* Neural Approaches *in* Engineering, John Wiley & Sons, Inc., New York, NY, © 1997.

47. Upadhyaya, B.R., Class Notes NE 522, Data Sampling, Page 2, Spring 1997.

48. Upadhyaya, B.R., Uhrig, R.E., etalia, Performance Monitoring and Control of Nuclear Power Plants, Final Report – Volume 1, University of Tennessee, Knoxville, TN, September, 1994.

49. Vigneau, E., and Devaux, M. F., and Qannari, E. M., and Robert, P., PRINCIPAL COMPONENT REGRESSION, RIDGE REGRESSION AND RIDGE PRINCIPAL COMPONENT REGRESSION IN SPECTROSCOPY CALIBRATION, Journal of Chemometrics, Vol. 11, 239-249 (1997).

50. Wonnacott, Ronald J., and Wonnacott, Thomas, H., Econometrics, John Wiley & Sons, Inc., New York, NY, © 1970.

51. Wood, C. E. , *Thermal Performance Program Papers*, Tennessee Valley Authority, Watts Bar Nuclear Plant, 1998.

52. Wrest, Darryl J., Instrument Surveillance and Calibration Verification Through Plant Wide Monitoring Using Autoassociative Neural Networks, a thesis presented for the Master of Science Degree, The University of Tennessee, Knoxville, August, 1996.

53. Xu, Xiao, Hines, J. Wesley, Uhrig, Robert E., Sensor Validation and Instrument Calibration Monitoring – Phase One Final Report, Maintenance and Reliability Center, University of Tennessee, December, 1999.

Appendix

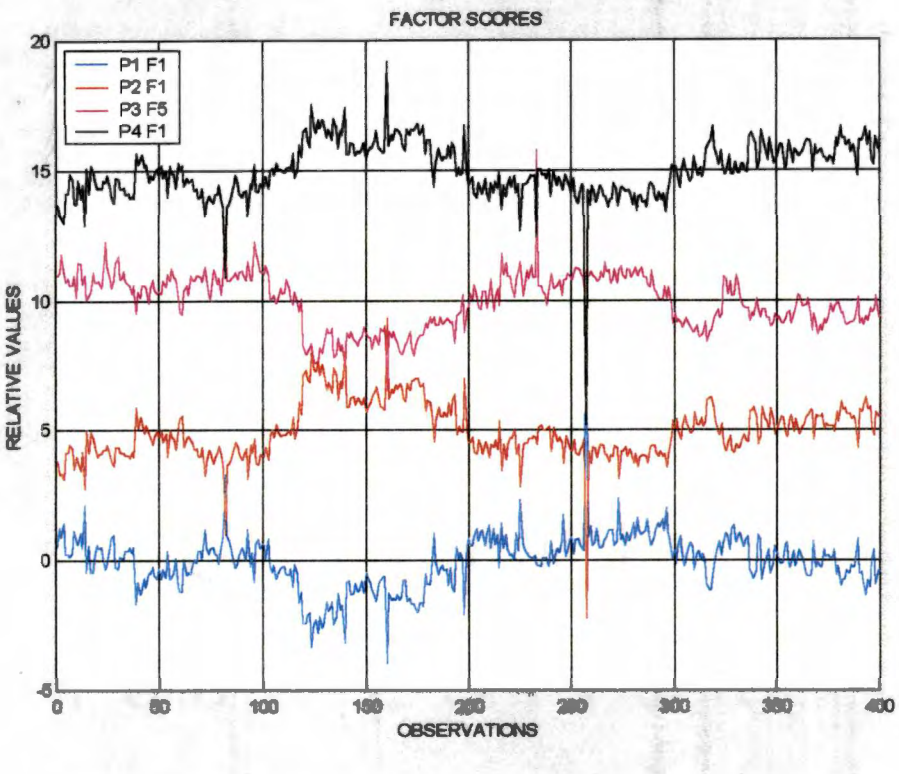

Figure A-1 Factor scores obtained from four different pattern matrices

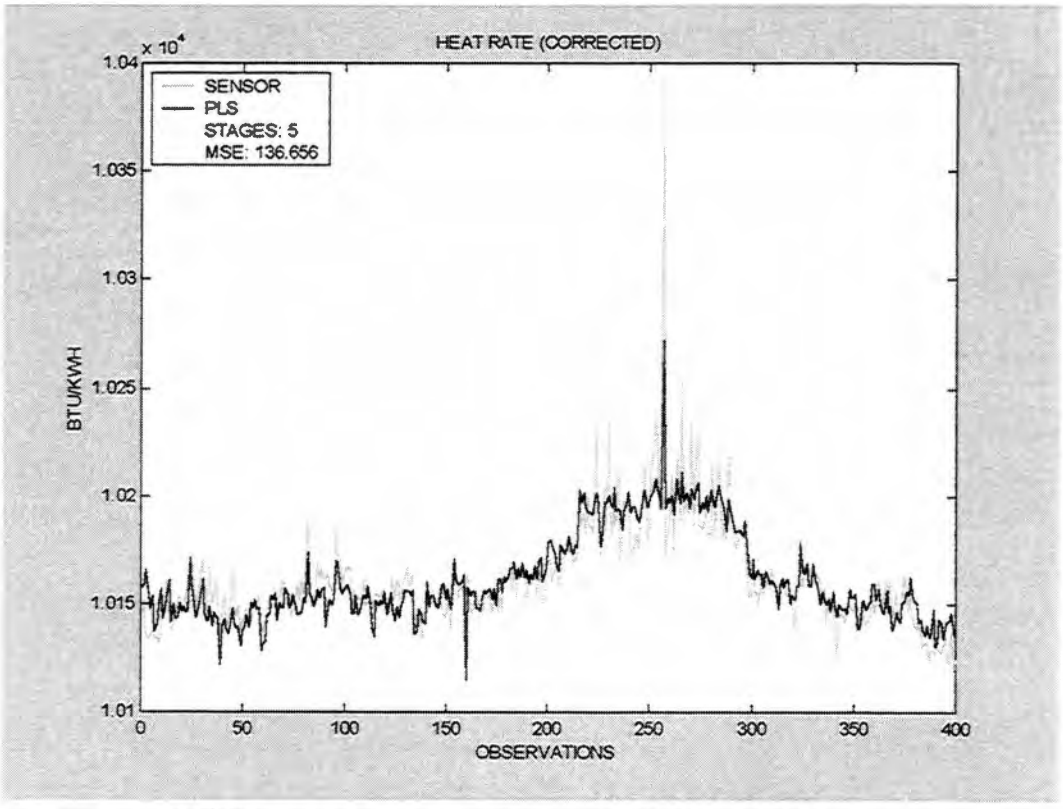

Figure A-2 PLS estimate, 5 stages, predictors are 43 state variables, no pre-scaling

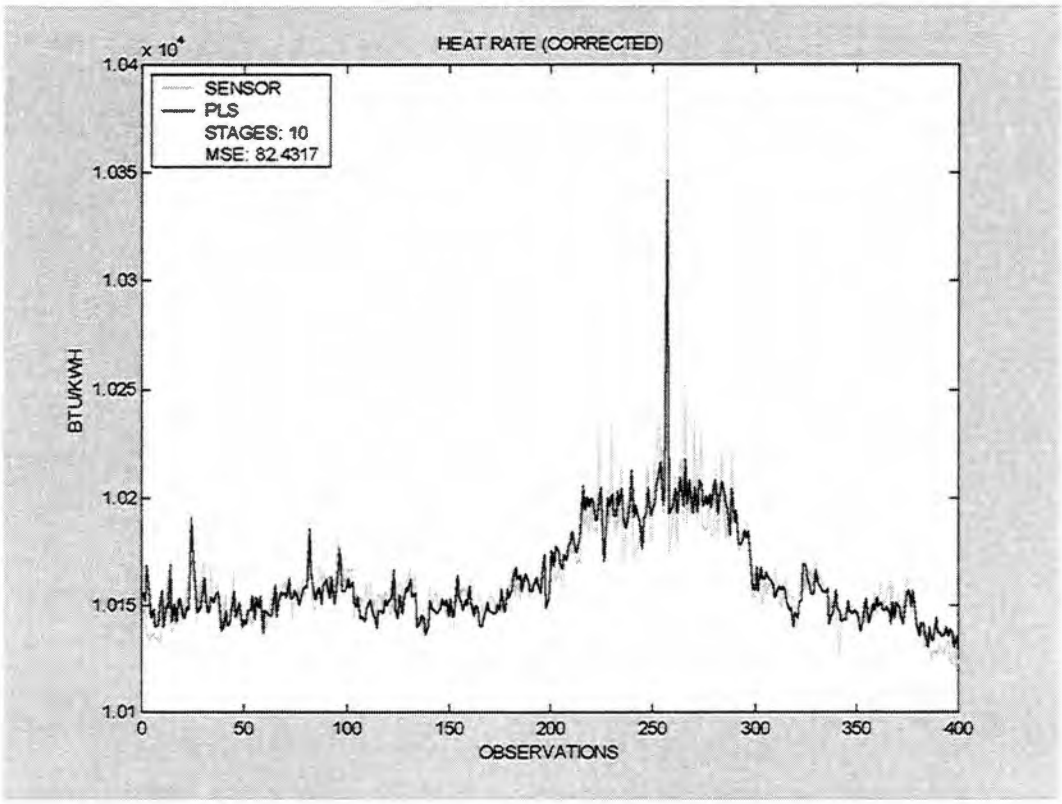

Figure A-3 PLS estimate, 10 stages, predictors are 43 state variables, no pre-scaling

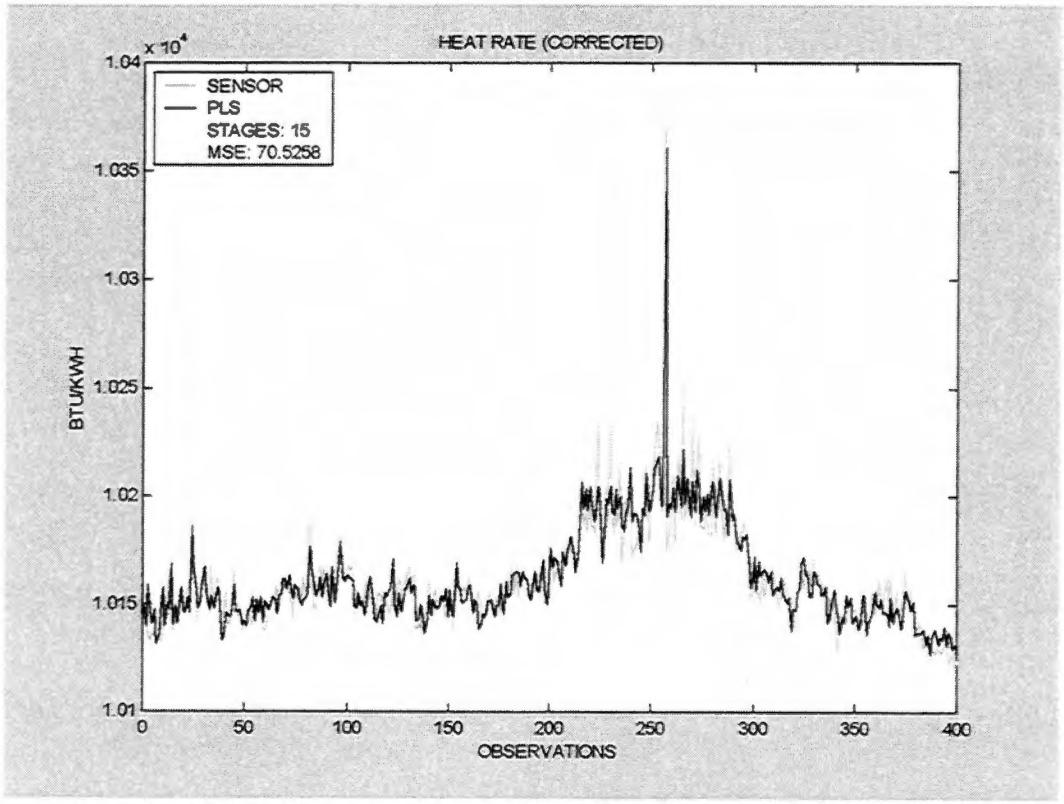

Figure A-4 PLS estimate, 15 stages, predictors are 43 state variables, no pre-scaling

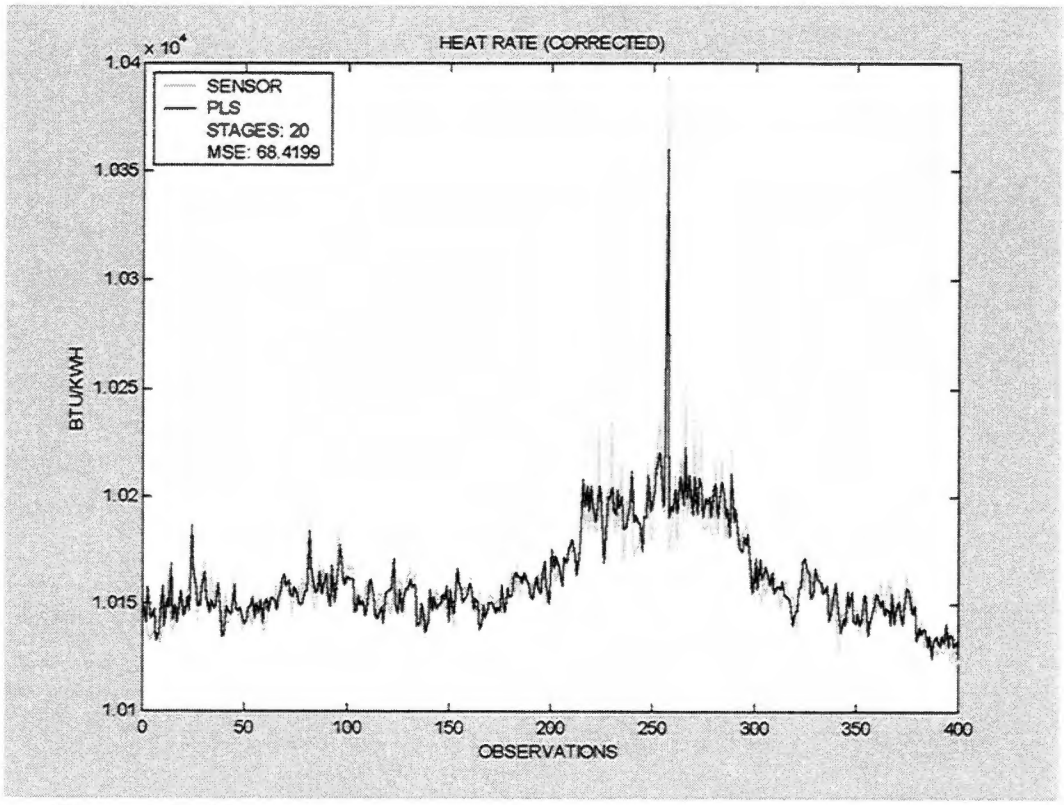

Figure A-5 PLS estimate, 20 stages, predictors are 43 state variables, no pre-scaling

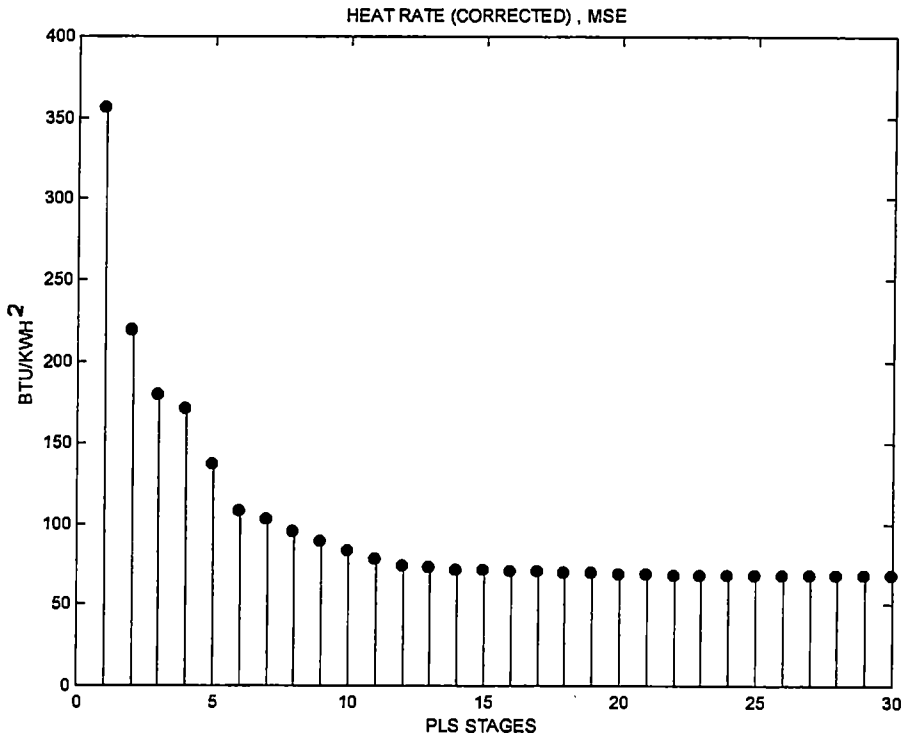

Figure A-6 MSE versus # PLS stages, predictors are 43 state variables, no pre-scaling

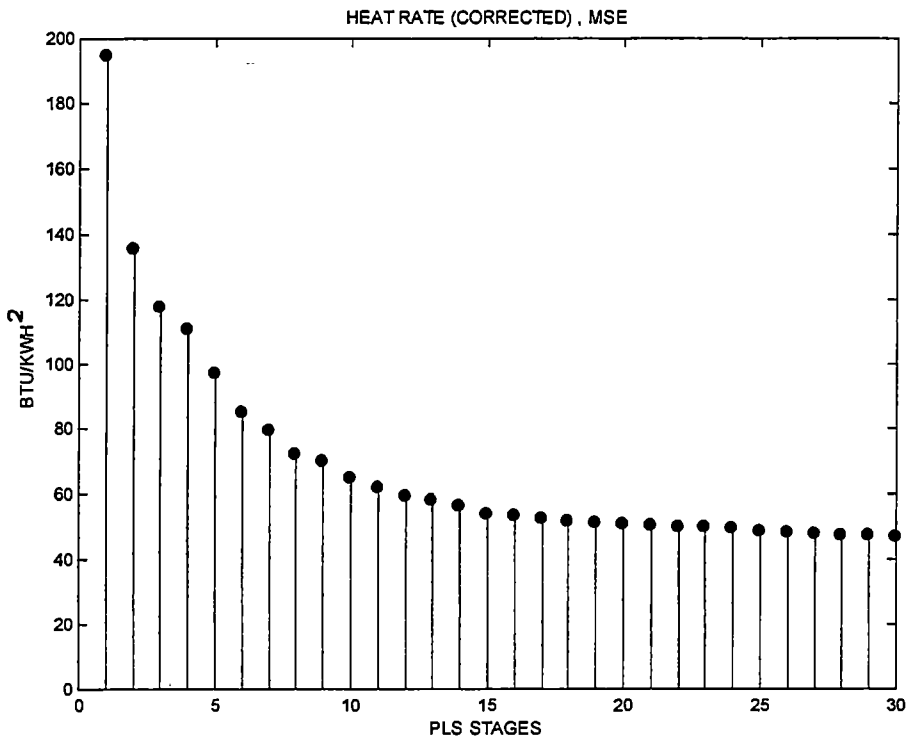

Figure A-7 MSE versus # PLS stages, predictors are 43 state variables, SD pre-scaling

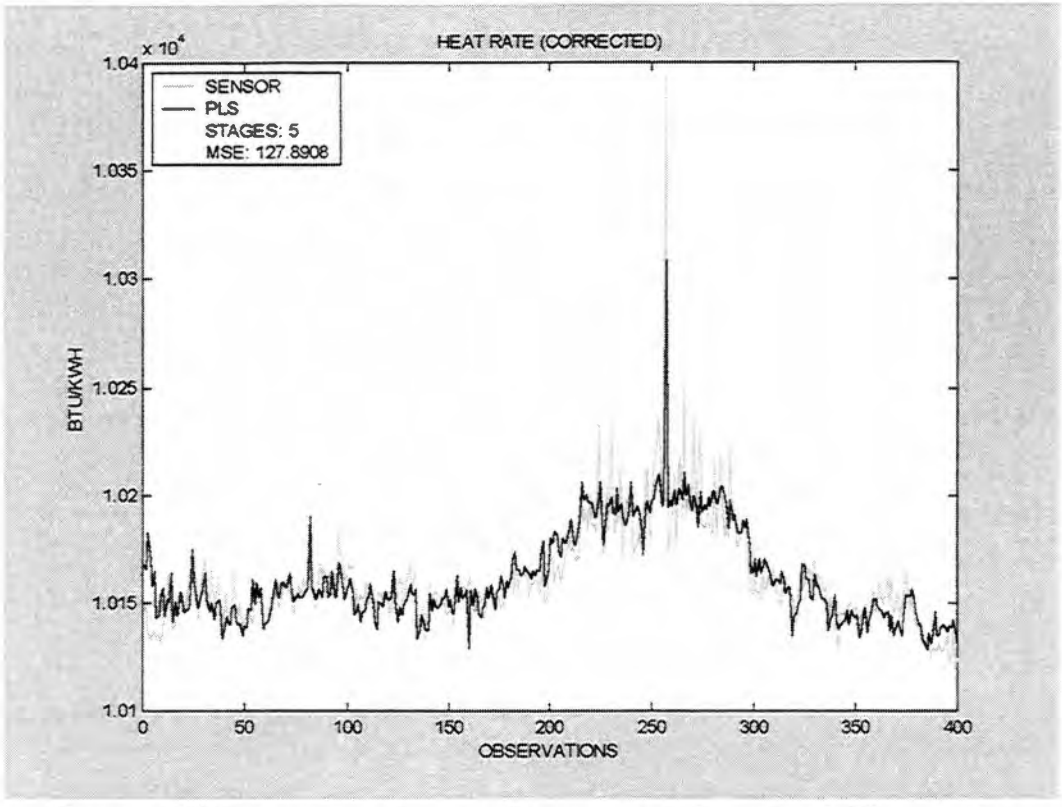

Figure A-8 PLS estimate, predictors are factor scores (5 dimensional), SD pre-scaling

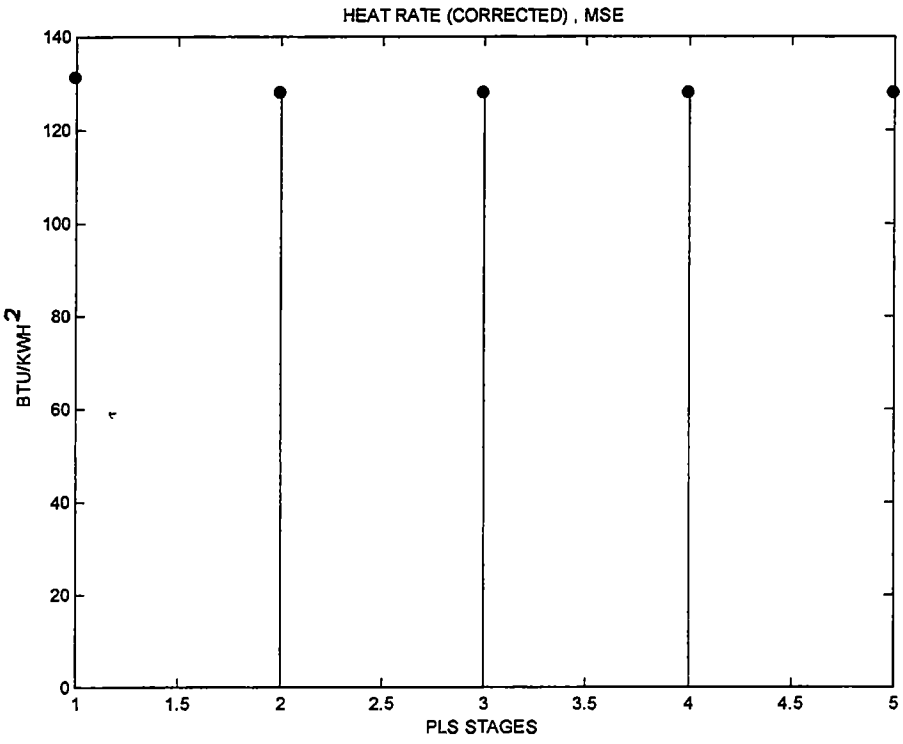

Figure A-9 MSE versus # PLS stages, predictors are factor scores (5 dimensional), SD pre-scaling

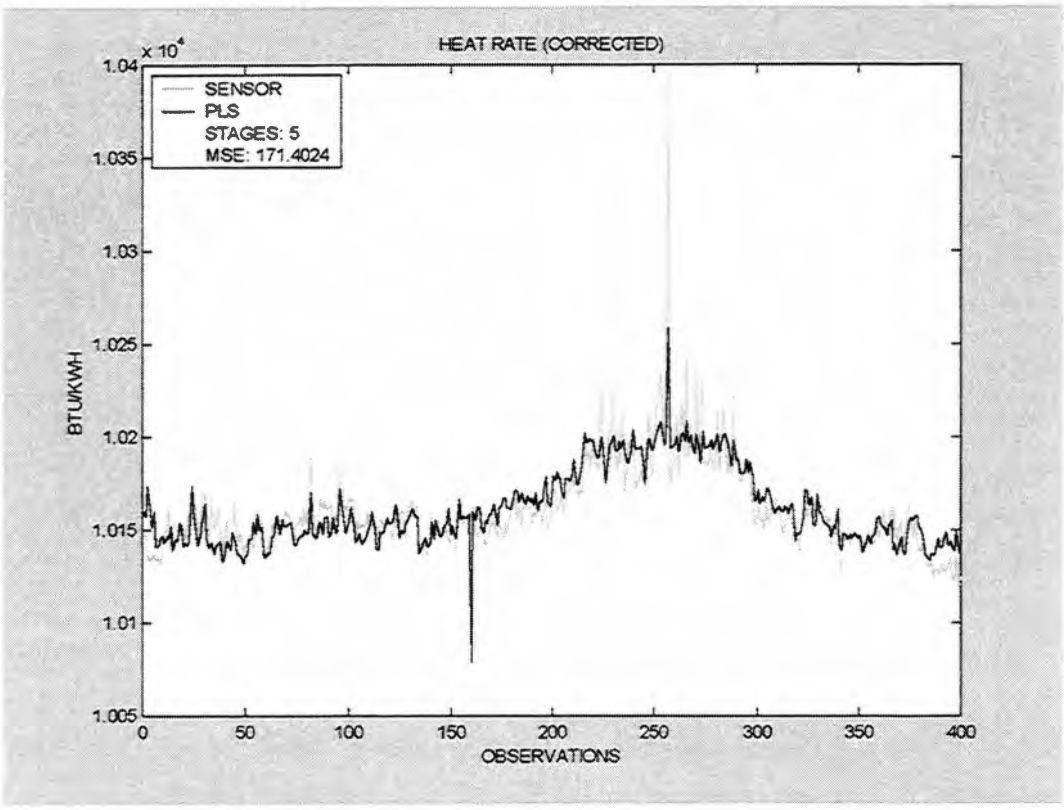

Figure A-10 PLS estimate, predictors are principal component scores (5 dimensional), SD pre-scaling

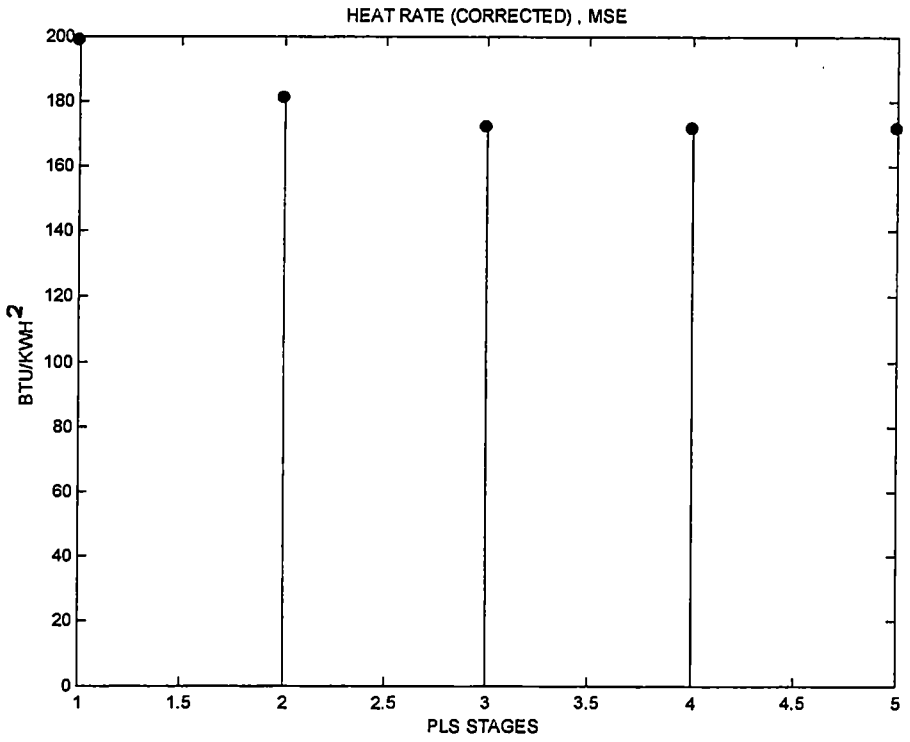

Figure A-11 MSE versus # PLS stages, predictors are principal component scores (5 dimensional), SD pre-scaling

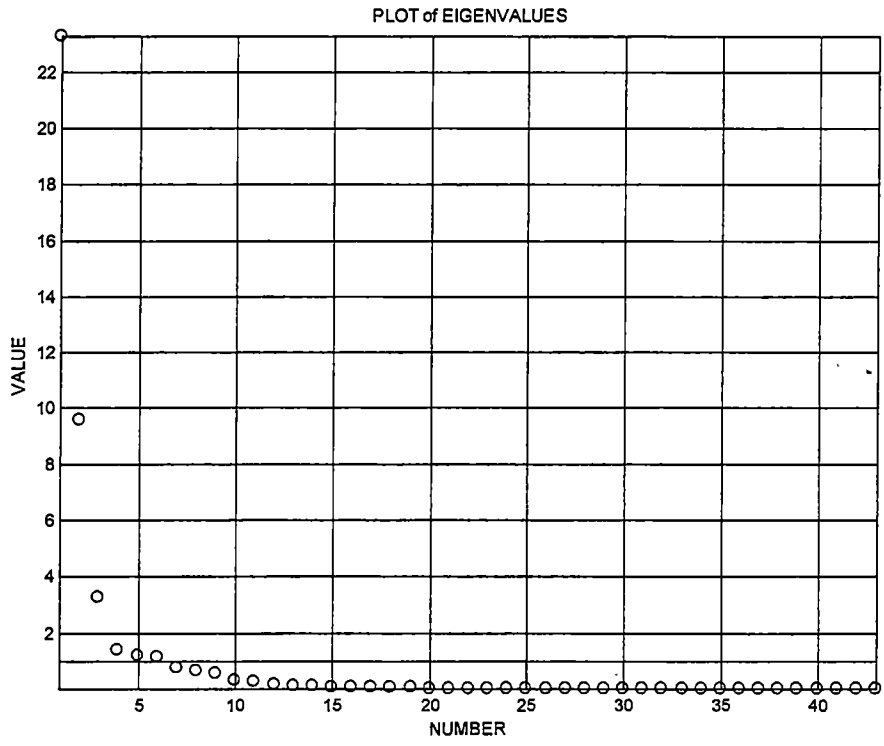

Figure A-12 Eigenvalues for 43 state variables

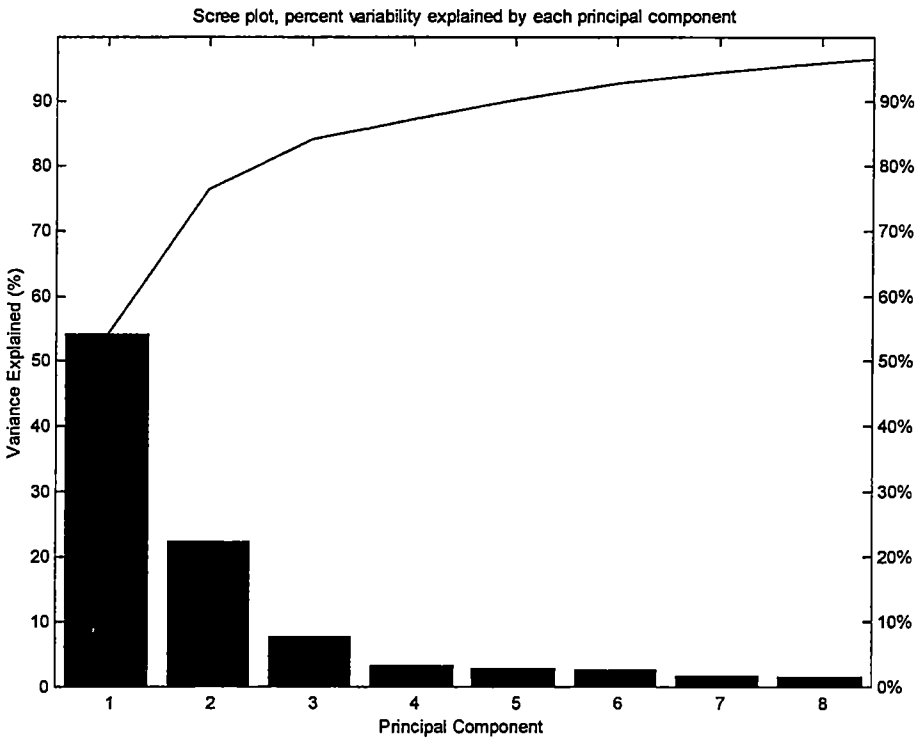

Figure A-13 Scree plot for principal components

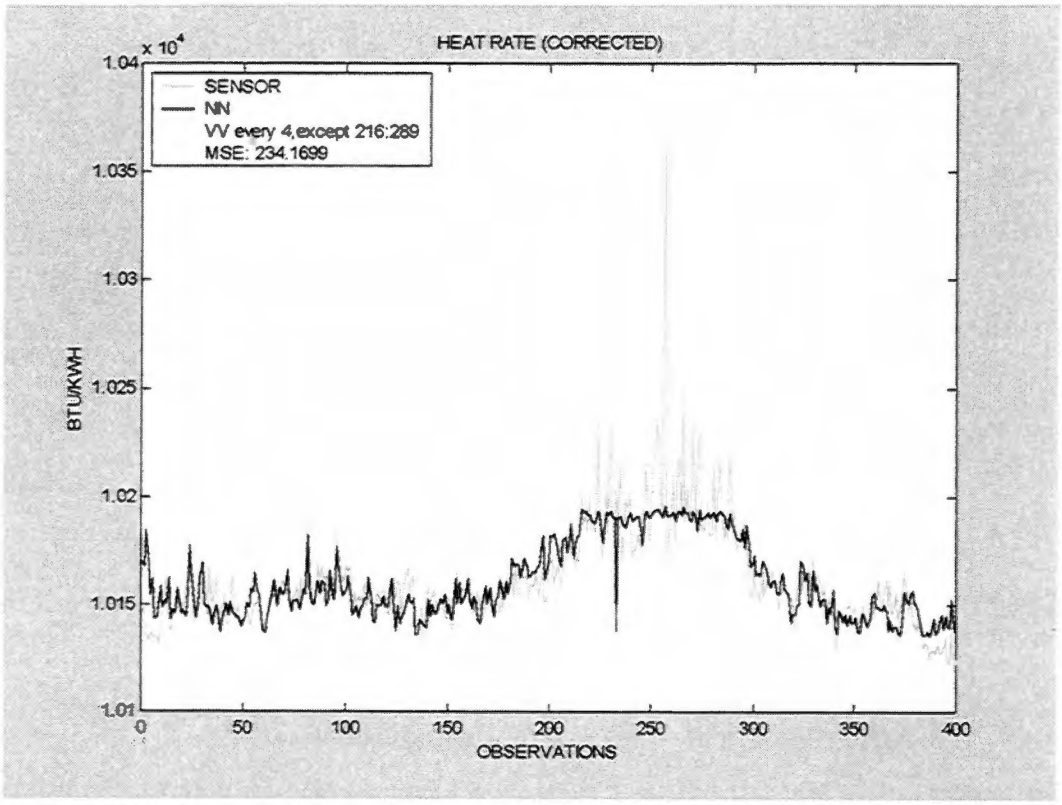

Figure A-14 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 2 hidden nodes (tansig) and linear output layer [VV every 4, except 216:289]

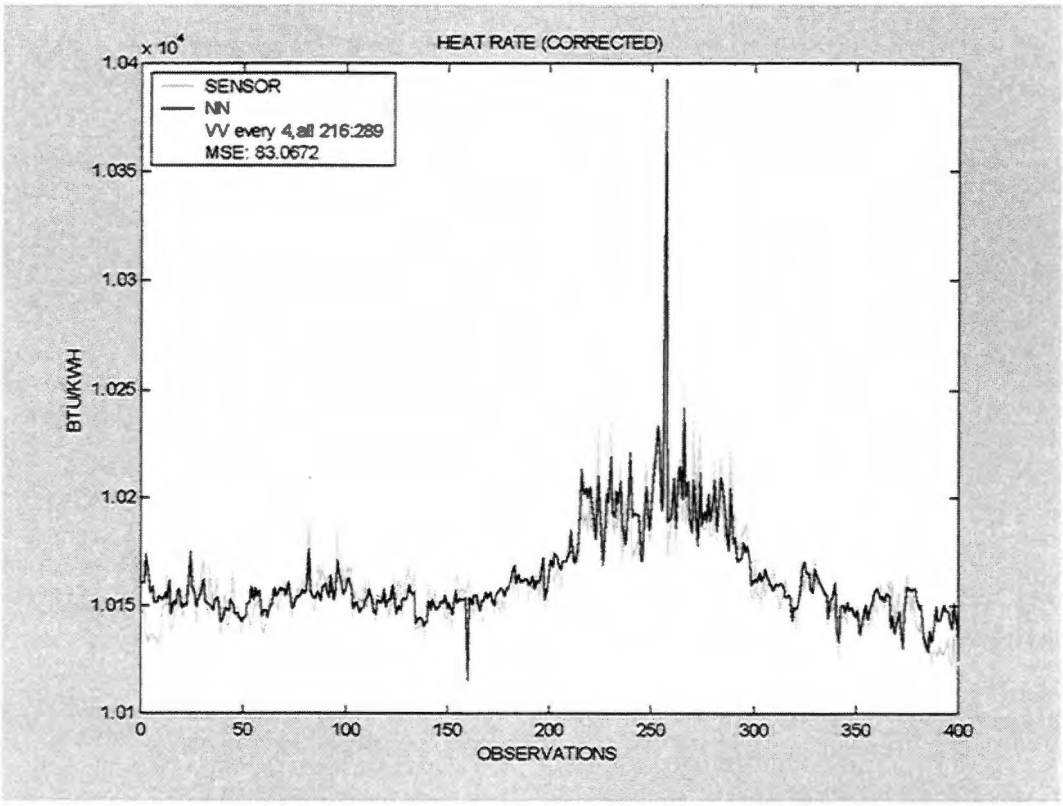

Figure A-15 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 2 hidden nodes (tansig) and linear output layer, [VV every 4, all 216:289]

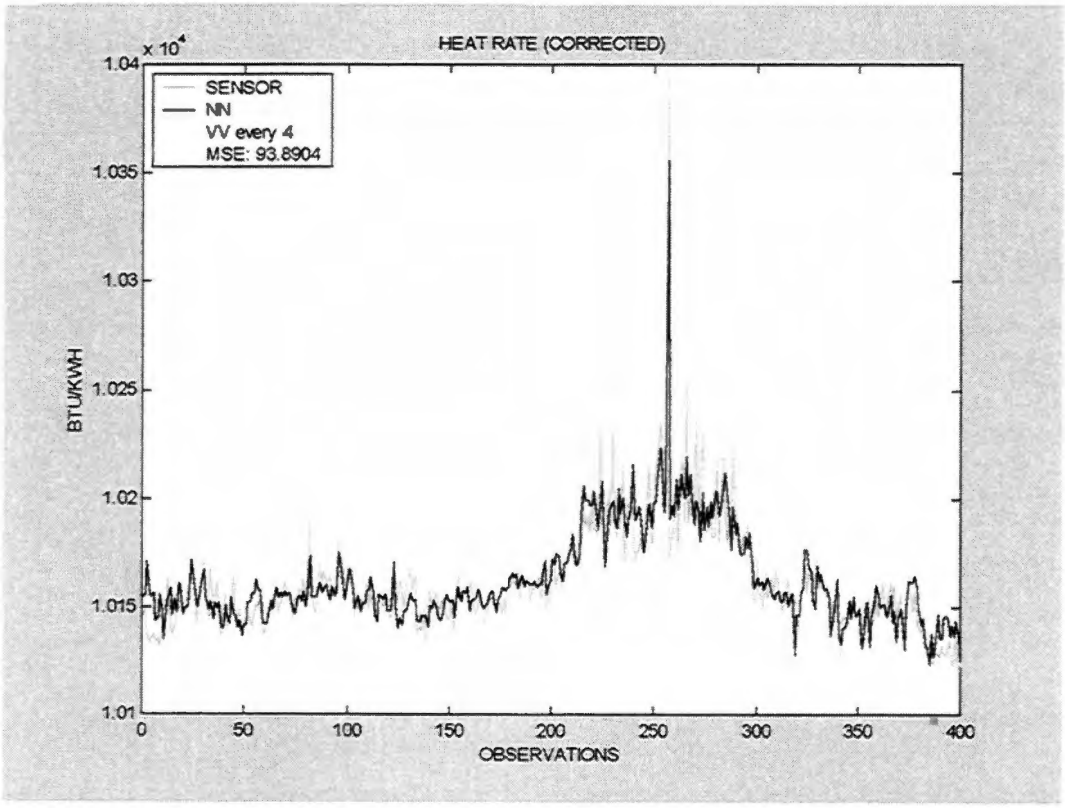

Figure A-16 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 2 hidden nodes (tansig) and linear output layer, [VV every 4]

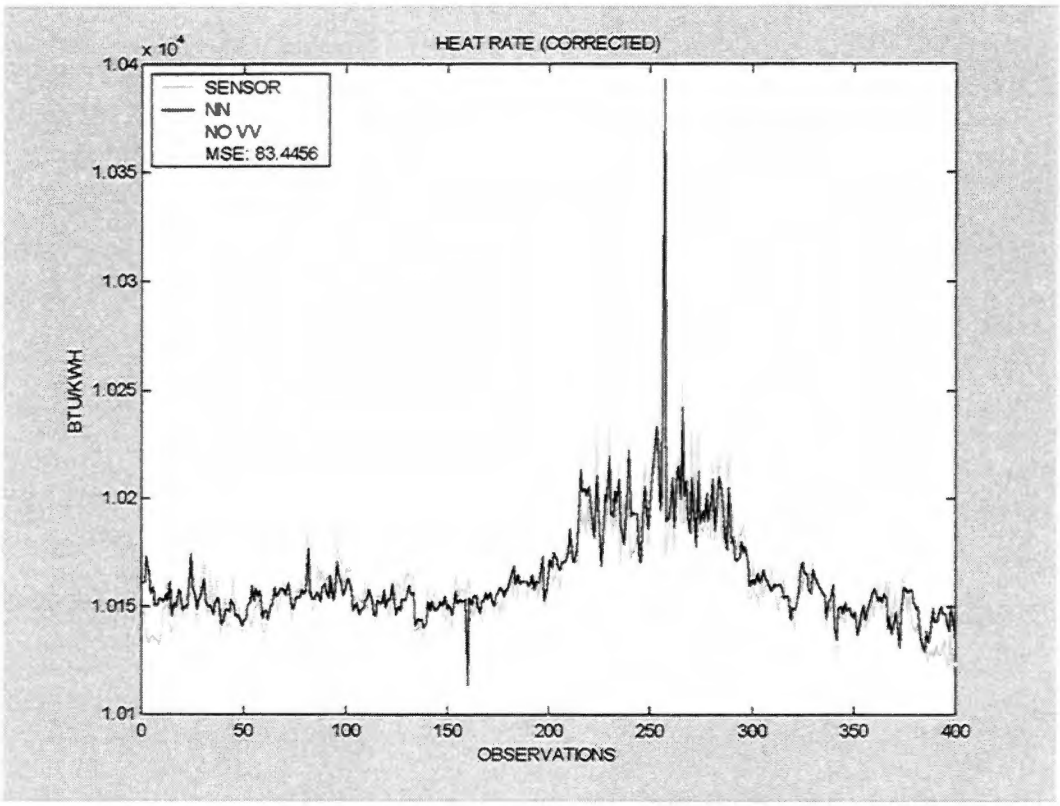

Figure A-17 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 2 hidden nodes (tansig) and linear output layer, [No VV]

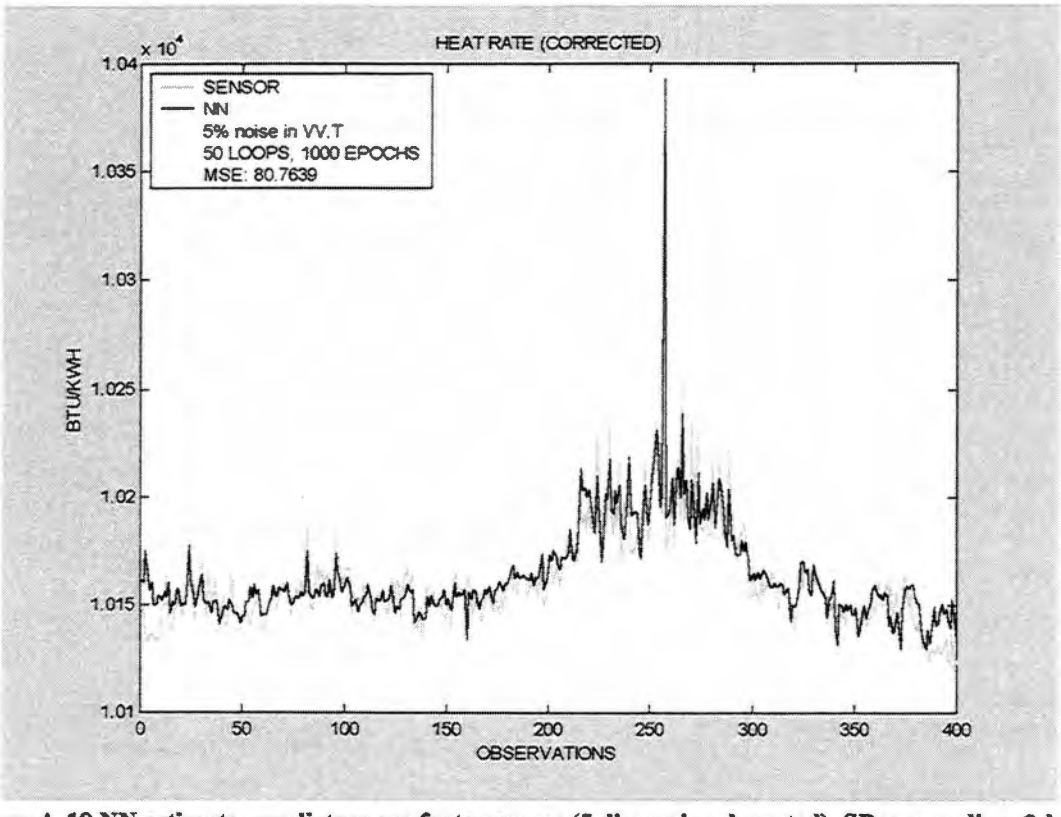

Figure A-18 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 2 hidden nodes (tansig) and linear output layer, [5% noise in VV.T]

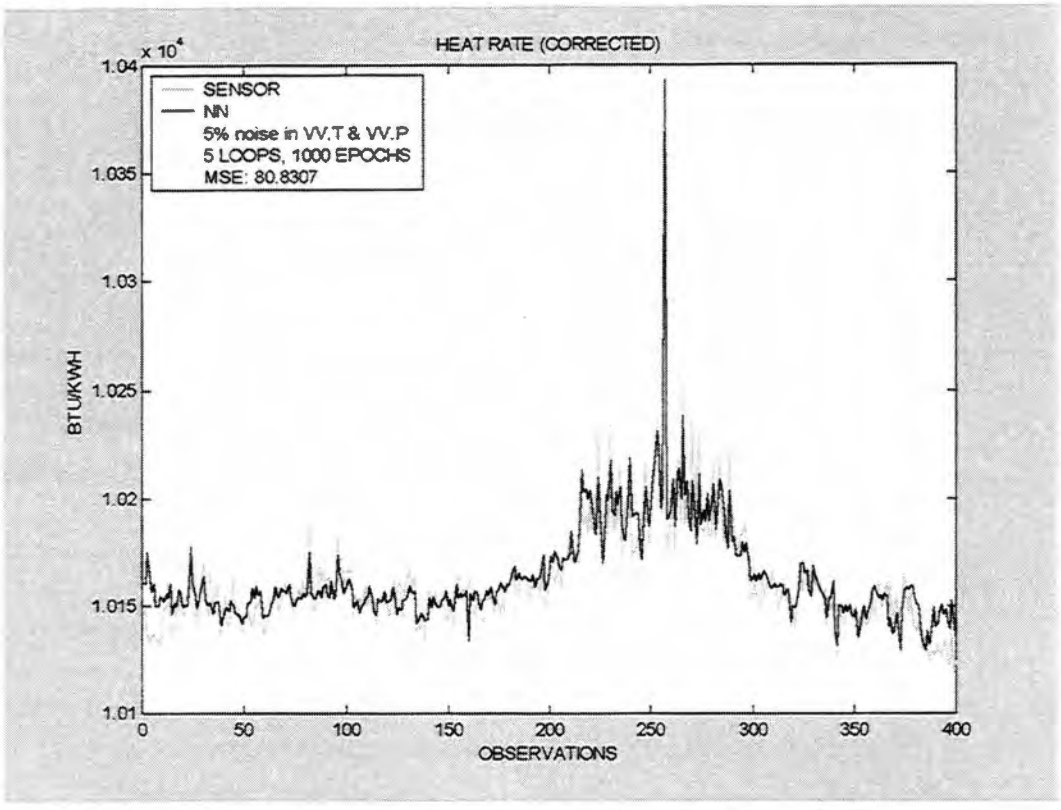

Figure A-19 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 2 hidden nodes (tansig) and linear output layer, [5% noise in VV.T & VV.P]

Table A-1 MSE for various neural network models

Best MSE	Elapsed time (seconds)*	Time per loop (seconds)	Total epochs	Type training and validation (defaults: 5 loops at 1000 epochs, 2-node hidden (middle) layer, P & T all 400 observations)
80.8	387.2	77.4	5005	no validation vectors
74.8	280.5	56.1	5005	no validation vectors, 3-node middle layer
64.4	345.0	69.0	5005	no validation vectors, 4-node middle layer
63.2	366.8	73.4	5005	no validation vectors, 5-node middle layer
60.3	500.4	100.1	5005	no validation vectors, 6-node middle layer
58.5	590.4	118.1	5005	no validation vectors, 7-node middle layer
45.2	908.6	181.7	5005	no validation vectors, 10-node middle layer
116.1	12.3	2.5	97	VV every 3
93.9	15.7	3.1	151	VV every 4
234.2	10.0	2.0	69	VV every 4, except observations 216:289
83.1	164.3	32.9	2357	VV every 4, all observations 216:289
89.4	142.7	28.5	2367	VV every 4, all observations 216:289, 3-node middle layer
144.9	15.0	3.0	118	VV every 4, 4-node middle layer
147.8	13.3	2.7	97	VV every 5
98.8	14.1	2.8	123	VV every 6
81.0	124.7	24.9	1388	VV every 7
137.7	13.9	2.8	126	VV every 7, 1-node middle layer
76.7	24.4	4.9	299	VV every 7, 3-node middle layer
136.8	17.4	3.5	160	VV every 7, 4-node middle layer
128.9	14.2	2.8	126	VV every 8
80.8	2767.4	55.3	43300	VV.T & VV.P same as T & P with 5% noise in VV.T, 50 loops
80.8	124.5	24.9	2012	VV.T & VV.P same as T & P with 5% noise in VV.T and VV.P
75.2	209.3	41.9	3426	VV.T & VV.P same as T & P with 5% noise in VV.T and VV.P, 3-node middle layer
65.5	241.2	48.2	3282	VV.T & VV.P same as T & P with 5% noise in VV.T and VV.P, 4-node middle layer
65.1	315.2	63.0	4319	VV.T & VV.P same as T & P with 5% noise in VV.T, 4-node middle layer
63.2	384.3	76.9	4141	VV.T & VV.P same as T & P with 5% noise in VV.T, 5-node middle layer
60.6	383.5	76.7	4014	VV.T & VV.P same as T & P with 5% noise in VV.T, 6-node middle layer
81.4	114.2	22.8	1882	VV.T & VV.P same as T & P with 5% noise in P and T
74.4	245.3	49.1	3757	VV.T & VV.P same as T & P with 5% noise in P and T, 3-node middle layer
81.3	139.2	27.8	2374	VV.T & VV.P same as T & P with 5% noise in P
80.8	1055.1	211.0	15005	VV.T & VV.P same as T & P, 3000 epochs
74.2	994.5	198.9	15005	VV.T & VV.P same as T & P, 3000 epochs, 3-node middle layer
83.4	286.2	57.2	5005	P & T every 1,2,3; VV every 4 except no validation vectors were used

* display every 10 epochs

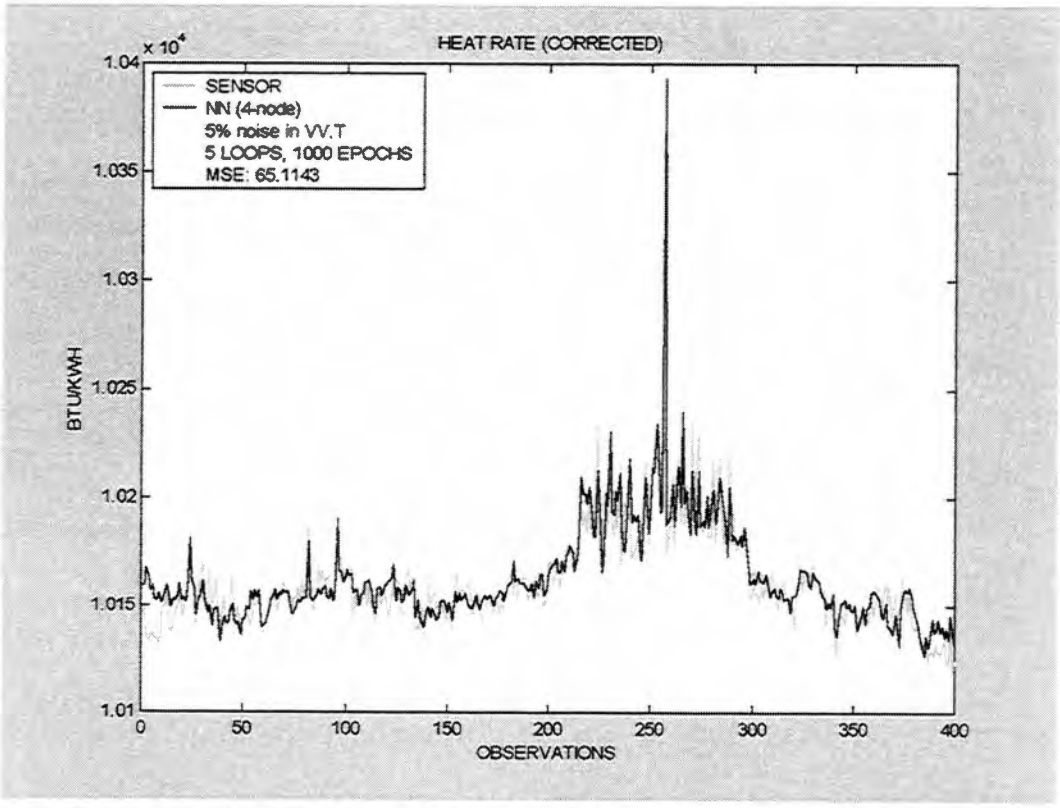

Figure A-20 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 4 hidden nodes (tansig) and linear output layer, [5% noise in VV.T]

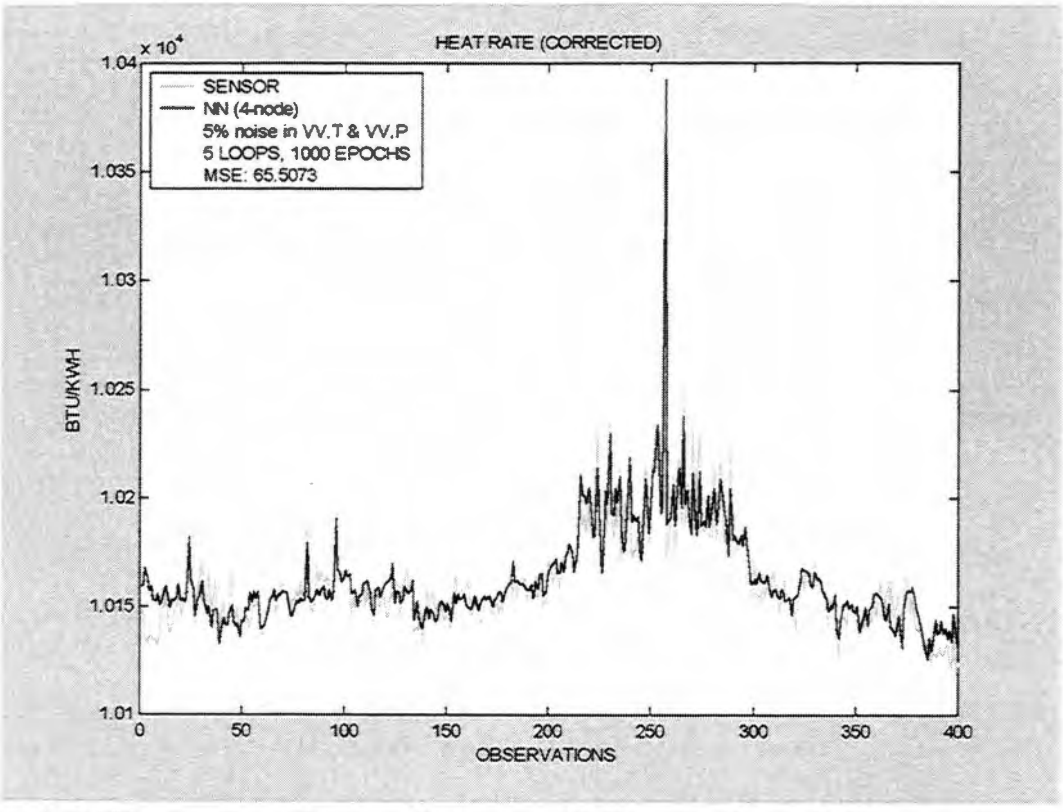

Figure A-21 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 3-layer NN with 4 hidden nodes (tansig) and linear output layer, [5% noise in VV.T & VV.P]

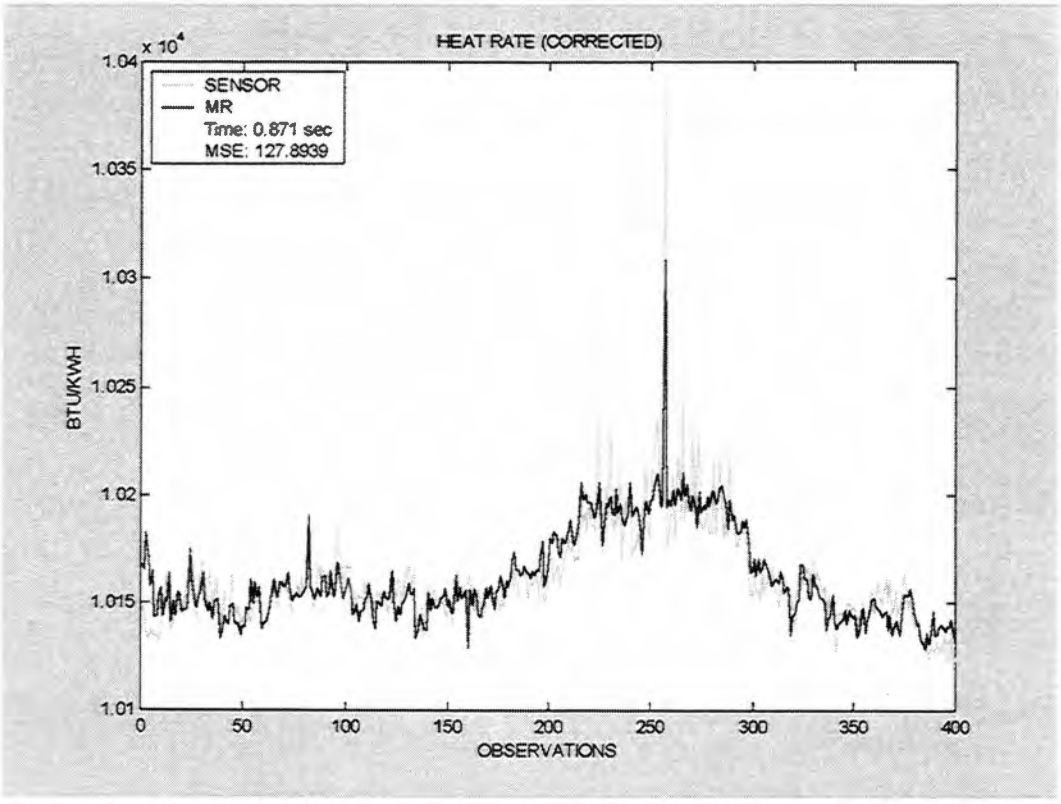

Figure A-22 Multiple regression estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling

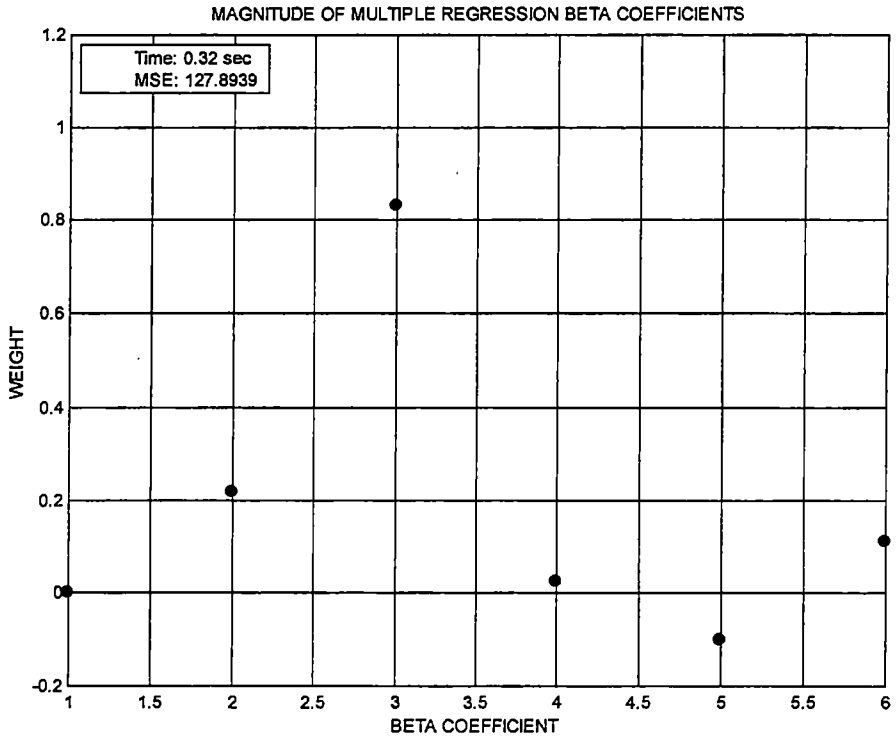

Figure A-23 Beta Coefficients for multiple regression estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling

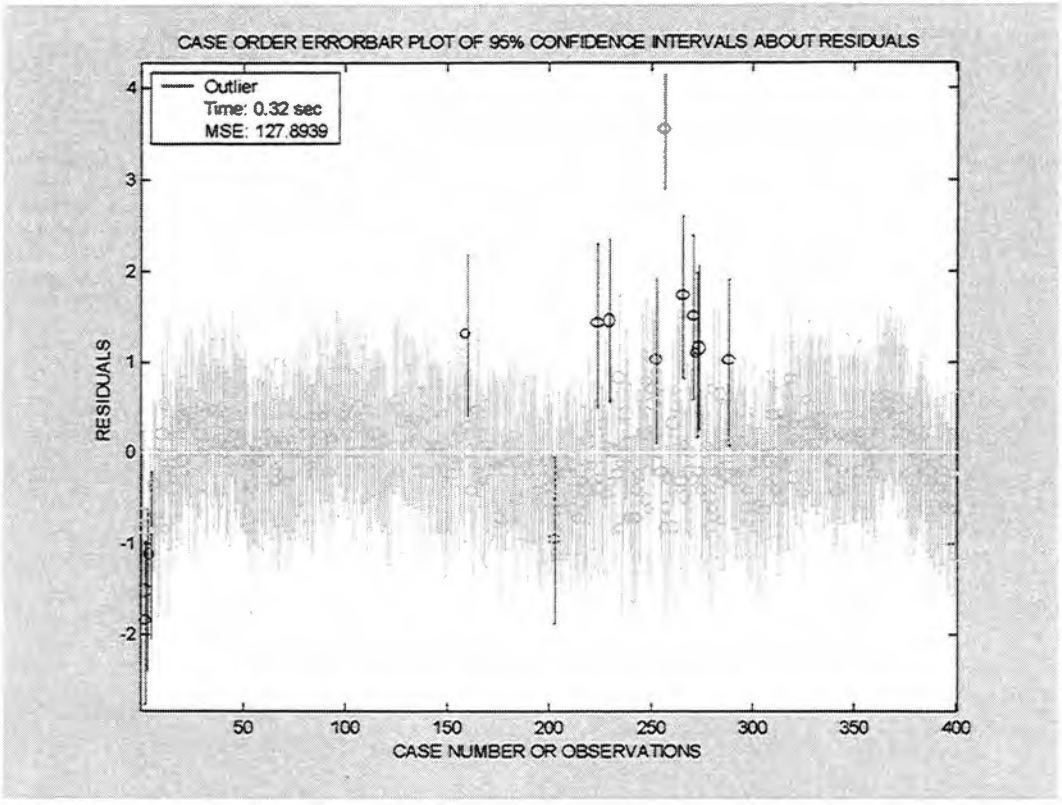

Figure A-24 Error Bar Plot for multiple regression estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling

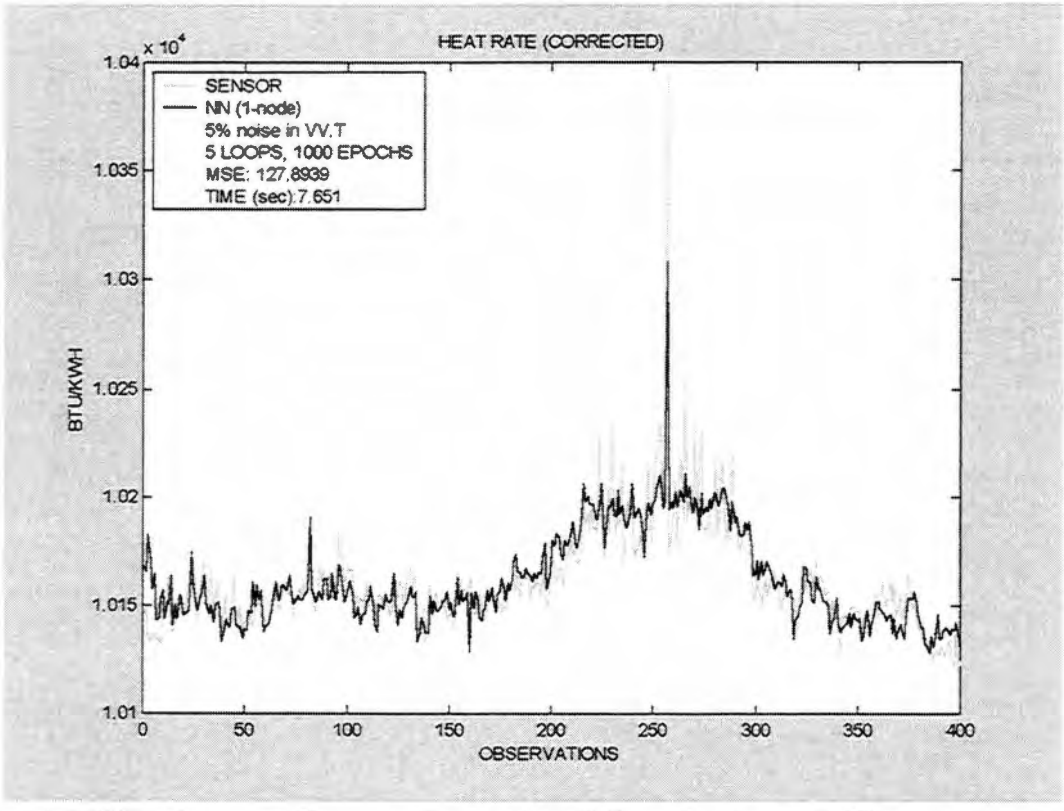

Figure A-25 NN estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling, 2-layer NN with 1 hidden node (linear) and linear output layer, [5% noise in VV.T]

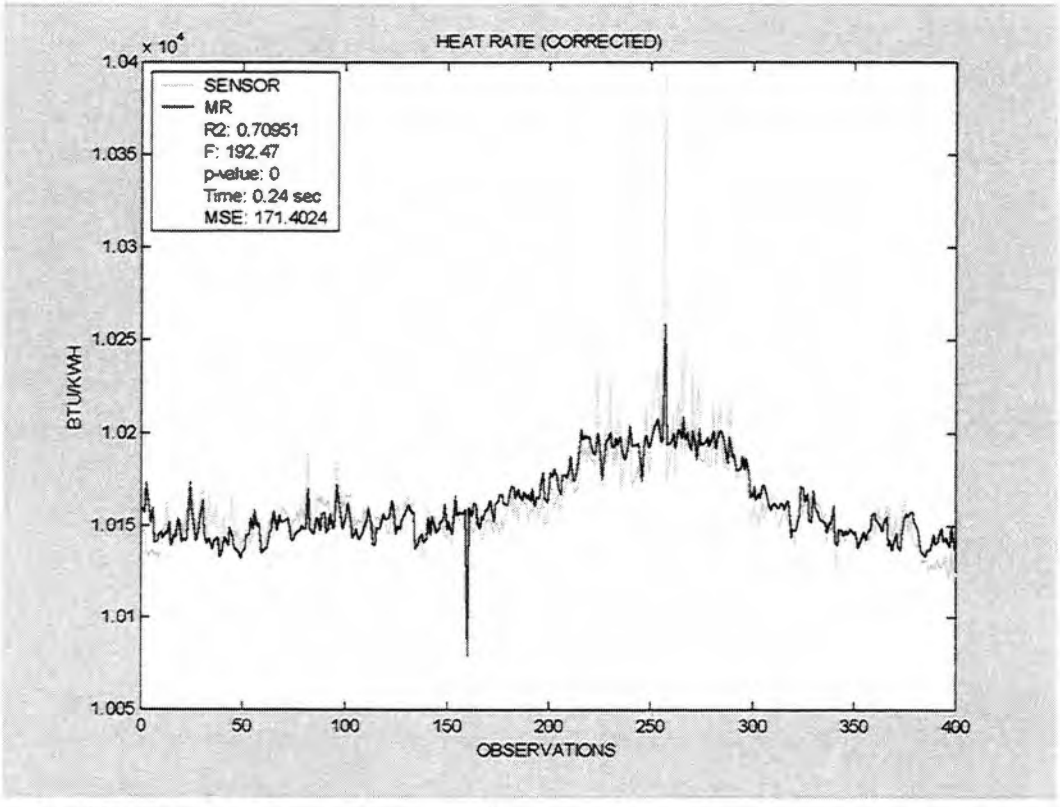

Figure A-26 Multiple regression estimate, predictors are principal component scores (5 dimensional), SD pre-scaling

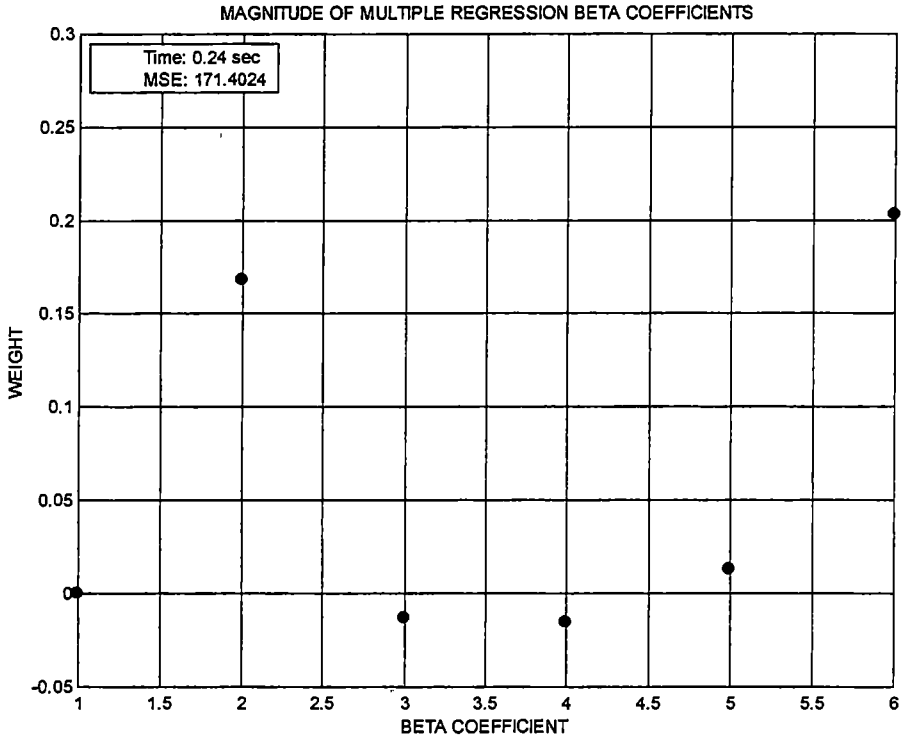

Figure A-27 Beta Coefficients for multiple regression estimate, predictors are principal component scores (5 dimensional), SD pre-scaling

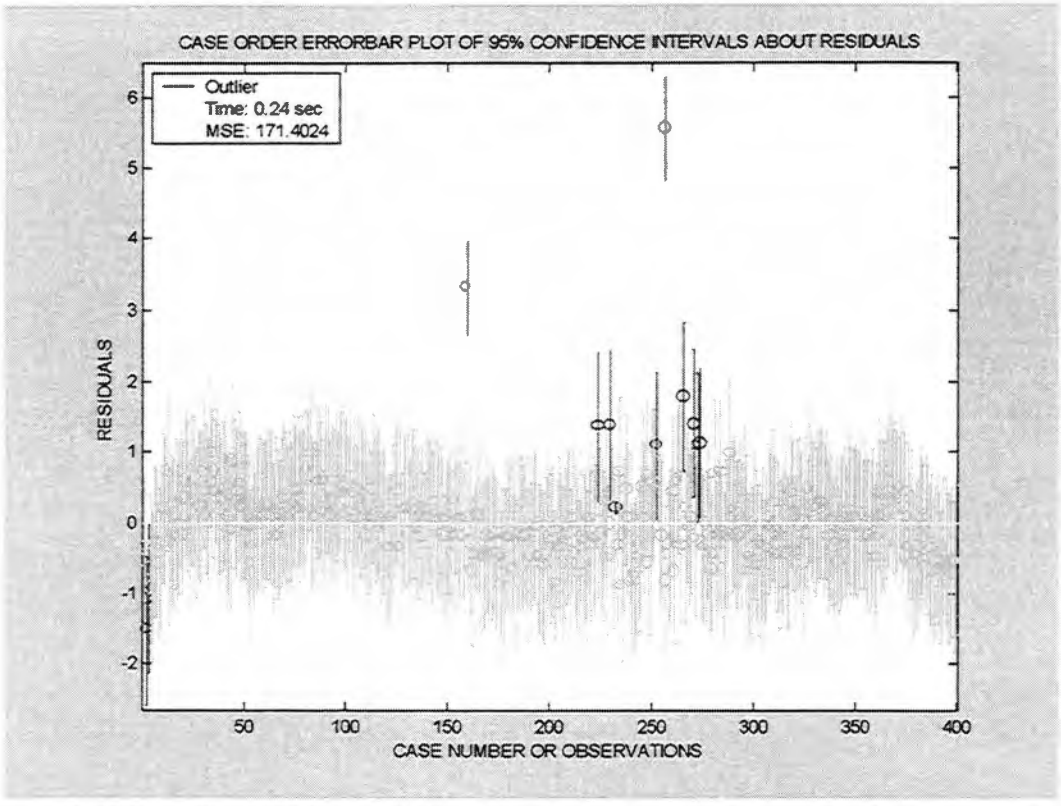

Figure A-28 Error Bar Plot for multiple regression estimate, predictors are principal component scores (5 dimensional), SD pre-scaling

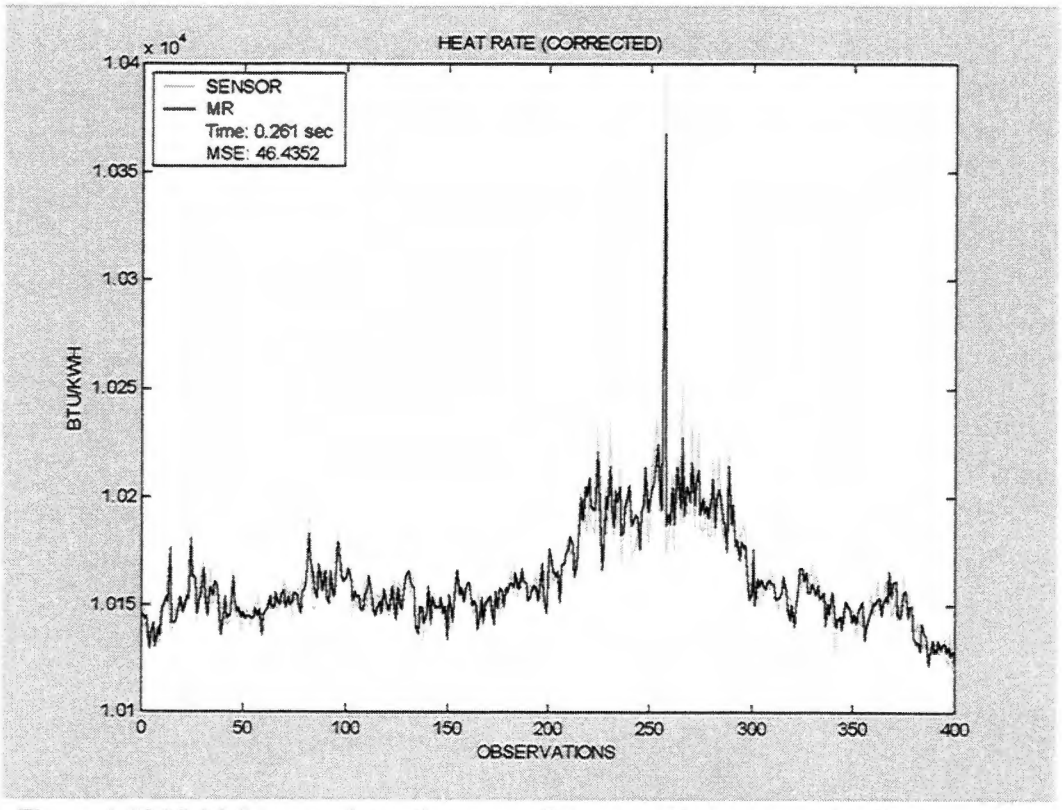

Figure A-29 Multiple regression estimate, predictors are 43 state variables, SD pre-scaling

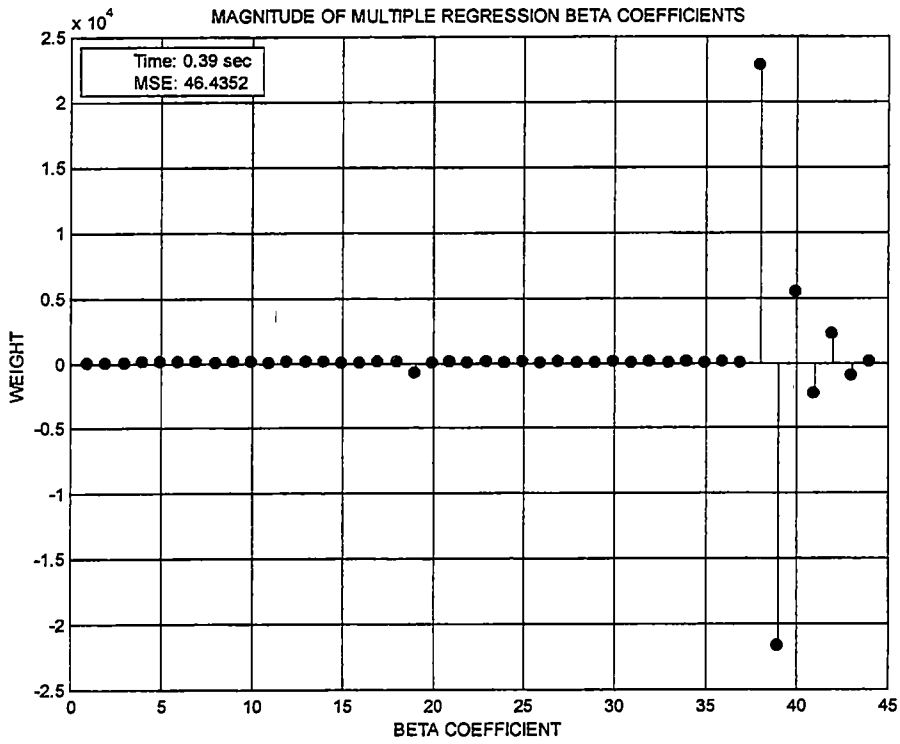

Figure A-30 Beta Coefficients for multiple regression estimate, predictors are 43 state variables, SD pre-scaling

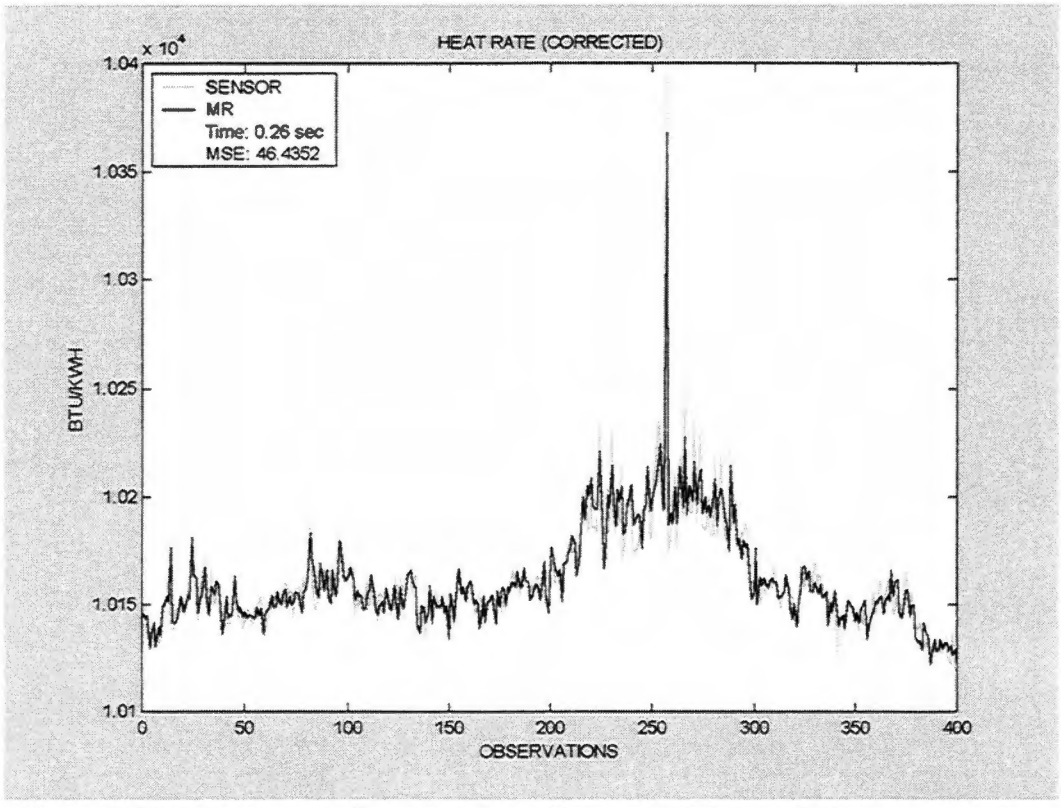

Figure A-31 Multiple regression estimate, predictors are 43 state variables, no pre-scaling

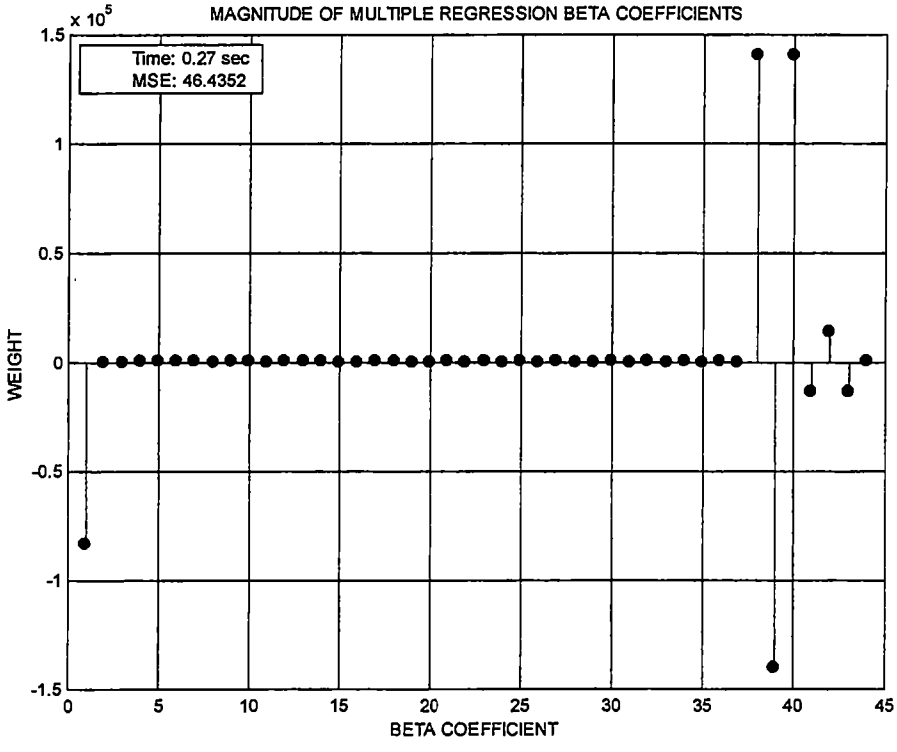

Figure A-32 Beta Coefficients for multiple regression estimate, predictors are 43 state variables, no pre-scaling

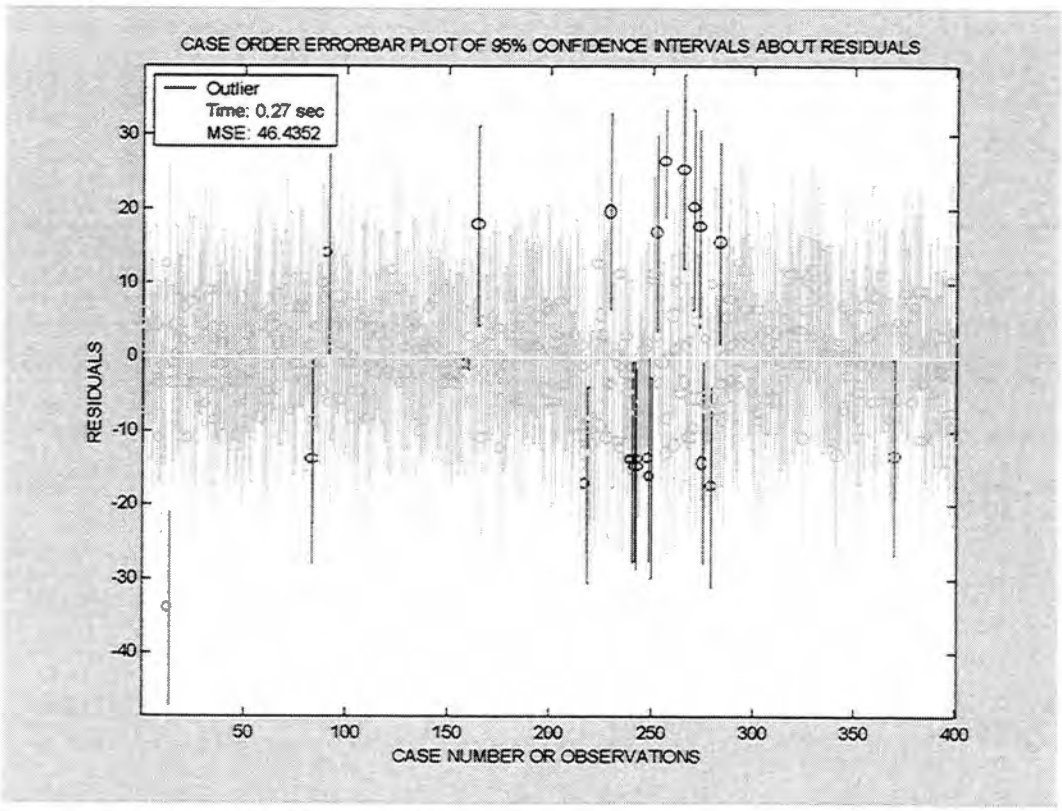

Figure A-33 Error Bar Plot for multiple regression estimate, predictors are 43 state variables, no pre-scaling

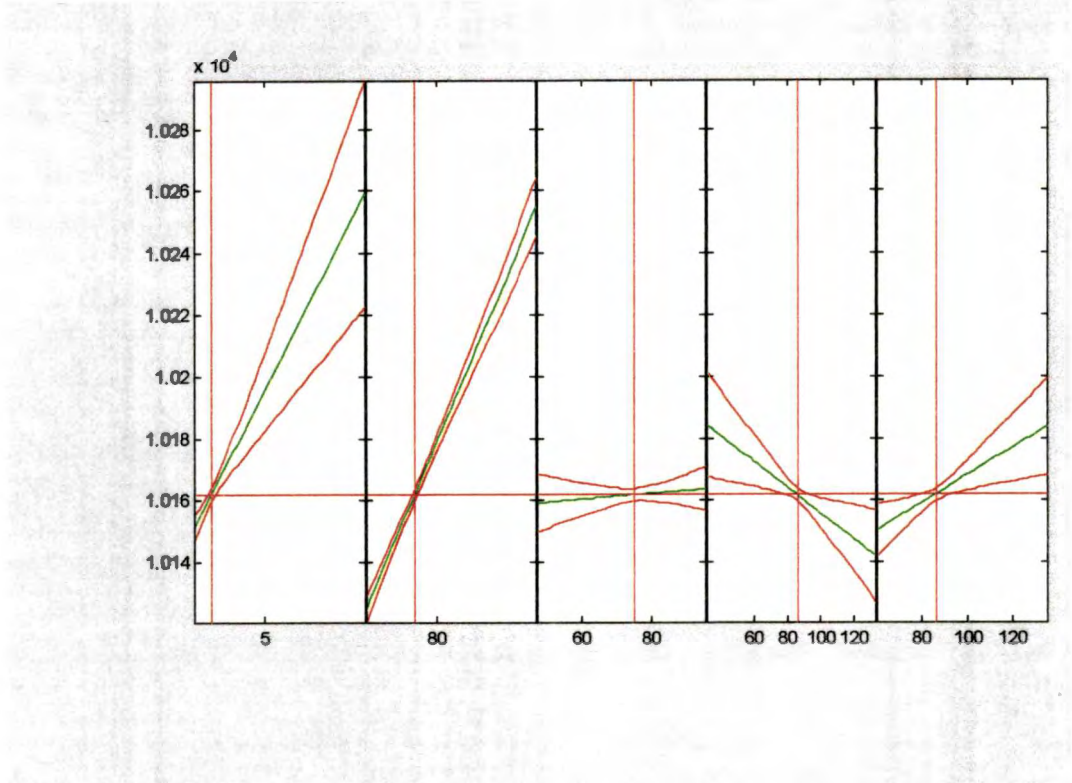

Figure A-34 Interactive fitting and visualization of response surface for linear multiple regression estimate, predictors are factor scores (5 dimensional, sorted), SD pre-scaling

Table A-2 MSE values for various factor scores using multiple regression

Number of Factors	MSE using MR	R^2	Notes
1	202.41	65.70%	Common Factors
2	201.52	65.85%	Factor score estimation by ideal variables
3	201.09	65.92%	
4	190.02	67.80%	
5	127.89	78.33%	
6	125.36	78.75%	
7	123.53	79.06%	
10	121.78	79.36%	
5	224.54	61.95%	Factor score estimation by regression, R observed
5	224.49	61.96%	Factor score estimation by theoretical model
5	122.71	79.20%	Factor score estimation by theoretical model, (U2 = I - pattern*pattern')
5	224.48	61.96%	Factor score estimation using Bartlett's approach with Anderson and Rubin modification
5	122.71	79.20%	Factor score estimation using Bartlett's approach with Anderson and Rubin modification, (U2 = I - pattern*pattern')
1	202.58	65.67%	PCA factors (largest eigenvalues)
2	201.57	65.84%	
3	201.10	65.92%	
4	200.97	65.94%	
5	171.40	70.95%	First five principal component scores
6	138.48	76.53%	
7	130.31	77.92%	
10	119.03	79.83%	
43	46.44	92.13%	All principal component scores
5	111.93	81.03%	Five principal component scores that correlate highest with heat rate Component indices 1, 5, 6, 12, 18

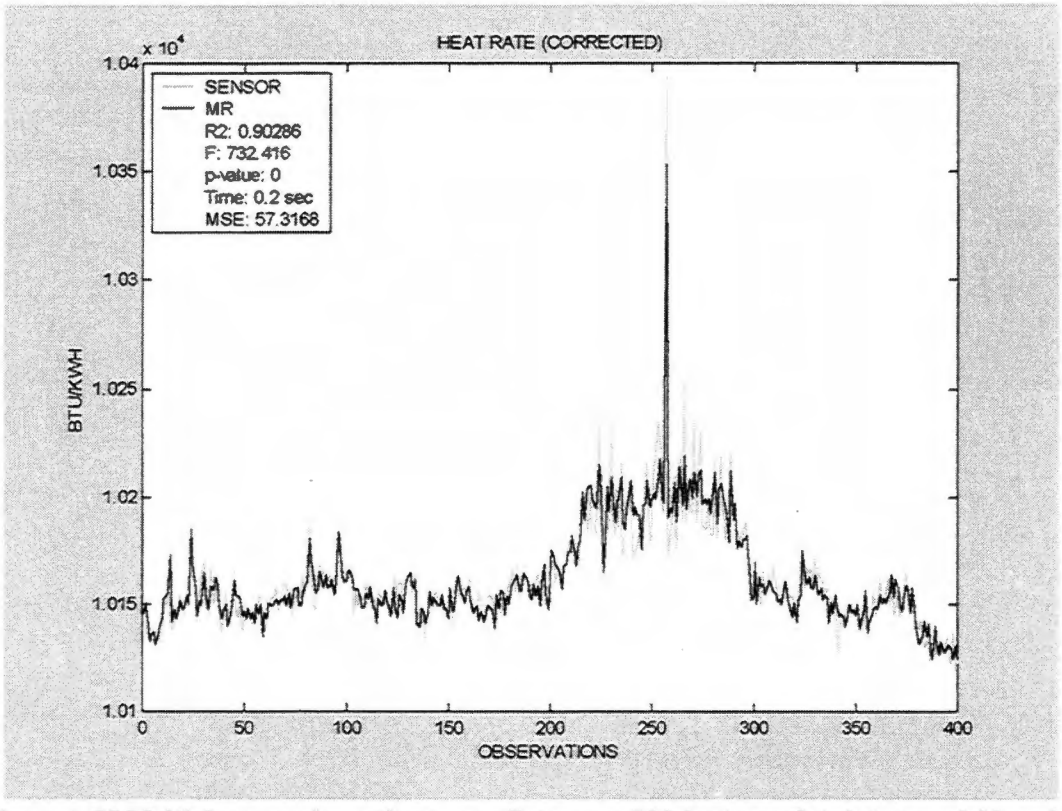

Figure A-35 Multiple regression estimate, predictors are 5 highest correlated state variables with heat rate, SD pre-scaling

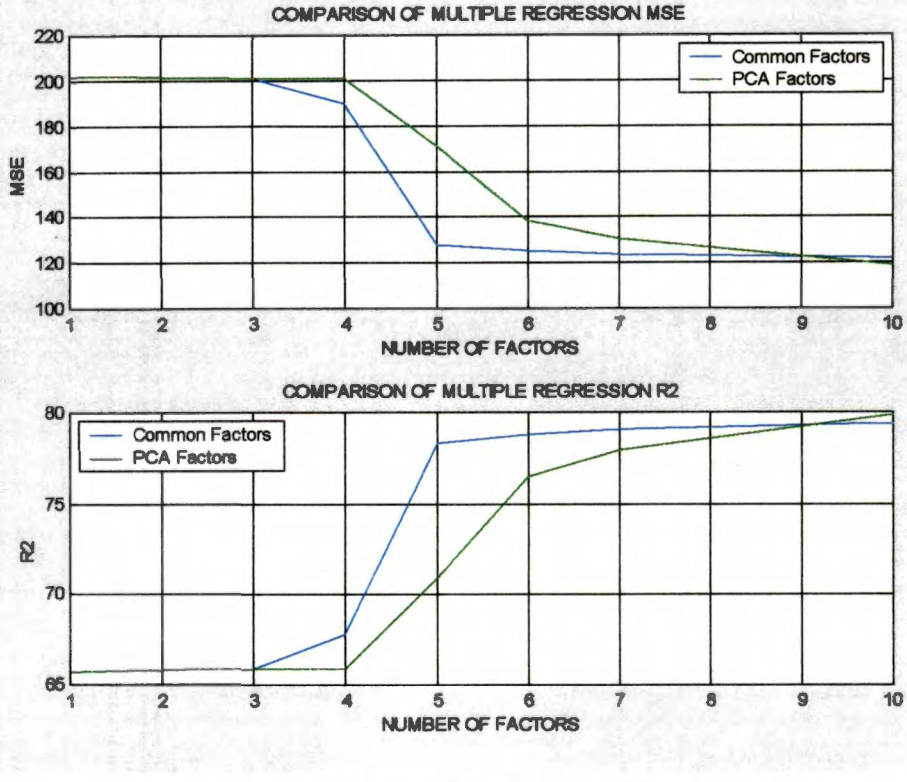

Figure A-36 MR MSE and R^2 for Common and PCA Factors

Table A-3 State variable descriptions

Index	State Variable
1	'AVG CONDENSER PRESSURE'
2	'Avg. CNDSR(A)IN CW TEMP L A/B'
3	'Avg. CNDSR(A)IN CW TEMP L C/D'
4	'Avg. CNDSR(B)IN CW TEMP L A/B'
5	'Avg. CNDSR(B)IN CW TEMP L C/D'
6	'Avg. CNDSR(C) OUT CW TEMP A/B'
7	'Avg. CNDSR(C) OUT CW TEMP C/D'
8	'Avg. CNDSR(C)IN CW TEMP L A/B'
9	'Avg. CNDSR(C)IN CW TEMP L C/D'
10	'CNDS PMP A EFF'
11	'CNDS PMP B EFF'
12	'CNDS PMP C EFF'
13	'CONDENSER(A) PRESSURE'
14	'CONDENSER(B) PRESSURE'
15	'CONDENSER(C) PRESSURE'
16	'FW PMP A EFF'
17	'FW PMP B EFF'
18	'FW PMP B FW SP VOL'
19	'HTR 1A DRAIN TEMP'
20	'HTR 1B DRAIN TEMP'
21	'HTR 1C DRAIN TEMP'
22	'HTR 2A DRAIN TEMP'
23	'HTR 2B DRAIN TEMP'
24	'HTR 2C DRAIN TEMP'
25	'HTR 3A DRAIN TEMP'
26	'HTR 3B DRAIN TEMP'
27	'HTR 3C DRAIN TEMP'
28	'HTR 4A DRAIN TEMP'
29	'HTR 4B DRAIN TEMP'
30	'HTR 4C DRAIN TEMP'
31	'HTR 5A DRAIN TEMP'
32	'HTR 5B DRAIN TEMP'
33	'HTR 6A DRAIN TEMP'
34	'HTR 6B DRAIN TEMP'
35	'HTR 7A DRAIN TEMP'
36	'HTR 7B DRAIN TEMP'
37	'HTR DRN PMP A EFF'
38	'HTR DRN PMP A EFF DESIGN'
39	'HTR DRN PMP A EFF DEV'
40	'HTR DRN PMP B EFF'
41	'HTR DRN PMP B EFF DESIGN'
42	'HTR DRN PMP B EFF DEV'
43	'TRB HEAT RATE GROSS DESIGN'

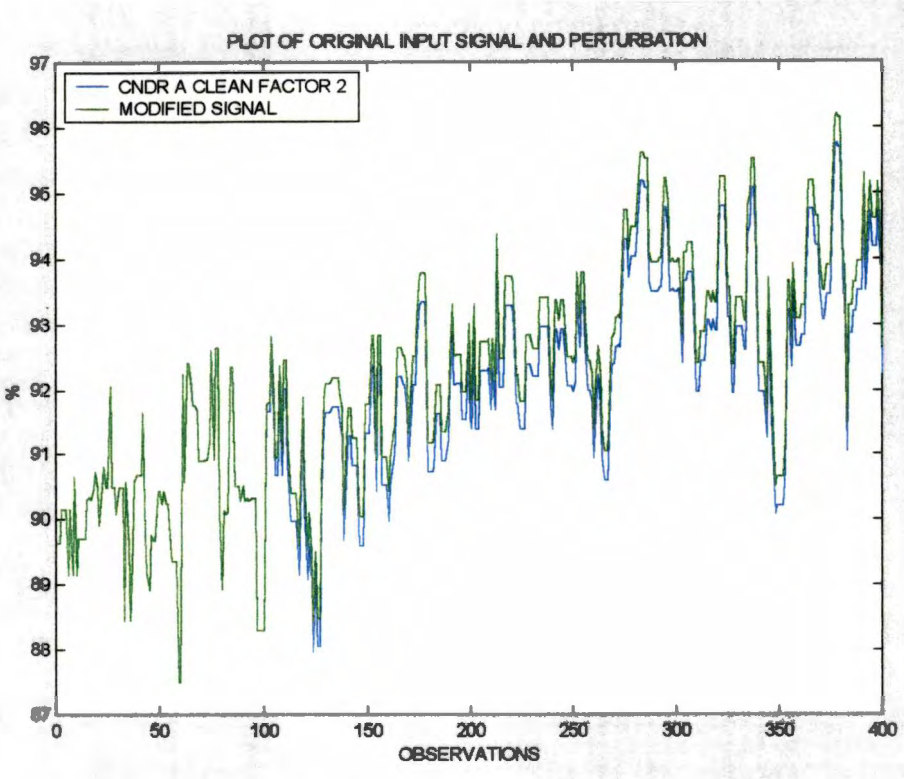

Figure A-37 Change in set point signal for a .5% perturbation

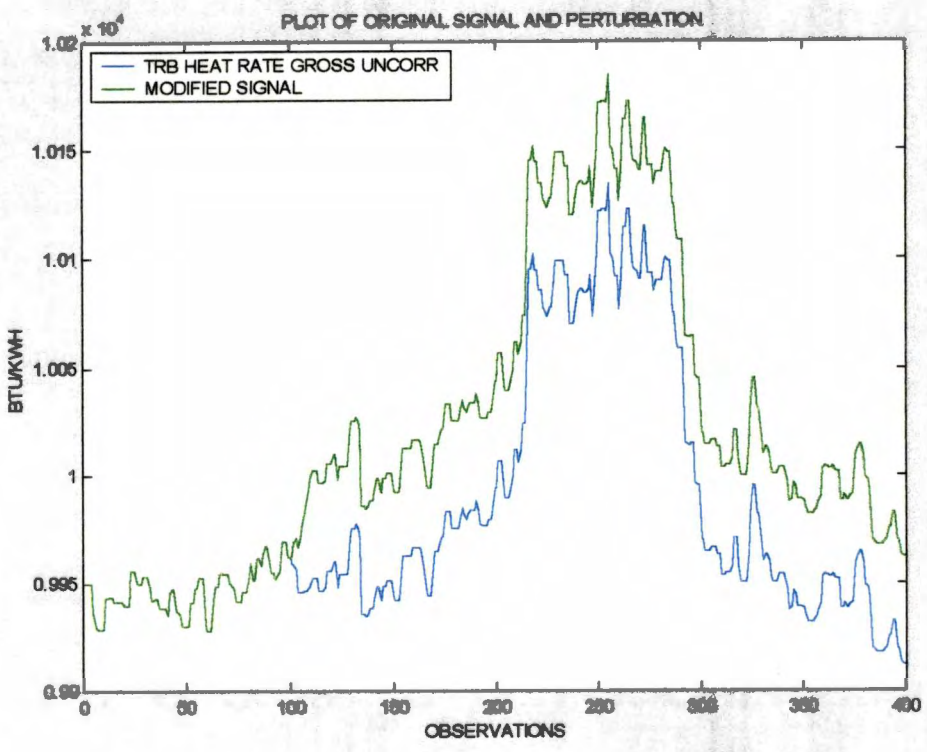

Figure A-38 Change in set point signal for a .5% perturbation

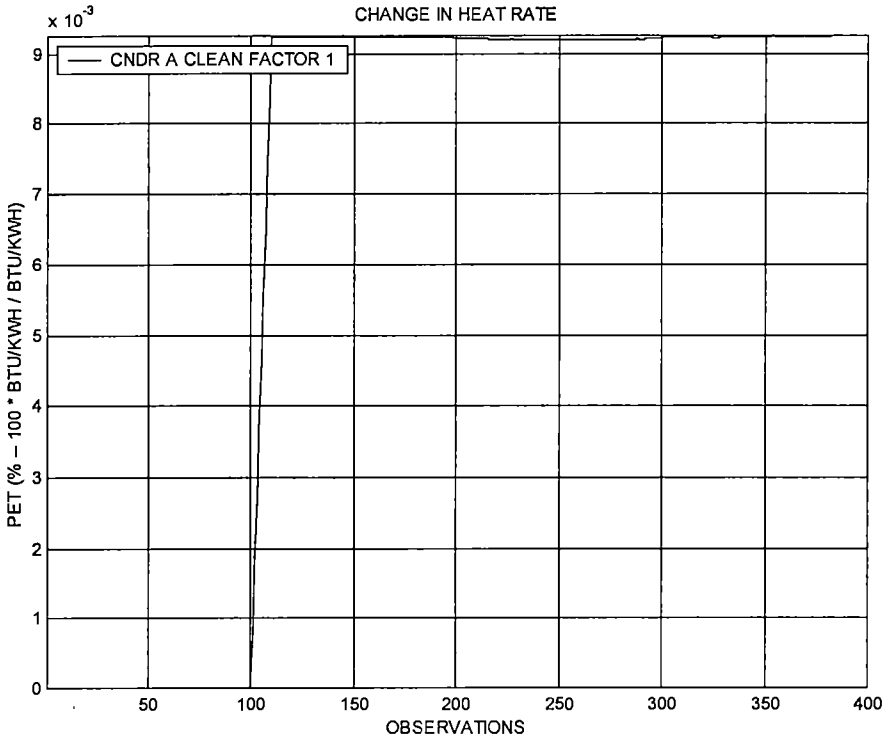

Figure A-39 Change in heat rate for a 5% change in input, MR-MR mappings

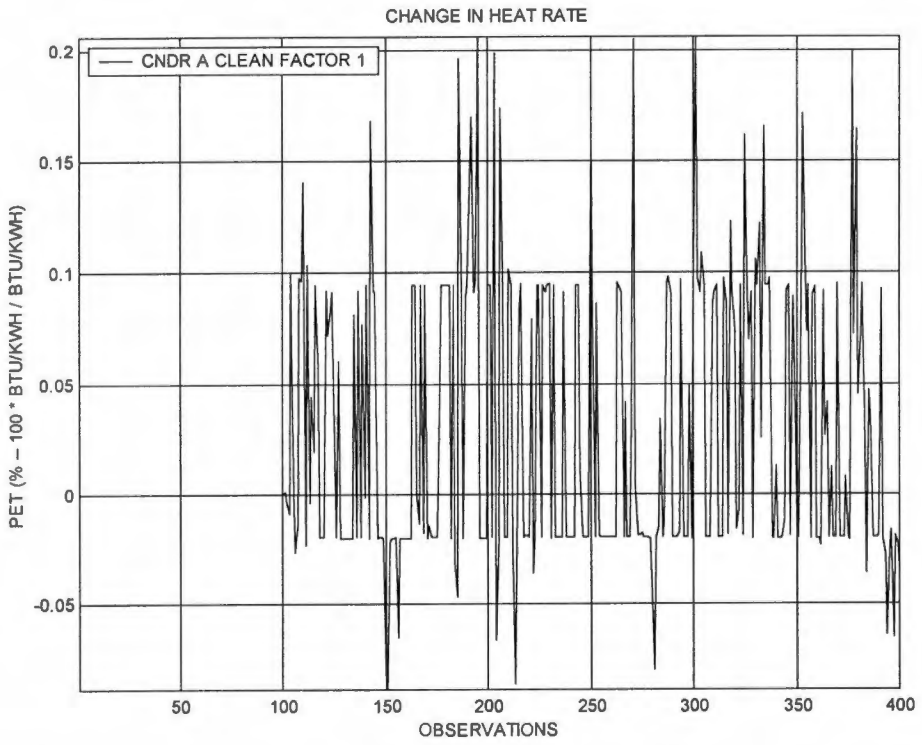

Figure A-40 Change in heat rate for a 5% change in input, MR-NN mappings, NN overtrained

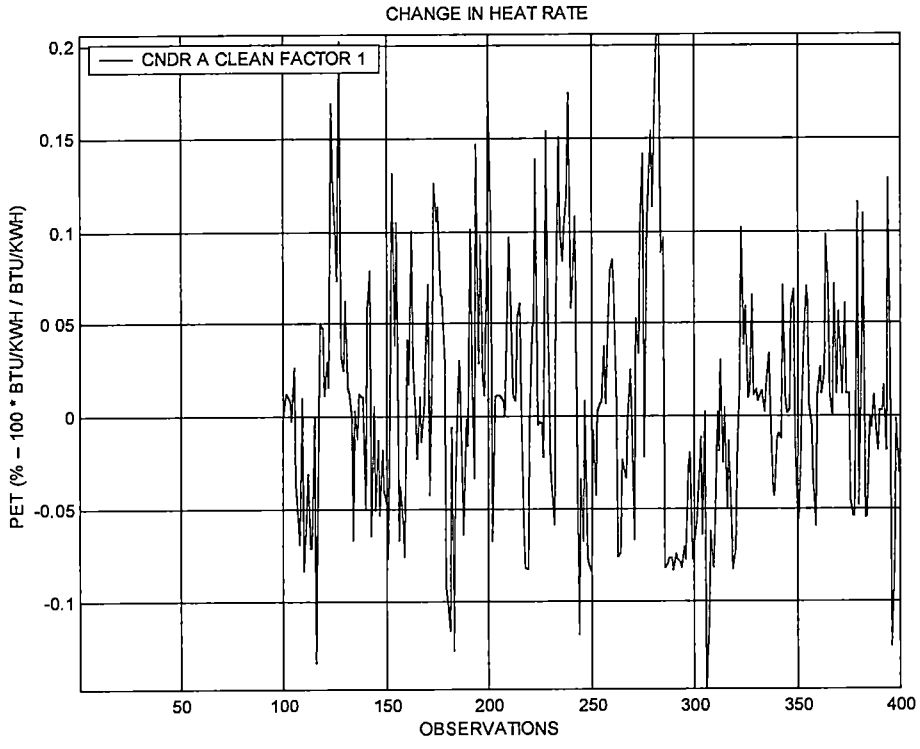

Figure A-41 Change in heat rate for a 5% change in input, MR-NN mappings, all PCA inputs to NN with cross-validation

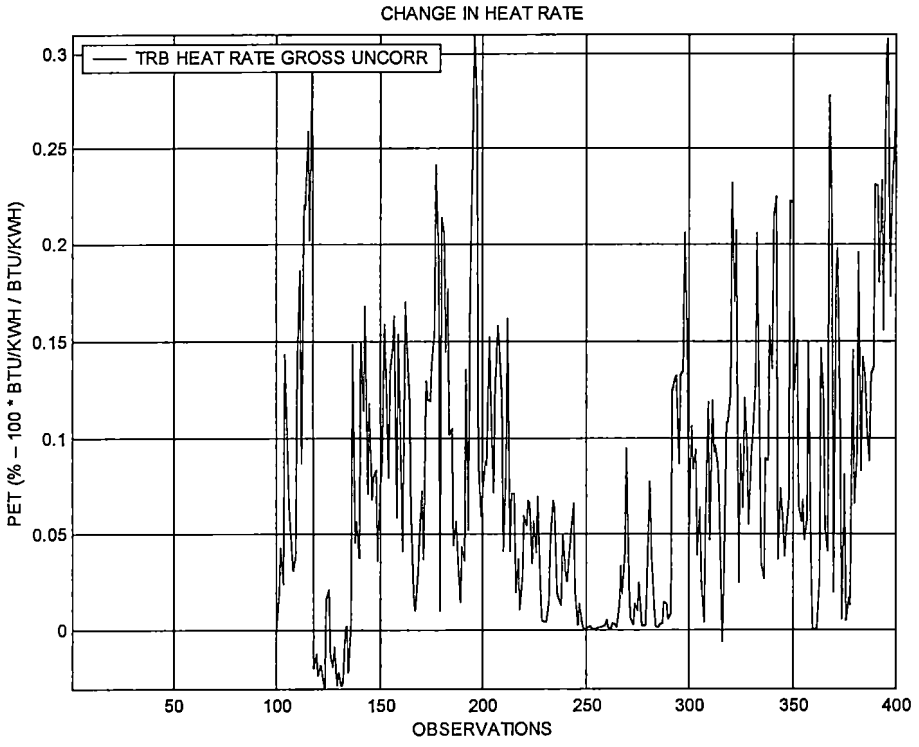

Figure A-42 Change in heat rate for a .5% change in input, MR-NN mappings, 21 SD scores with cross-validation

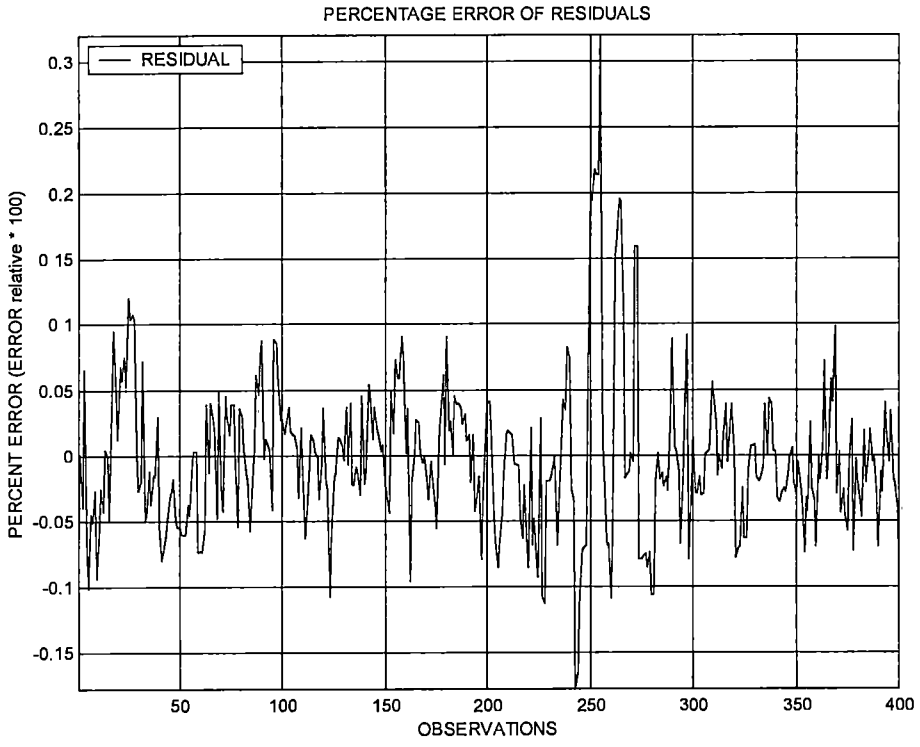

Figure A-43 Typical residual plot showing change in heat rate mapping (MAP2) using a NN with cross-validation (21 SD scores as predictors, heat rate as criterion)

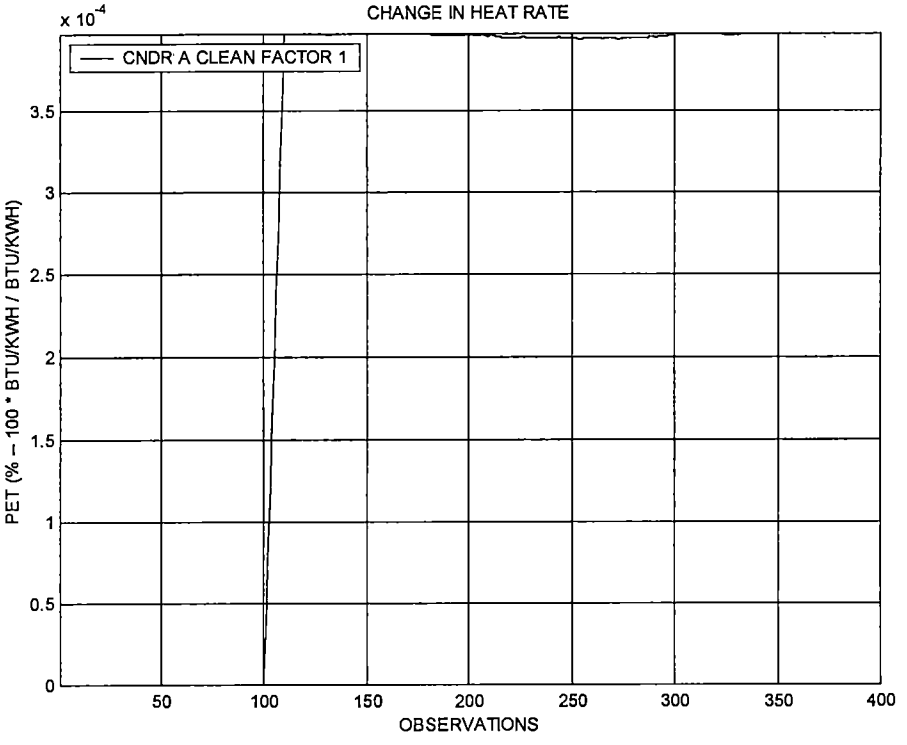

Figure A-44 Change in heat rate for a .5% change in input, MR-NN mappings, 5 PCA inputs to NN with cross-validation

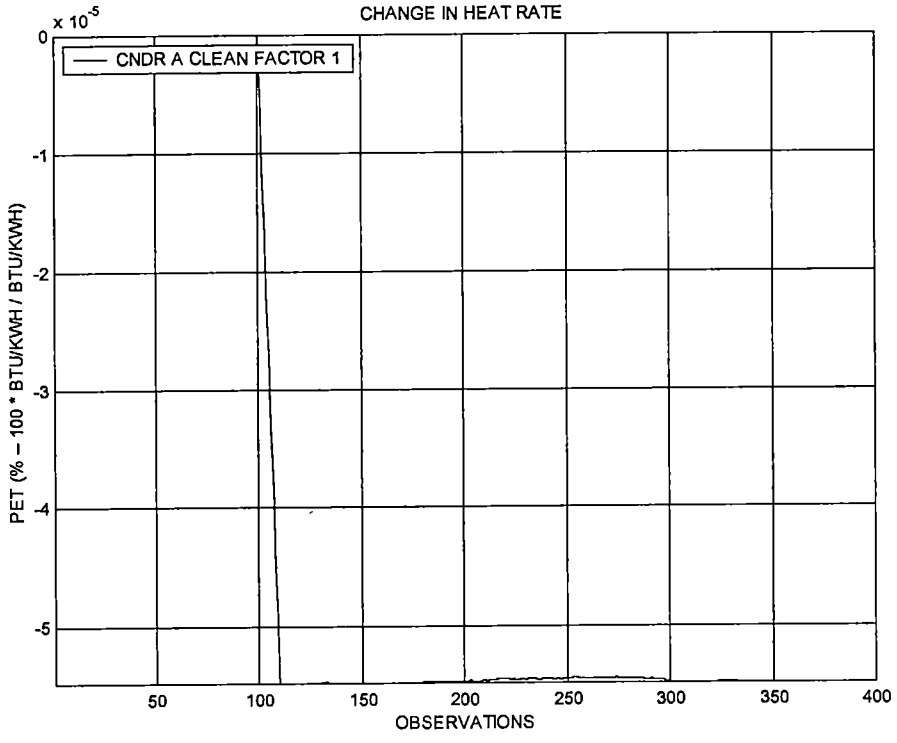

Figure A-45 Change in heat rate for a .5% change in input, MR-NN mappings, 5 FACTOR inputs to NN with cross-validation

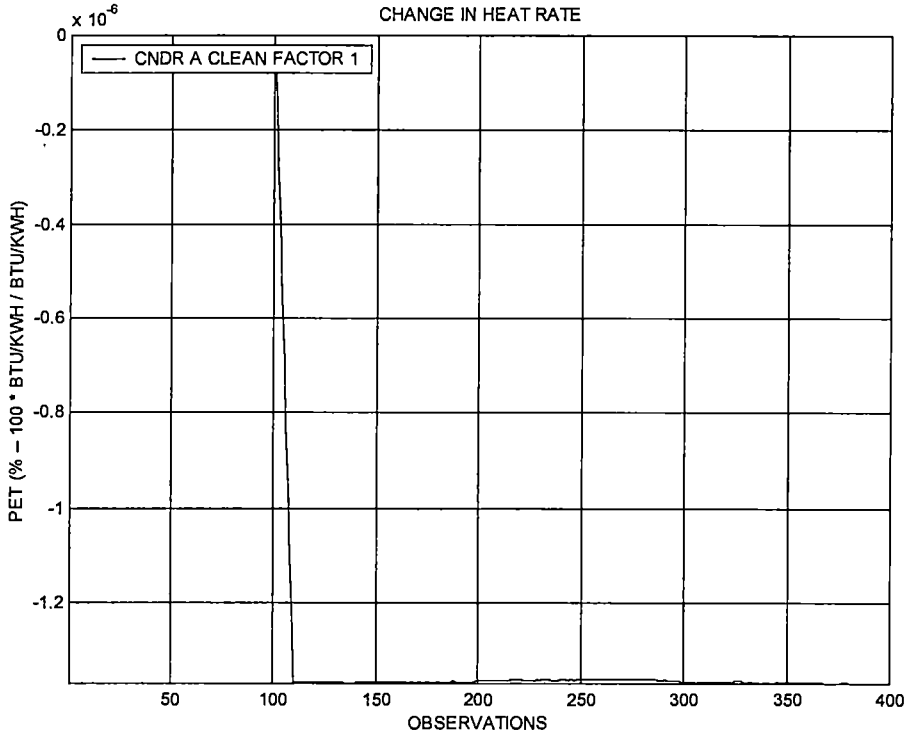

Figure A-46 Change in heat rate for a .5% change in input, MR-MR mappings, 5 PCA inputs

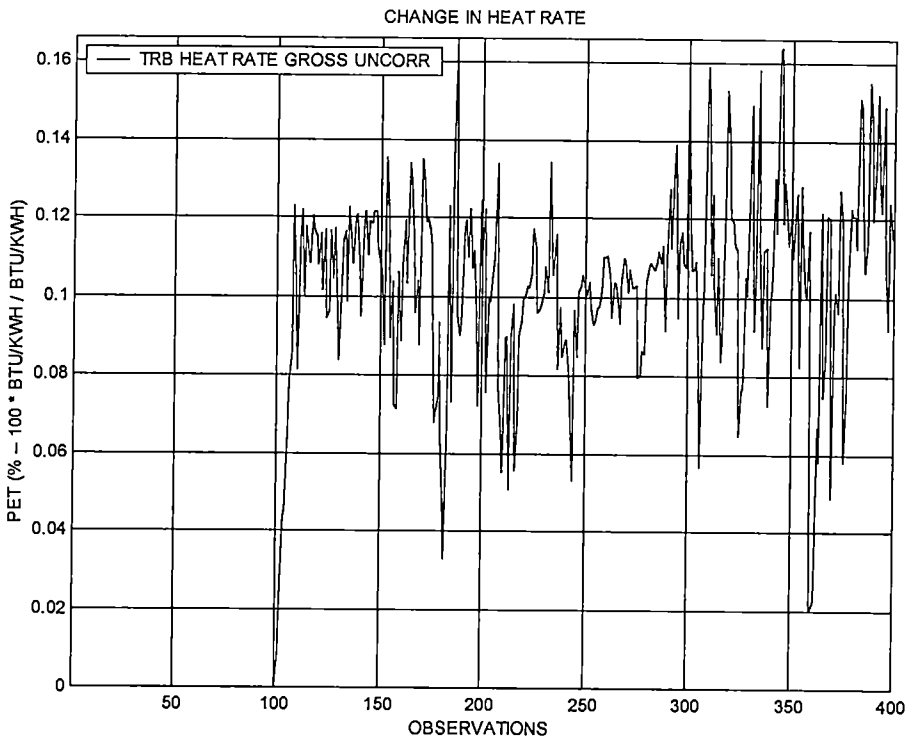

Figure A-47 Change in heat rate for a .5% change in input, MR-NN mappings, (per=3, nodes=4, loops=5, tansig-purelin, .005 pert, epochs=300, 21 SD inputs)

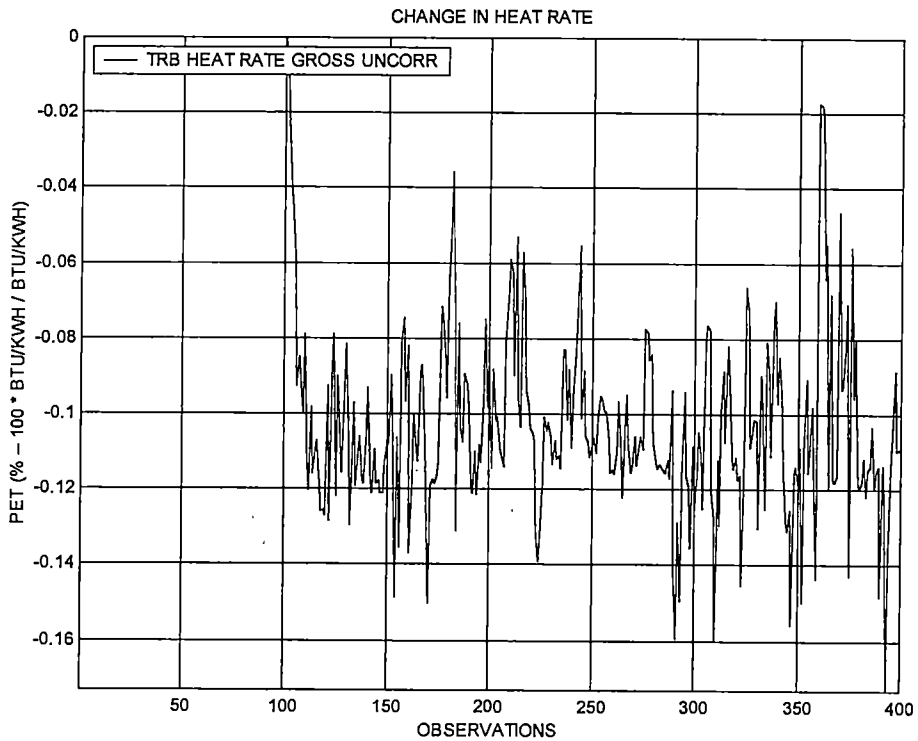

Figure A-48 Change in heat rate for a negative .5% change in input, MR-NN mappings, (per=3, nodes=4, loops=5, tansig-purelin, -.005 pert, epochs=300, 21 SD inputs)

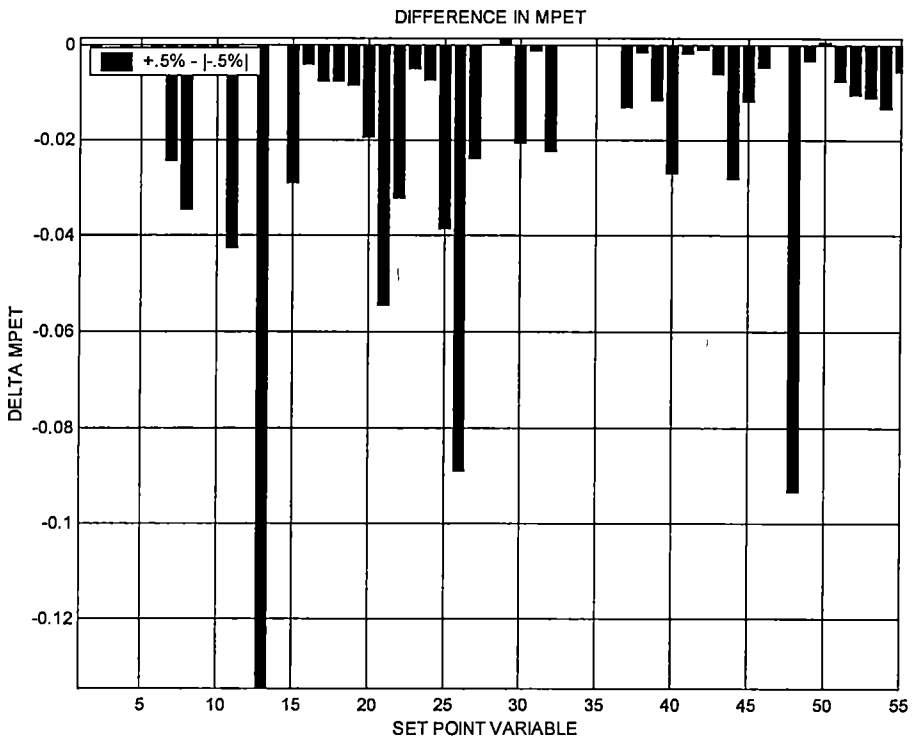

Figure A-49 Change in MPET for a +.5% and -.5% perturbation, MR-NN mappings, (per=3, nodes=4, loops=5, tansig-purelin, +.005 & -.005 pert, 21 SD inputs)

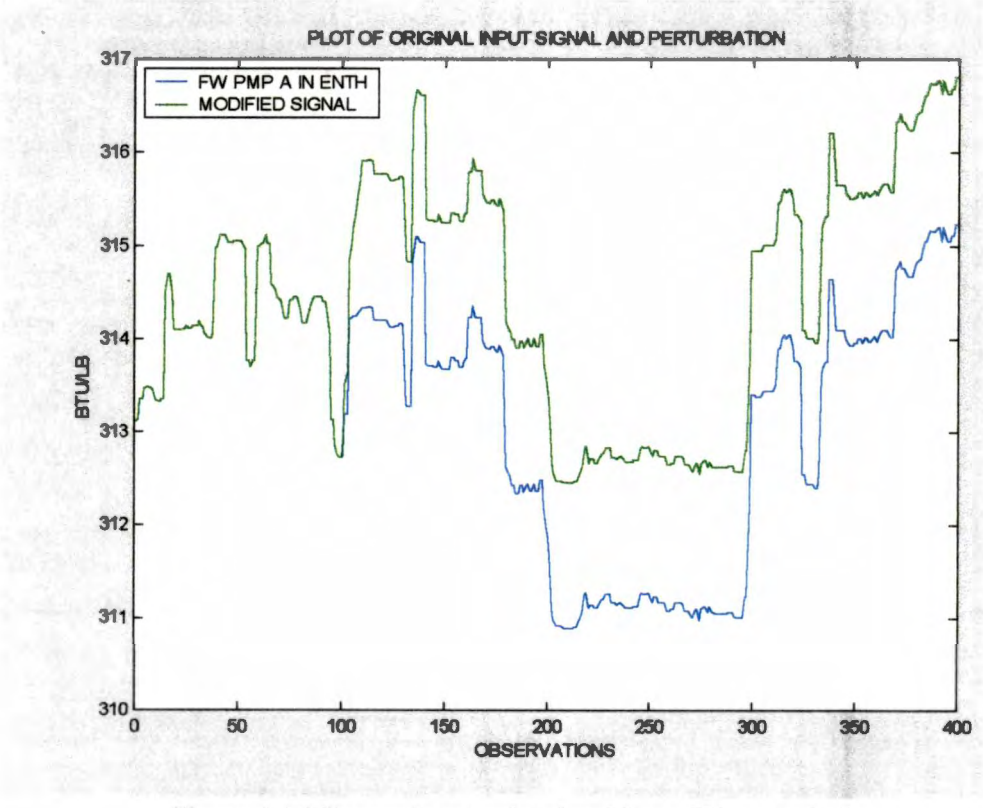

Figure A-50 Change in set point signal for a .5% perturbation

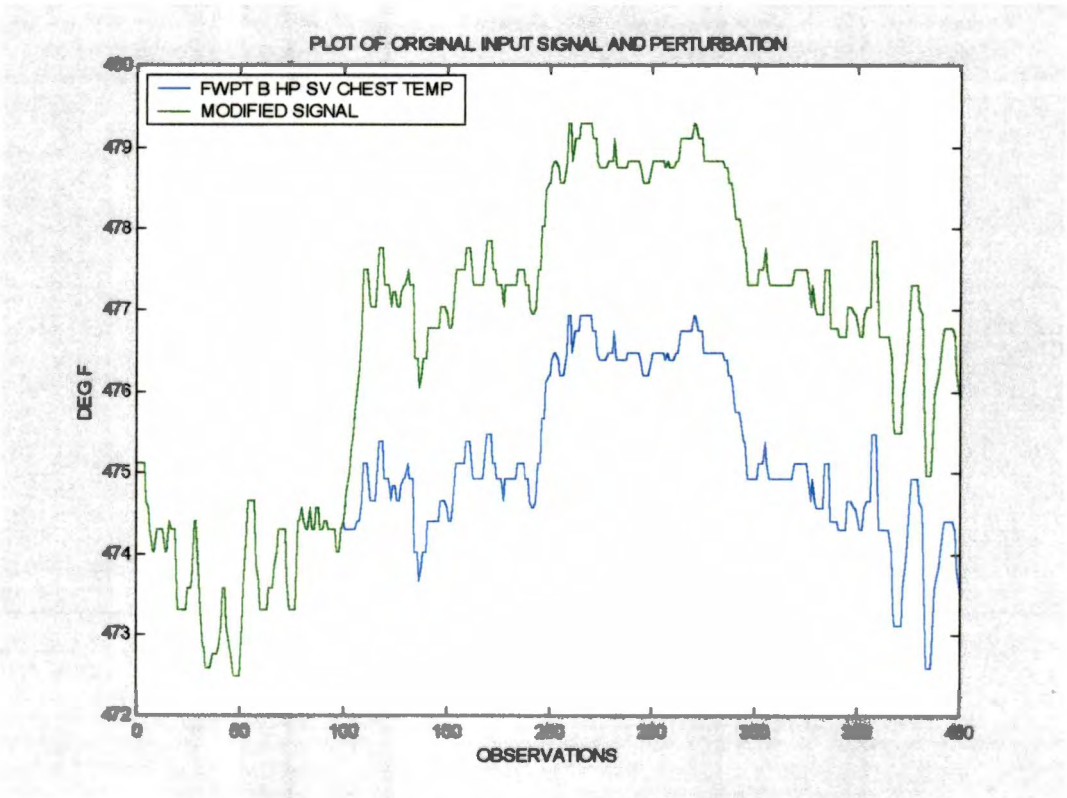

Figure A-51 Change in set point signal for a .5% perturbation

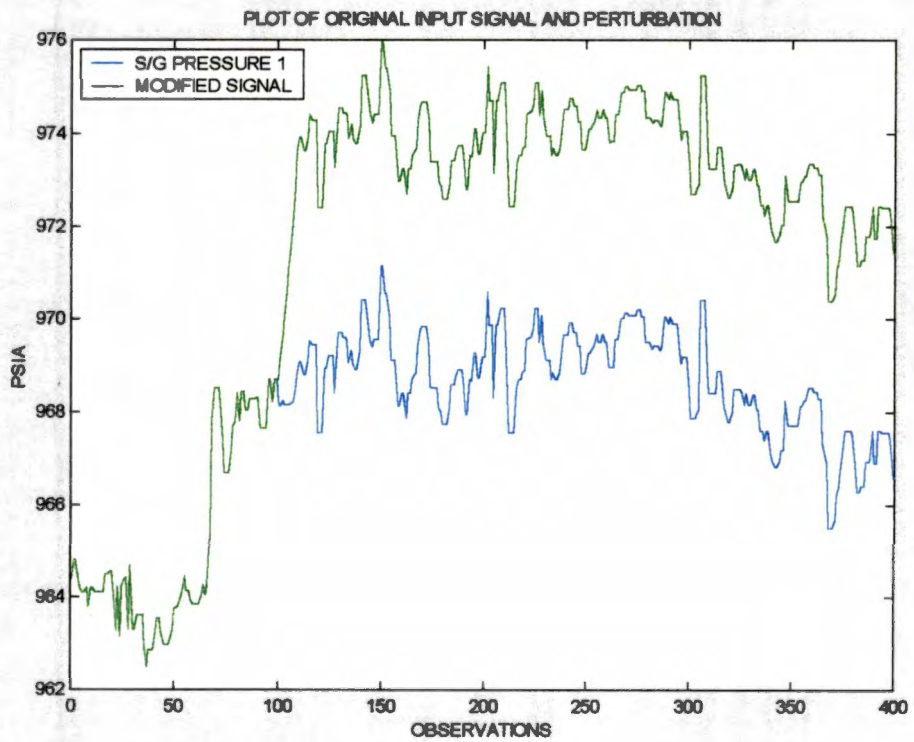

Figure A-52 Change in set point signal for a .5% perturbation

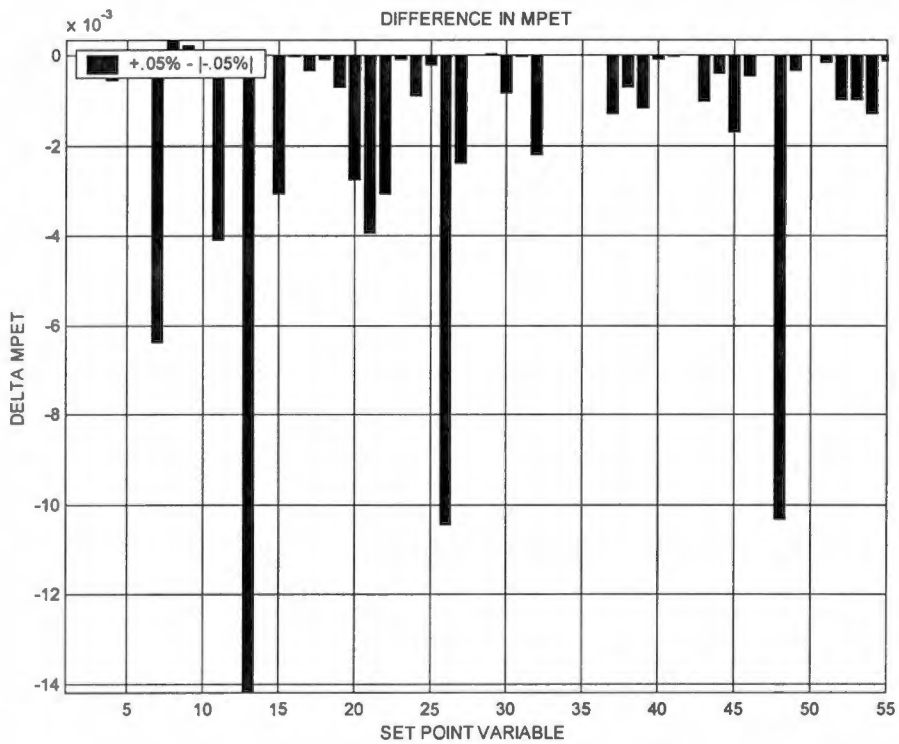

Figure A-53 Change in MPET for a +.05% and -.05% perturbation, MR-NN mappings, (per=3, nodes=4, loops=5, tansig-purelin, +.0005 & -.0005 pert, 21 SD inputs)

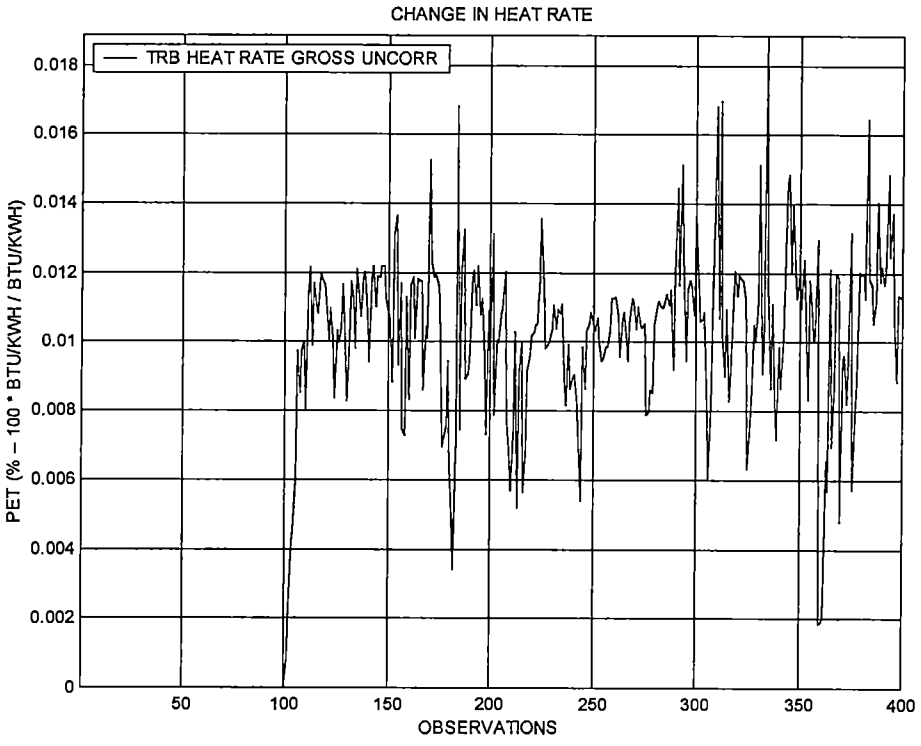

Figure A-54 Change in heat rate for a negative .05% change in input, MR-NN mappings, (per=3, nodes=4, loops=5, tansig-purelin, .0005 pert, epochs=300, 21 SD inputs)

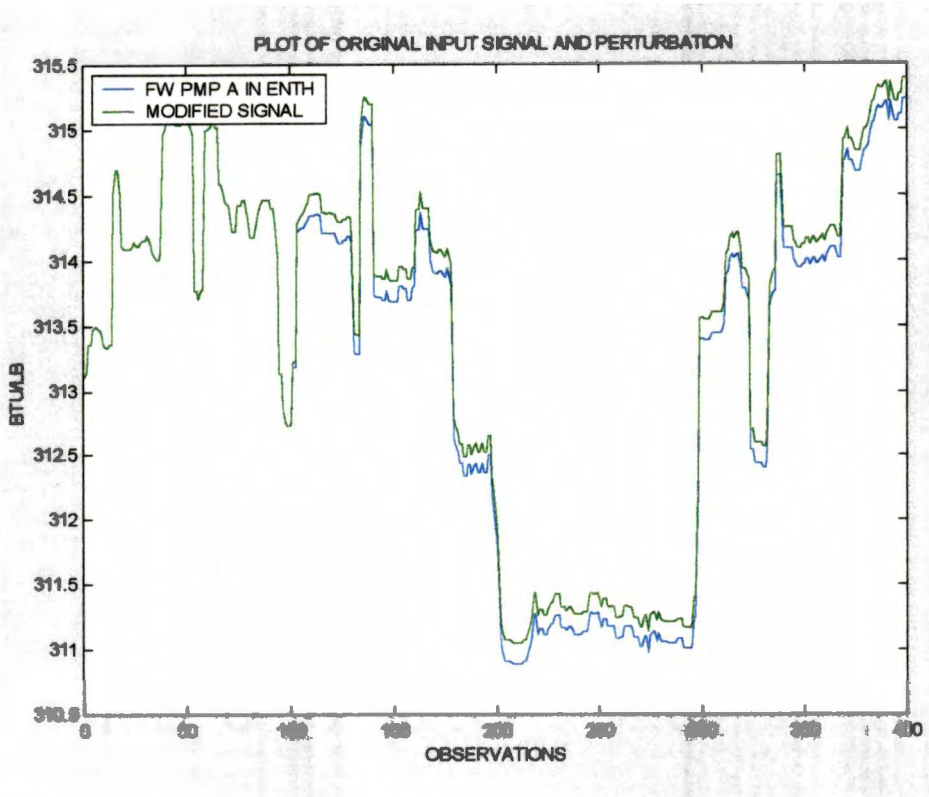

Figure A-55 Change in set point signal for a .05% perturbation

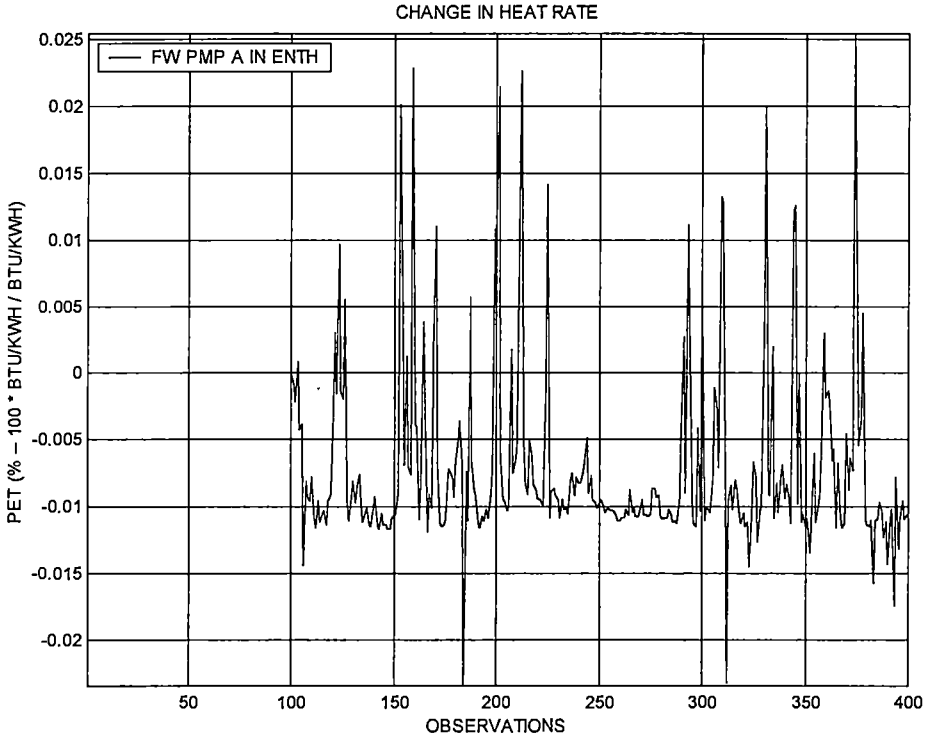

Figure A-56 Change in heat rate for a .05% change in input, MR-NN mappings, (per=3, nodes=4, loops=5, tansig-purelin, .0005 pert, epochs=300, 21 SD inputs)

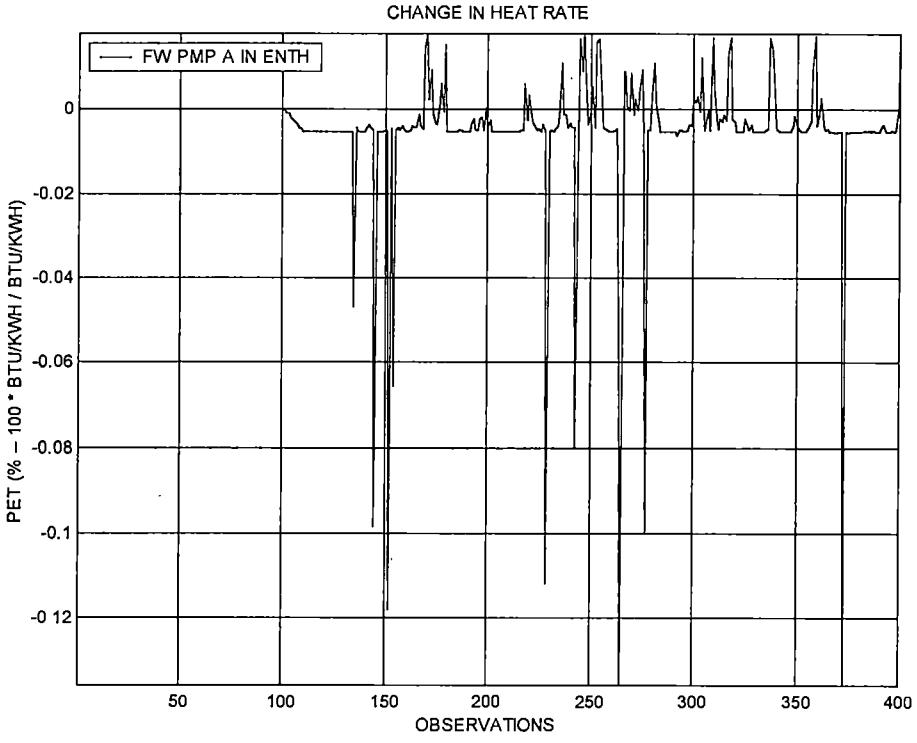

Figure A-57 Change in heat rate for a .05% change in input, MR-NN mappings, (no cross-validation, nodes=4, loops=5, tansig-purelin, .0005 pert, epochs=3000, 21 SD inputs)

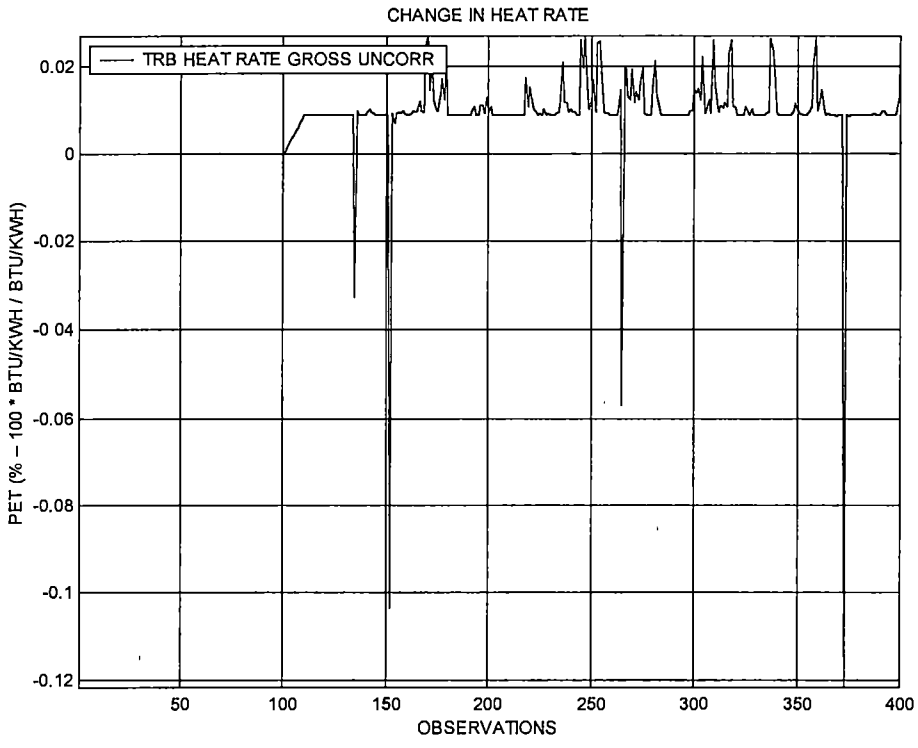

Figure A-58 Change in heat rate for a .05% change in input, MR-NN mappings, (no cross-validation, nodes=4, loops=5, tansig-purelin, .0005 pert, epochs=3000, 21 SD inputs)

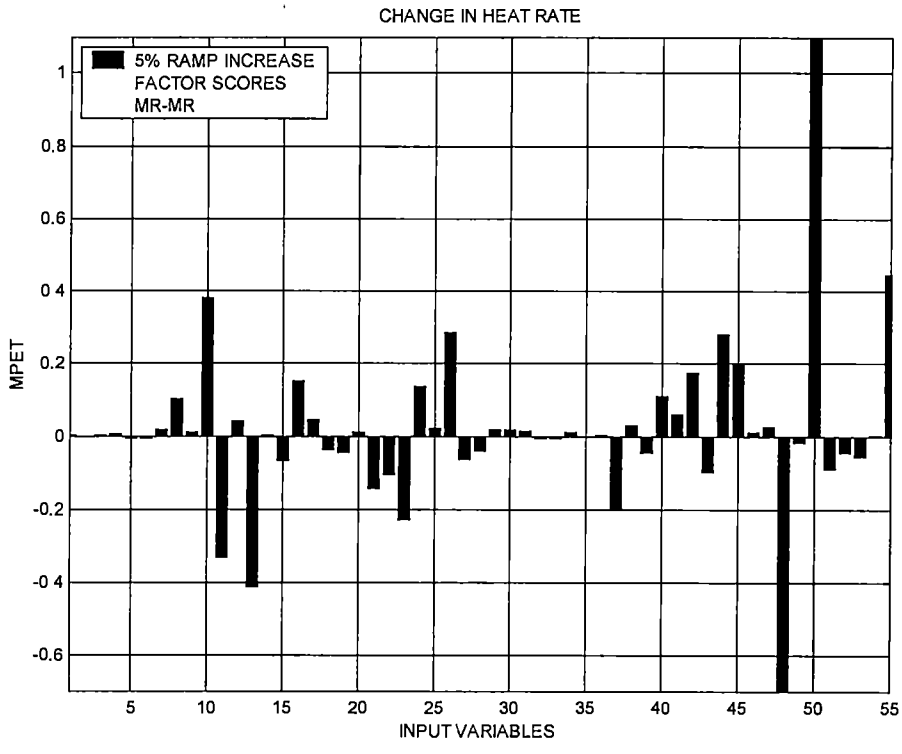

Figure A-59 Change in heat rate for a +5% perturbation in each setpoint variable (FACTOR Scores)

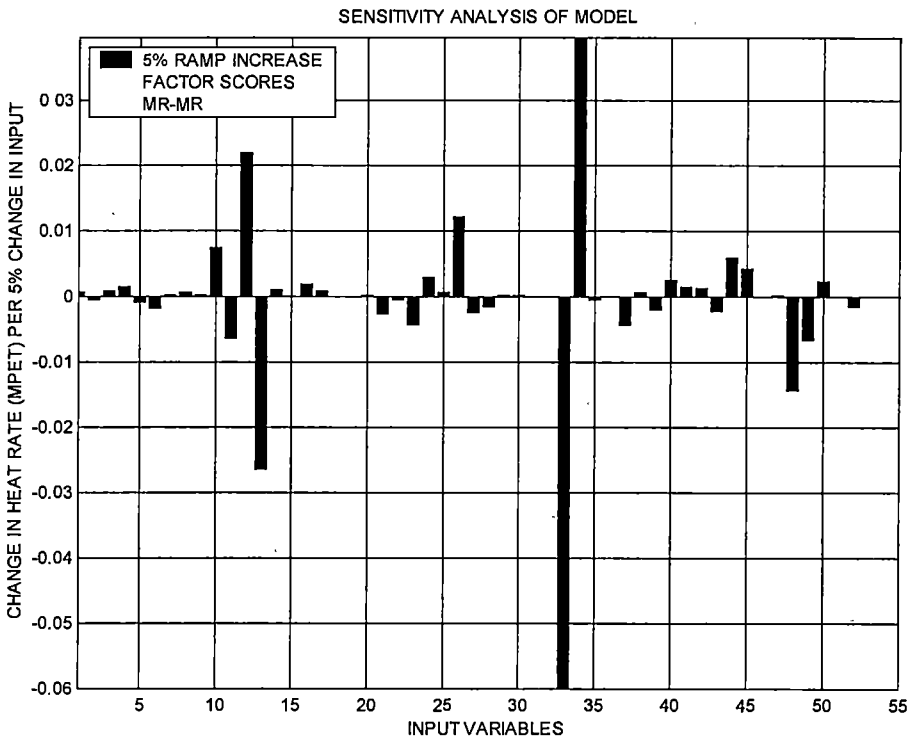

Figure A-60 Sensitivity plot showing change in heat rate per +5% perturbation in each setpoint variable (FACTOR Scores)

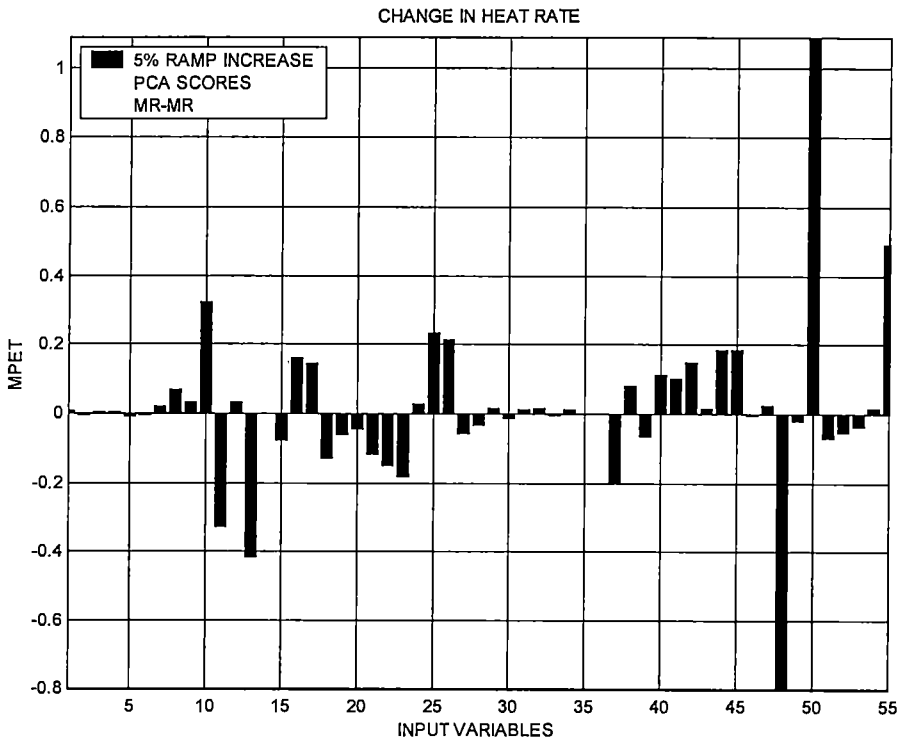

Figure A-61 Change in heat rate for a +5% perturbation in each setpoint variable (PCA Scores)

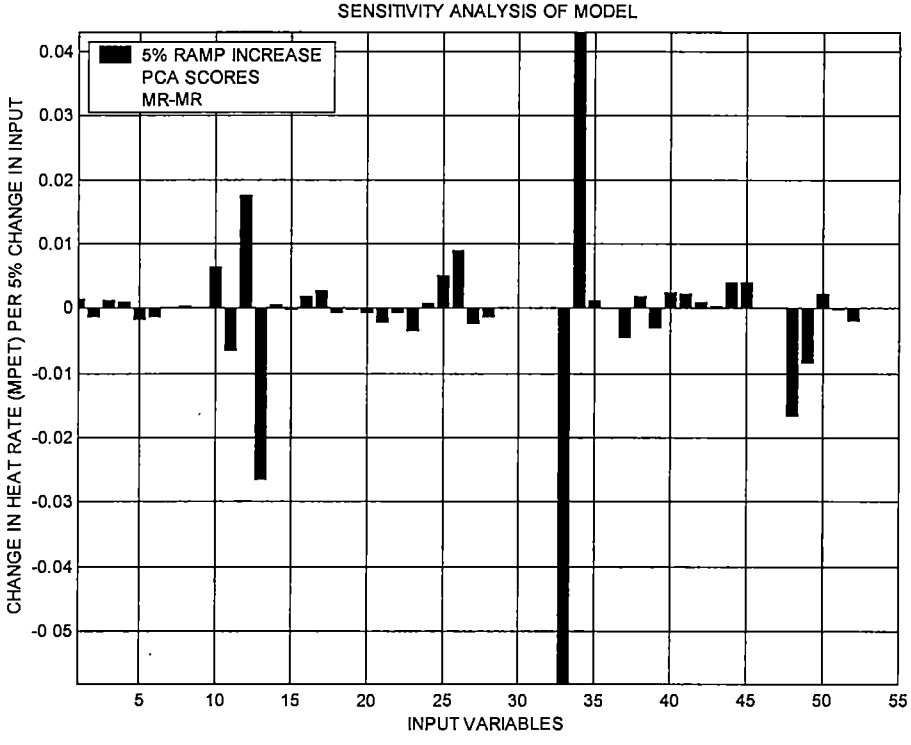

Figure A-62 Sensitivity plot showing change in heat rate per +5% perturbation in each setpoint variable (PCA Scores)

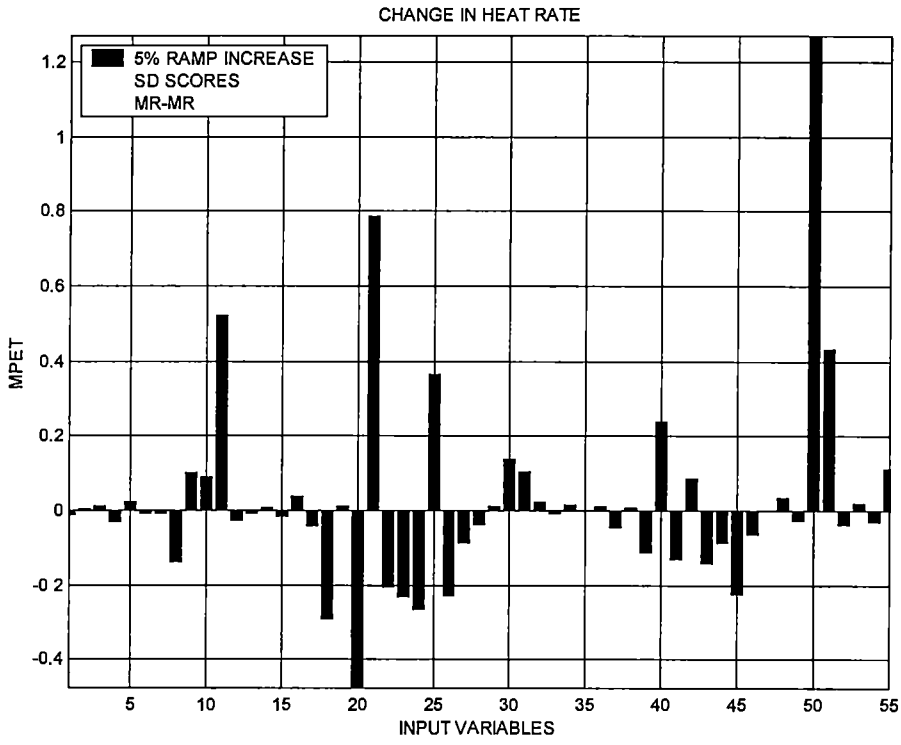

Figure A-63 Change in heat rate for a +5% perturbation in each setpoint variable (43 SD Scores)

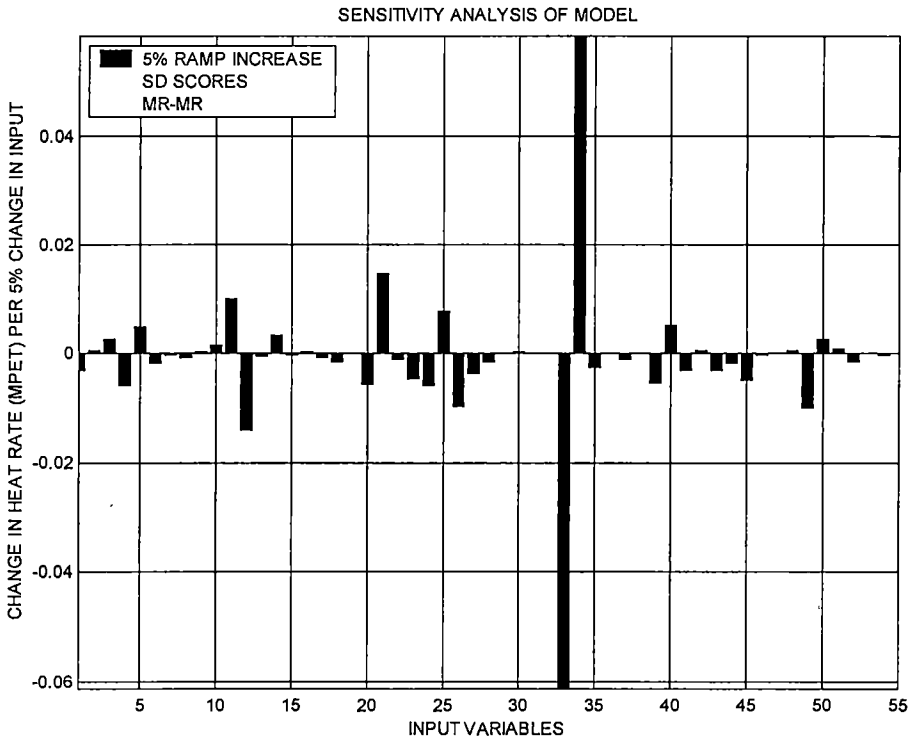

Figure A-64 Sensitivity plot showing change in heat rate per +5% perturbation in each setpoint variable (43 SD Scores)

Table A-4 Comparison of the ten most important setpoint variables affecting heat rate, for different state variable scores (MR-MR mappings)

FACTOR SCORES (5)	MPET	PCA SCORES (5)	MPET	43 SD SCORES	MPET
'TRB HEAT RATE GROSS UNCORR'	1.0976	'TRB HEAT RATE GROSS UNCORR'	1.0874	'TRB HEAT RATE GROSS UNCORR'	1.271
'S/G PRESSURE 1'	-0.6986	'S/G PRESSURE 1'	-0.80229	'FW PUMP(B) PRESSURE'	0.7849
'TOTAL STEAM FLOW'	0.4478	'TOTAL STEAM FLOW'	0.4906	'FEEDWATER PRESS 2'	0.51996
'FW PMP A IN ENTH'	-0.41301	'FW PMP A IN ENTH'	-0.41469	'FW PUMP(B) HEAD'	-0.4773
'FEEDWATER PRESS 1'	0.38239	'FEEDWATER PRESS 2'	-0.32566	'TRB HEAT RATE NET CORR'	0.43105
'FEEDWATER PRESS 2'	-0.33192	'FEEDWATER PRESS 1'	0.32166	'FWPT B HP STEAM INLET'	0.36474
'FWPT B HP SV CHEST TEMP'	0.28562	'FWPT B HP STEAM INLET'	0.23401	'FW PUMP(A) SPEED'	-0.29023
'AVG MAIN STEAM PRESSURE'	0.28371	'FWPT B HP SV CHEST TEMP'	0.21299	'FWPT A HP STEAM INLET'	-0.2661
'FW TO SG PRESS'	-0.22709	'MSR A 2ND STG RHTG STM PRESS'	-0.20102	'MSR C 2ND STG RHTG STM PRESS'	0.23809
'MSR A 2ND STG RHTG STM PRESS'	-0.19799	'MAIN STEAM PRESSURE (A)'	0.18502	'FW TO SG PRESS'	-0.22979

FIVE HIGHEST CORRELATED WITH HR	MPET	FIVE PRINCIPAL VARIABLES	MPET
'TRB HEAT RATE GROSS UNCORR'	1.3368	'TRB HEAT RATE GROSS UNCORR'	1.3368
'TRB HEAT RATE NET CORR'	0.54349	'TRB HEAT RATE NET CORR'	0.54349
'FEEDWATER PRESS 2'	0.48237	'FEEDWATER PRESS 2'	0.48237
'FW PUMP(B) PRESSURE'	0.42757	'FW PUMP(B) PRESSURE'	0.42757
'AVG MAIN STEAM PRESSURE'	-0.40068	'AVG MAIN STEAM PRESSURE'	-0.40068
'S/G PRESSURE 1'	0.36678	'S/G PRESSURE 1'	0.36678
'CONDENSATE PUMP(A) FLOW'	-0.32642	'CONDENSATE PUMP(A) FLOW'	-0.32642
'FW TO SG PRESS'	-0.32459	'FW TO SG PRESS'	-0.32459
'TURB STM CHEST PRESS'	-0.29683	'TURB STM CHEST PRESS'	-0.29683
'FW PUMP(A) SPEED'	-0.26739	'FW PUMP(A) SPEED'	-0.26739

Table A-5 Comparison of the ten most important setpoint variables affecting heat rate, for different state variable scores (PLS-PLS mappings)

FACTOR SCORES stages: MAP1-55 MAP2-2	MPET	PCA SCORES stages:MAP1-55 MAP2-4	MPET	ALL 43 STATE stages: MAP1-55 MAP2-3	MPET
'TRB HEAT RATE GROSS UNCORR'	1.1003	'TRB HEAT RATE GROSS UNCORR'	1.0949	'TRB HEAT RATE GROSS UNCORR'	1.0787
'S/G PRESSURE 1'	-0.69159	'S/G PRESSURE 1'	-0.79237	'S/G PRESSURE 1'	-0.97335
'TOTAL STEAM FLOW'	0.45215	'TOTAL STEAM FLOW'	0.50324	'FW PMP A IN ENTH'	-0.56038
'FW PMP A IN ENTH'	-0.41754	'FW PMP A IN ENTH'	-0.4302	'TOTAL STEAM FLOW'	0.42214
'FEEDWATER PRESS 1'	0.38276	'FEEDWATER PRESS 1'	0.33134	'FEEDWATER PRESS 1'	0.39156
'FEEDWATER PRESS 2'	-0.32969	'FEEDWATER PRESS 2'	-0.31556	'FWPT B HP STEAM INLET'	0.3761
'FWPT B HP SV CHEST TEMP'	0.2839	'FWPT B HP STEAM INLET'	0.21812	'FW PUMP(A) HEAD'	0.32266
'AVG MAIN STEAM PRESSURE'	0.28379	'FWPT B HP SV CHEST TEMP'	0.2101	'FW PUMP(B) HEAD'	-0.29647
'FW TO SG PRESS'	-0.22945	'MSR A 2ND STG RHTG STM PRESS'	-0.20009	'AVG MAIN STEAM PRESSURE'	0.22251
'MSR A 2ND STG RHTG STM PRESS'	-0.19861	'FW TO SG PRESS'	-0.19228	'MSR C 2ND STG RHTG STM PRESS'	0.20086

FACTOR SCORES stages: MAP1-55 MAP2-5	MPET	PCA SCORES stages:MAP1-55 MAP2-5	MPET	ALL 43 STATE stages: MAP1-55 MAP2-15	MPET
'TRB HEAT RATE GROSS UNCORR'	1.0976	'TRB HEAT RATE GROSS UNCORR'	1.0874	'TRB HEAT RATE GROSS UNCORR'	1.2318
'S/G PRESSURE 1'	-0.6986	'S/G PRESSURE 1'	-0.80229	'FW PUMP(B) PRESSURE'	0.76732
'TOTAL STEAM FLOW'	0.4478	'TOTAL STEAM FLOW'	0.4906	'FEEDWATER PRESS 2'	0.68212
'FW PMP A IN ENTH'	-0.41301	'FW PMP A IN ENTH'	-0.41469	'TRB HEAT RATE NET CORR'	0.34517
'FEEDWATER PRESS 1'	0.38239	'FEEDWATER PRESS 2'	-0.32566	'TOTAL STEAM FLOW'	0.33437
'FEEDWATER PRESS 2'	-0.33192	'FEEDWATER PRESS 1'	0.32166	'MAIN STEAM PRESSURE (A)'	-0.31534
'FWPT B HP SV CHEST TEMP'	0.28562	'FWPT B HP STEAM INLET'	0.23401	'FW PUMP(A) SPEED'	-0.30661
'AVG MAIN STEAM PRESSURE'	0.28371	'FWPT B HP SV CHEST TEMP'	0.21299	'FW PUMP(B) HEAD'	-0.29681
'FW TO SG PRESS'	-0.22709	'MSR A 2ND STG RHTG STM PRESS'	-0.20102	'FW PUMP(B) SPEED'	-0.29096
'MSR A 2ND STG RHTG STM PRESS'	-0.19799	'MAIN STEAM PRESSURE (A)'	0.18502	'CONDENSATE PUMP(C) FLOW'	0.23006

Table A-6 Comparison of the ten most important setpoint variables affecting heat rate, for PCA state variable scores (MR-MR mappings)

PCA SCORES (5)	MPET	PCA SCORES (6)	MPET	PCA SCORES (7)	MPET
'TRB HEAT RATE GROSS UNCORR'	1.0874	TRB HEAT RATE GROSS UNCORR'	1.0686	'S/G PRESSURE 1'	-0.9972
'S/G PRESSURE 1'	-0.80229	'S/G PRESSURE 1'	-0.99976	'TRB HEAT RATE GROSS UNCORR'	0.98206
'TOTAL STEAM FLOW'	0.4906	'TOTAL STEAM FLOW'	0.4819	'FW PMP A IN ENTH'	-0.52218
'FW PMP A IN ENTH'	-0.41469	'FEEDWATER PRESS 1'	0.40179	'FEEDWATER PRESS 1'	0.45934
'FEEDWATER PRESS 2'	-0.32566	'FW PMP A IN ENTH'	-0.39016	'TOTAL STEAM FLOW'	0.36898
'FEEDWATER PRESS 1'	0.32166	'FW PUMP(A) HEAD'	0.29832	'AVG MAIN STEAM PRESSURE'	0.30806
'FWPT B HP STEAM INLET'	0.23401	'AVG MAIN STEAM PRESSURE'	0.27865	'FW PUMP(A) HEAD'	0.30282
'FWPT B HP SV CHEST TEMP'	0.21299	'FEEDWATER PRESS 2'	-0.27179	'FW PUMP(A) SPEED'	-0.28837
'MSR A 2ND STG RHTG STM PRESS'	-0.20102	'MAIN STEAM PRESSURE (A)'	0.26021	'FW TO SG PRESS'	-0.25151
'MAIN STEAM PRESSURE (A)'	0.18502	'FWPT B HP SV CHEST TEMP'	0.25957	'CONDENSATE PUMP(B) FLOW'	0.23419

PCA SCORES (8)	MPET	PCA SCORES (9)	MPET	PCA SCORES (10)	MPET
'S/G PRESSURE 1'	-1.0399	'S/G PRESSURE 1'	-1.0315	'TRB HEAT RATE GROSS UNCORR'	0.9606
'TRB HEAT RATE GROSS UNCORR'	0.97282	'TRB HEAT RATE GROSS UNCORR'	0.97327	'S/G PRESSURE 1'	-0.95708
'FW PMP A IN ENTH'	-0.70622	'FW PMP A IN ENTH'	-0.71418	'FW PMP A IN ENTH'	-0.68727
'TOTAL STEAM FLOW'	0.48363	'FWPT B HP STEAM INLET'	0.54036	'FWPT B HP STEAM INLET'	0.53535
'FEEDWATER PRESS 1'	0.46958	'TOTAL STEAM FLOW'	0.48659	'TOTAL STEAM FLOW'	0.51204
'FWPT B HP STEAM INLET'	0.43585	'FEEDWATER PRESS 1'	0.46827	'FEEDWATER PRESS 1'	0.36664
'FW PUMP(A) HEAD'	0.32757	'FW PUMP(A) HEAD'	0.32505	'FW PUMP(B) HEAD'	-0.33664
'FW PUMP(B) HEAD'	-0.2881	'FW PUMP(B) HEAD'	-0.28428	'FWPT A HP STEAM INLET'	-0.32988
'MSR C 2ND STG RHTG STM PRESS'	0.22566	'FW PUMP(A) SPEED'	-0.23561	'FW PUMP(A) HEAD'	0.30405
'FW PUMP(A) SPEED'	-0.21386	'MSR C 2ND STG RHTG STM PRESS'	0.22228	'AVG MAIN STEAM PRESSURE'	0.21499

Table A-7 Comparison of the ten most important setpoint variables affecting heat rate, for FACTOR state variable scores (MR-MR mappings)

FACTOR SCORES (5)	MPET	FACTOR SCORES (6)	MPET
'TRB HEAT RATE GROSS UNCORR'	1.0976	'TRB HEAT RATE GROSS UNCORR'	1.0921
'S/G PRESSURE 1'	-0.6986	'S/G PRESSURE 1'	-0.77226
'TOTAL STEAM FLOW'	0.4478	'TOTAL STEAM FLOW'	0.46271
'FW PMP A IN ENTH'	-0.41301	'FEEDWATER PRESS 1'	0.42015
'FEEDWATER PRESS 1'	0.38239	'FW PMP A IN ENTH'	-0.41526
'FEEDWATER PRESS 2'	-0.33192	'AVG MAIN STEAM PRESSURE'	0.32977
'FWPT B HP SV CHEST TEMP'	0.28562	'FWPT B HP SV CHEST TEMP'	0.31955
'AVG MAIN STEAM PRESSURE'	0.28371	'FEEDWATER PRESS 2'	-0.31768
'FW TO SG PRESS'	-0.22709	'FW TO SG PRESS'	-0.23657
'MSR A 2ND STG RHTG STM PRESS'	-0.19799	'MAIN STEAM PRESSURE (A)'	0.23579

Table A-8 Comparison of the ten most important setpoint variables affecting heat rate, for different state variable scores (MR-MR mappings), (original 43 state variables reduced to 21 by removing highly collinear variables)

PCA SCORES (5)	MPET	PCA SCORES (6)	MPET	PCA SCORES (7)	MPET
'FW PMP A IN ENTH'	-0.99392	'FW PMP A IN ENTH'	-0.96823	'FW PMP A IN ENTH'	-1.0201
'TRB HEAT RATE GROSS UNCORR'	0.78133	'S/G PRESSURE 1'	-0.82912	'S/G PRESSURE 1'	-0.87279
'S/G PRESSURE 1'	-0.77291	'TRB HEAT RATE GROSS UNCORR'	0.76337	'TRB HEAT RATE GROSS UNCORR'	0.7727
'TOTAL STEAM FLOW'	0.70022	'TOTAL STEAM FLOW'	0.661	'TOTAL STEAM FLOW'	0.72089
'FW PUMP(B) HEAD'	-0.4546	'FW PUMP(B) HEAD'	-0.46491	'FWPT B HP STEAM INLET'	0.57919
'FWPT B HP STEAM INLET'	0.40574	'FEEDWATER PRESS 1'	0.37065	'FW PUMP(B) HEAD'	-0.50096
'FEEDWATER PRESS 1'	0.33483	'FW PUMP(A) HEAD'	0.36416	'FW PUMP(A) HEAD'	0.39585
'FW PUMP(A) HEAD'	0.31297	'FWPT B HP STEAM INLET'	0.36131	'FEEDWATER PRESS 1'	0.38537
'MSR A 2ND STG RHTG STM PRESS'	-0.25974	'FW PUMP(A) SPEED'	-0.30398	'FW PUMP(A) SPEED'	-0.29947
'FW PUMP(A) SPEED'	-0.24613	'TURB STM CHEST PRESS'	0.23938	'MSR D 2ND STG RHTG STM PRESS'	0.24641

PCA SCORES (8)	MPET	FACTOR SCORES (5)	MPET	21 SD SCORES	MPET
'FW PMP A IN ENTH'	-0.949	'FW PMP A IN ENTH'	-1.0402	'TRB HEAT RATE GROSS UNCORR'	1.0728
'S/G PRESSURE 1'	-0.7941	'S/G PRESSURE 1'	-0.83853	'S/G PRESSURE 1'	-0.71567
'TRB HEAT RATE GROSS UNCORR'	0.78554	'TRB HEAT RATE GROSS UNCORR'	0.70088	'FW PMP A IN ENTH'	-0.59335
'TOTAL STEAM FLOW'	0.74483	'FEEDWATER PRESS 1'	0.52629	'FWPT B HP SV CHEST TEMP'	-0.48409
'FWPT B HP STEAM INLET'	0.61061	'TOTAL STEAM FLOW'	0.44977	'MSR C 2ND STG RHTG STM PRESS'	0.46162
'FW PUMP(B) HEAD'	-0.4789	'FW PUMP(B) HEAD'	-0.44435	'CONDENSATE PUMP(C) FLOW'	0.29791
'FW PUMP(A) HEAD'	0.3538	'FW PUMP(A) HEAD'	0.39524	'FW PUMP(B) PRESSURE'	-0.29503
'FW PUMP(A) SPEED'	-0.30691	'FW TO SG PRESS'	-0.31934	'TRB HEAT RATE NET CORR'	0.28559
'FWPT A HP STEAM INLET'	-0.30329	'MAIN STEAM PRESSURE (A)'	-0.28617	'AVG MAIN STEAM PRESSURE'	0.26848
'FEEDWATER PRESS 1'	0.2974	'FW PUMP(A) SPEED'	-0.28559	'TOTAL STEAM FLOW'	0.23953

Table A-9 Comparison of the ten most important setpoint variables affecting heat rate, for different state variable scores and mappings (original 43 state variables reduced to 21 by removing highly collinear variables)

PLS-PLS (55-13) Coeff = .001 Mappings 21 st	MPET	PLS-PLS (55-20) DeltaCoeff >= 1e-6 Mappings 21 st	MPET	MR-MR Mappings 21 SD SCORES	MPET
'TRB HEAT RATE GROSS UNCORR'	1.065	'TRB HEAT RATE GROSS UNCORR'	1.0716	'TRB HEAT RATE GROSS UNCORR'	1.0728
'S/G PRESSURE 1'	-0.73377	'S/G PRESSURE 1'	-0.7213	'S/G PRESSURE 1'	-0.71567
'FW PMP A IN ENTH'	-0.59566	'FW PMP A IN ENTH'	-0.59107	'FW PMP A IN ENTH'	-0.59335
'FWPT B HP SV CHEST TEMP'	-0.49005	'FWPT B HP SV CHEST TEMP'	-0.48718	'FWPT B HP SV CHEST TEMP'	-0.48409
'MSR C 2ND STG RHTG STM PRESS'	0.46635	'MSR C 2ND STG RHTG STM PRESS'	0.46163	'MSR C 2ND STG RHTG STM PRESS'	0.46162
'CONDENSATE PUMP(C) FLOW'	0.30384	'FW PUMP(B) PRESSURE'	-0.29885	'CONDENSATE PUMP(C) FLOW'	0.29791
'TRB HEAT RATE NET CORR'	0.29602	'CONDENSATE PUMP(C) FLOW'	0.29837	'FW PUMP(B) PRESSURE'	-0.29503
'FW PUMP(B) PRESSURE'	-0.29433	'TRB HEAT RATE NET CORR'	0.28476	'TRB HEAT RATE NET CORR'	0.28559
'AVG MAIN STEAM PRESSURE'	0.26017	'AVG MAIN STEAM PRESSURE'	0.26752	'AVG MAIN STEAM PRESSURE'	0.26848
'TOTAL STEAM FLOW'	0.22991	'TOTAL STEAM FLOW'	0.24558	'TOTAL STEAM FLOW'	0.23953

PLS - 21 PCA SCORES MAP2-MR	MPET	PLS - 12 Arbitrary Stage Variables MAP2-MR	MPET
'TRB HEAT RATE GROSS UNCORR'	1.0742	'TRB HEAT RATE GROSS UNCORR'	1.0386
'S/G PRESSURE 1'	-0.55497	'S/G PRESSURE 1'	-0.80014
'FW PMP A IN ENTH'	-0.51296	'FW PMP A IN ENTH'	-0.5363
'MSR C 2ND STG RHTG STM PRESS'	0.45027	'AVG MAIN STEAM PRESSURE'	0.48977
'FWPT B HP SV CHEST TEMP'	-0.42801	'FEEDWATER PRESS 1'	0.40591
'TRB HEAT RATE NET CORR'	0.40843	'FWPT B HP STEAM INLET'	-0.38402
'AVG MAIN STEAM PRESSURE'	0.34353	'FWPT B HP SV CHEST TEMP'	-0.38241
'CONDENSATE PUMP(C) FLOW'	0.32082	'CONDENSATE PUMP(C) FLOW'	0.28589
'FW PUMP(B) SPEED'	-0.22154	'FW PUMP(B) SPEED'	-0.28058
'TOTAL STEAM FLOW'	0.20634	'FW PUMP(A) SPEED'	-0.26005

Table A-10 Comparison of the ten most important setpoint variables affecting heat rate, for 21 SD state variable scores (MR-NN mappings), top row--MPET, bottom row--median PET

MR-NN Mappings 21 SD .005 change in input	MPET	MR-NN Mappings 21 SD .0005 change in input	MPET
'TRB HEAT RATE GROSS UNCORR'	0.081109	'TRB HEAT RATE GROSS UNCORR'	0.00986
'FW PMP A IN ENTH'	-0.05557	'FEEDWATER PRESS 1'	0.006029
'TOTAL STEAM FLOW'	0.048784	'FW PMP A IN ENTH'	-0.00574
'S/G PRESSURE 1'	-0.03576	'MSR C 2ND STG RHTG STM PRESS'	0.004401
'FWPT B HP SV CHEST TEMP'	-0.03534	'AVG MAIN STEAM PRESSURE'	0.003726
'FWPT A HP STEAM INLET'	-0.03374	'HTR DRAIN PUMP(B) FLOW'	0.003454
'FW PUMP(B) PRESSURE'	-0.02605	'CONDENSATE PUMP(C) FLOW'	0.003252
'MSR A 2ND STG RHTG STM PRESS'	-0.02293	'FW PUMP(B) SPEED'	-0.00294
'FW PUMP(A) PRESSURE'	0.022194	'TOTAL STEAM FLOW'	0.002765
'CONDENSATE PUMP(B) FLOW'	-0.02129	'FW PUMP(B) PRESSURE'	0.00274

MR-NN Mappings 21 SD .0005 change in input	Median PET
'FW PUMP(B) PRESSURE'	0.014365
'FW PUMP(A) PRESSURE'	-0.011334
'FWPT B HP STEAM INLET'	-0.010799
'FW PUMP(B) HEAD'	-0.01069
'MSR A 2ND STG RHTG STM PRESS'	0.010364
'TRB HEAT RATE GROSS UNCORR'	0.0091177
'FEEDWATER PRESS 1'	0.0089781
'CONDENSATE PUMP(A) FLOW'	0.0083636
'FW PUMP(A) HEAD'	0.0072266
'S/G PRESSURE 1'	-0.0067053

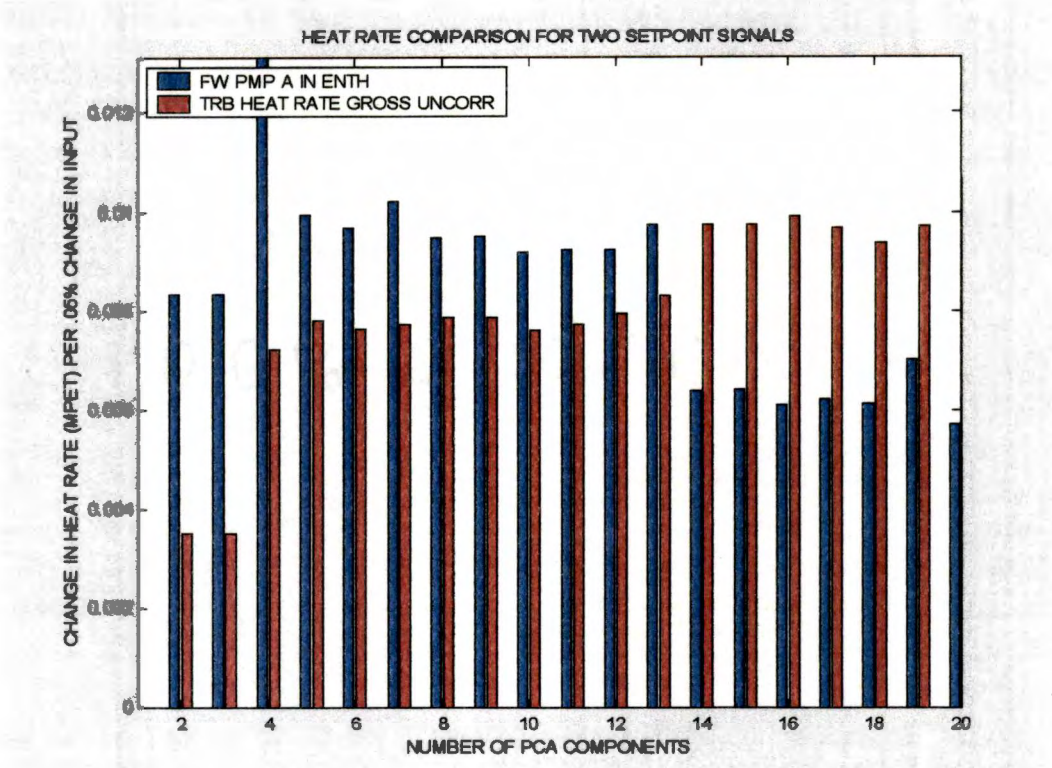

Figure A-65 Change in heat rate as a function of number of state variable PCA components, set point variable -- TRB HEAT RATE GROSS UNCORR and FW PMP A IN ENTH, MR-MR mappings, 21 state matrix, pert = .0005, (all values shown are absolute)

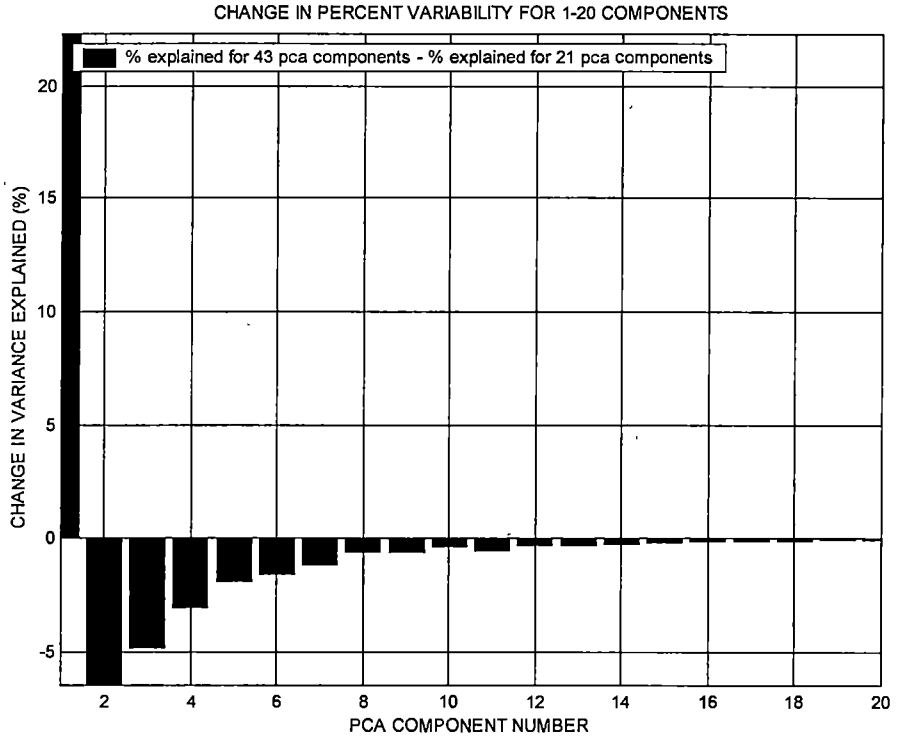

Figure A-66 Change in variance explained for state variables, 20 major PCA components (rank 43) and 20 major PCA components (rank 21)

Table A-11 Set Point variable descriptions

Index	Set Point Variables
1	'CNDR A CLEAN FACTOR 1'
2	'CNDR A CLEAN FACTOR 2'
3	'CNDR B CLEAN FACTOR 1'
4	'CNDR B CLEAN FACTOR 2'
5	'CNDR C CLEAN FACTOR 1'
6	'CNDR C CLEAN FACTOR 2'
7	'CONDENSATE PUMP(A) FLOW'
8	'CONDENSATE PUMP(B) FLOW'
9	'CONDENSATE PUMP(C) FLOW'
10	'FEEDWATER PRESS 1'
11	'FEEDWATER PRESS 2'
12	'FP TURB(A) STEAM FLOW'
13	'FW PMP A IN ENTH'
14	'FW PMP A SUCT FLOW DP'
15	'FW PUMP(A) FLOW'
16	'FW PUMP(A) HEAD'
17	'FW PUMP(A) PRESSURE'
18	'FW PUMP(A) SPEED'
19	'FW PUMP(B) FLOW'
20	'FW PUMP(B) HEAD'
21	'FW PUMP(B) PRESSURE'
22	'FW PUMP(B) SPEED'
23	'FW TO SG PRESS'
24	'FWPT A HP STEAM INLET'
25	'FWPT B HP STEAM INLET'
26	'FWPT B HP SV CHEST TEMP'
27	'DRN PUMP(A) ABS DIS PRES'
28	'DRN PUMP(B) ABS DIS PRES'
29	'HTR DRAIN PUMP(A) FLOW'
30	'HTR DRAIN PUMP(B) FLOW'
31	'HTR DRN PUMP(A) DEL PRES'
32	'HTR DRN PUMP(B) DEL PRES'
33	'HTR NO 1B TERM TEMP DIFF'
34	'HTR NO 2B TERM TEMP DIFF'
35	'HTR NO 2C TERM TEMP DIFF'
36	'TOTAL DRAIN & COND FLOW'
37	'MSR A 2ND STG RHTG STM PRESS'
38	'MSR B 2ND STG RHTG STM PRESS'
39	'MSR C 1ST STG RHTG STM PRESS'
40	'MSR C 2ND STG RHTG STM PRESS'
41	'MSR D 2ND STG RHTG STM PRESS'
42	'RH A 2ND STG STM OUT FLW'
43	'TURB STM CHEST PRESS'
44	'AVG MAIN STEAM PRESSURE'
45	'MAIN STEAM PRESSURE (A)'
46	'S/G 1 ENERGY REMOVAL RATE'
47	'S/G 2 ENERGY REMOVAL RATE'
48	'S/G PRESSURE 1'
49	'SG 1 LEVEL'
50	'TRB HEAT RATE GROSS UNCORR'
51	'TRB HEAT RATE NET CORR'
52	'TURB DIFF EXPANSION'
53	'TOTAL FEED FLOW'
54	'TOTAL FEEDWATER FLOW'
55	'TOTAL STEAM FLOW'

Table A-12 List of Set Point Variables that were removed (shown in 2nd column) because of high collinearity (>.95) with other set point variables (shown in 1st column)

'CNDR A TTD 1'	'FW PUMP(A) SUCTION PRESS'
'CNDR A TTD 2'	'FW PUMP(B) DISCHARGE TEM'
'CNDR B CLEAN FACTOR 1'	'FW PUMP(B) FLOW'
'CNDR B CW TMP RISE 1'	'CNDR B TTD 1'
'CNDR B CW TMP RISE 1'	'CNDR B TTD 2'
'CNDR B CW TMP RISE 1'	'CNDR C CW TMP RISE 1'
'CNDR B CW TMP RISE 1'	'FP TURB(B) EXH PRESS'
'CNDR B CW TMP RISE 1'	'FP TURB(B) STEAM FLOW'
'CNDR B CW TMP RISE 1'	'FW PMP A FW SP VOL'
'CNDR B TTD 1'	'CNDR B TTD 2'
'CNDR B TTD 1'	'CNDR C CW TMP RISE 1'
'CNDR B TTD 1'	'FP TURB(B) EXH PRESS'
'CNDR B TTD 1'	'FP TURB(B) STEAM FLOW'
'CNDR B TTD 1'	'FW PMP A FW SP VOL'
'CNDR B TTD 2'	'CNDR C CW TMP RISE 1'
'CNDR B TTD 2'	'FP TURB(B) EXH PRESS'
'CNDR B TTD 2'	'FP TURB(B) STEAM FLOW'
'CNDR B TTD 2'	'FW PMP A FW SP VOL'
'CNDR C CLEAN FACTOR 1'	'FW PUMP(B) HEAD'
'CNDR C CW TMP RISE 1'	'FP TURB(B) EXH PRESS'
'CNDR C CW TMP RISE 1'	'FP TURB(B) STEAM FLOW'
'CNDR C CW TMP RISE 1'	'FW PMP A FW SP VOL'
'CNDR C CW TMP RISE 2'	'FW PUMP(B) PRESSURE'
'CNDR HW LVL 1C'	'FW PMP A OUT ENTH'
'CNDSR (C) C/D CW DIFF TMP'	'FW PUMP(B) SPEED'
'CW C/D DIFF TEMP'	'FEEDWATER PRESS 1'
'CW C/D DIFF TEMP'	'FEEDWATER PRESS 2'
'CW C/D DIFF TEMP'	'FP TURB(A) EXH PRESS'
'CW C/D DIFF TEMP'	'FP TURB(A) STEAM FLOW'
'FEEDWATER PRESS 1'	'FEEDWATER PRESS 2'
'FEEDWATER PRESS 1'	'FP TURB(A) EXH PRESS'
'FEEDWATER PRESS 1'	'FP TURB(A) STEAM FLOW'
'FEEDWATER PRESS 2'	'FP TURB(A) EXH PRESS'
'FEEDWATER PRESS 2'	'FP TURB(A) STEAM FLOW'
'FP TURB(A) EXH PRESS'	'FP TURB(A) STEAM FLOW'
'FP TURB(B) EXH PRESS'	'FP TURB(B) STEAM FLOW'
'FP TURB(B) EXH PRESS'	'FW PMP A FW SP VOL'
'FP TURB(B) STEAM FLOW'	'FW PMP A FW SP VOL'
'FW PMP A IN ENTH'	'FW PMP A OUT ENTH'
'FW PMP A IN ENTR'	'FW PMP A OUT ENTH'
'FW PMP A OUT ENTH'	'FW PMP B SUCT FLOW DP'
'FW PMP A SUCT FLOW DP'	'FW PMP B FW TMP RISE'
'FW PMP B IN ENTH'	'FW PMP B IN ENTR'
'FW PUMP(A) FLOW'	'FW PUMP(A) PRESSURE'

Table A-13 List of State Variables that were removed (shown in 2nd column) because of high collinearity (>.95) with other state variables (shown in 1st column)

AVG CONDENSER PRESSURE'	'Avg. CNDSR(A)IN CW TEMP L A/B'
'AVG CONDENSER PRESSURE'	'Avg. CNDSR(A)IN CW TEMP L C/D'
'AVG CONDENSER PRESSURE'	'Avg. CNDSR(B)IN CW TEMP L A/B'
'AVG CONDENSER PRESSURE'	'Avg. CNDSR(B)IN CW TEMP L C/D'
'AVG CONDENSER PRESSURE'	'Avg. CNDSR(C) OUT CW TEMP A/B'
'AVG CONDENSER PRESSURE'	'Avg. CNDSR(C) OUT CW TEMP C/D'
'AVG CONDENSER PRESSURE'	'Avg. CNDSR(C)IN CW TEMP L A/B'
'AVG CONDENSER PRESSURE'	'Avg. CNDSR(C)IN CW TEMP L C/D'
'AVG CONDENSER PRESSURE'	'CONDENSER(A) PRESSURE'
'AVG CONDENSER PRESSURE'	'CONDENSER(B) PRESSURE'
'AVG CONDENSER PRESSURE'	'CONDENSER(C) PRESSURE'
'AVG CONDENSER PRESSURE'	'HTR 1A DRAIN TEMP'
'AVG CONDENSER PRESSURE'	'HTR 1B DRAIN TEMP'
'AVG CONDENSER PRESSURE'	'HTR 1C DRAIN TEMP'
'AVG CONDENSER PRESSURE'	'HTR 2A DRAIN TEMP'
'AVG CONDENSER PRESSURE'	'HTR 2B DRAIN TEMP'
'AVG CONDENSER PRESSURE'	'HTR 2C DRAIN TEMP'
'AVG CONDENSER PRESSURE'	'TRB HEAT RATE GROSS DESIGN'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'Avg. CNDSR(A)IN CW TEMP L C/D'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'Avg. CNDSR(B)IN CW TEMP L A/B'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'Avg. CNDSR(B)IN CW TEMP L C/D'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'Avg. CNDSR(C) OUT CW TEMP A/B'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'Avg. CNDSR(C) OUT CW TEMP C/D'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'Avg. CNDSR(C)IN CW TEMP L A/B'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'Avg. CNDSR(C)IN CW TEMP L C/D'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'CONDENSER(A) PRESSURE'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'CONDENSER(B) PRESSURE'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'CONDENSER(C) PRESSURE'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'HTR 1A DRAIN TEMP'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'HTR 1B DRAIN TEMP'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'HTR 1C DRAIN TEMP'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'HTR 2A DRAIN TEMP'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'HTR 2B DRAIN TEMP'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'HTR 2C DRAIN TEMP'
'Avg. CNDSR(A)IN CW TEMP L A/B'	'TRB HEAT RATE GROSS DESIGN'
'Avg. CNDSR(A)IN CW TEMP L C/D'	'Avg. CNDSR(B)IN CW TEMP L A/B'
'Avg. CNDSR(A)IN CW TEMP L C/D'	'Avg. CNDSR(B)IN CW TEMP L C/D'
'Avg. CNDSR(A)IN CW TEMP L C/D'	'Avg. CNDSR(C) OUT CW TEMP A/B'
'Avg. CNDSR(A)IN CW TEMP L C/D'	'Avg. CNDSR(C) OUT CW TEMP C/D'
'Avg. CNDSR(A)IN CW TEMP L C/D'	'Avg. CNDSR(C)IN CW TEMP L A/B'
'Avg. CNDSR(A)IN CW TEMP L C/D'	'Avg. CNDSR(C)IN CW TEMP L C/D'
'Avg. CNDSR(A)IN CW TEMP L C/D'	'CONDENSER(A) PRESSURE'
'Avg. CNDSR(A)IN CW TEMP L C/D'	'CONDENSER(B) PRESSURE'
'Avg. CNDSR(A)IN CW TEMP L C/D'	'CONDENSER(C) PRESSURE'
'Avg. CNDSR(A)IN CW TEMP L C/D'	'HTR 1A DRAIN TEMP'
'Avg. CNDSR(A)IN CW TEMP L C/D'	'HTR 1B DRAIN TEMP'

'Avg. CNDSR(A)IN CW TEMP L C/D'	'HTR 1C DRAIN TEMP'
'Avg. CNDSR(A)IN CW TEMP L C/D'	'HTR 2A DRAIN TEMP'
'Avg. CNDSR(A)IN CW TEMP L C/D'	'HTR 2B DRAIN TEMP'
'Avg. CNDSR(A)IN CW TEMP L C/D'	'HTR 2C DRAIN TEMP'
'Avg. CNDSR(A)IN CW TEMP L C/D'	'TRB HEAT RATE GROSS DESIGN'
'Avg. CNDSR(B)IN CW TEMP L A/B'	'Avg. CNDSR(B)IN CW TEMP L C/D'
'Avg. CNDSR(B)IN CW TEMP L A/B'	'Avg. CNDSR(C) OUT CW TEMP A/B'
'Avg. CNDSR(B)IN CW TEMP L A/B'	'Avg. CNDSR(C) OUT CW TEMP C/D'
'Avg. CNDSR(B)IN CW TEMP L A/B'	'Avg. CNDSR(C)IN CW TEMP L A/B'
'Avg. CNDSR(B)IN CW TEMP L A/B'	'Avg. CNDSR(C)IN CW TEMP L C/D'
'Avg. CNDSR(B)IN CW TEMP L A/B'	'CONDENSER(A) PRESSURE'
'Avg. CNDSR(B)IN CW TEMP L A/B'	'CONDENSER(B) PRESSURE'
'Avg. CNDSR(B)IN CW TEMP L A/B'	'CONDENSER(C) PRESSURE'
'Avg. CNDSR(B)IN CW TEMP L A/B'	'HTR 1A DRAIN TEMP'
'Avg. CNDSR(B)IN CW TEMP L A/B'	'HTR 1B DRAIN TEMP'
'Avg. CNDSR(B)IN CW TEMP L A/B'	'HTR 1C DRAIN TEMP'
'Avg. CNDSR(B)IN CW TEMP L A/B'	'HTR 2A DRAIN TEMP'
'Avg. CNDSR(B)IN CW TEMP L A/B'	'HTR 2B DRAIN TEMP'
'Avg. CNDSR(B)IN CW TEMP L A/B'	'HTR 2C DRAIN TEMP'
'Avg. CNDSR(B)IN CW TEMP L A/B'	'TRB HEAT RATE GROSS DESIGN'
'Avg. CNDSR(B)IN CW TEMP L C/D'	'Avg. CNDSR(C) OUT CW TEMP A/B'
'Avg. CNDSR(B)IN CW TEMP L C/D'	'Avg. CNDSR(C) OUT CW TEMP C/D'
'Avg. CNDSR(B)IN CW TEMP L C/D'	'Avg. CNDSR(C)IN CW TEMP L A/B'
'Avg. CNDSR(B)IN CW TEMP L C/D'	'Avg. CNDSR(C)IN CW TEMP L C/D'
'Avg. CNDSR(B)IN CW TEMP L C/D'	'CONDENSER(A) PRESSURE'
'Avg. CNDSR(B)IN CW TEMP L C/D'	'CONDENSER(B) PRESSURE'
'Avg. CNDSR(B)IN CW TEMP L C/D'	'CONDENSER(C) PRESSURE'
'Avg. CNDSR(B)IN CW TEMP L C/D'	'HTR 1A DRAIN TEMP'
'Avg. CNDSR(B)IN CW TEMP L C/D'	'HTR 1B DRAIN TEMP'
'Avg. CNDSR(B)IN CW TEMP L C/D'	'HTR 1C DRAIN TEMP'
'Avg. CNDSR(B)IN CW TEMP L C/D'	'HTR 2A DRAIN TEMP'
'Avg. CNDSR(B)IN CW TEMP L C/D'	'HTR 2B DRAIN TEMP'
'Avg. CNDSR(B)IN CW TEMP L C/D'	'HTR 2C DRAIN TEMP'
'Avg. CNDSR(B)IN CW TEMP L C/D'	'TRB HEAT RATE GROSS DESIGN'
'Avg. CNDSR(C) OUT CW TEMP A/B'	'Avg. CNDSR(C) OUT CW TEMP C/D'
'Avg. CNDSR(C) OUT CW TEMP A/B'	'Avg. CNDSR(C)IN CW TEMP L A/B'
'Avg. CNDSR(C) OUT CW TEMP A/B'	'Avg. CNDSR(C)IN CW TEMP L C/D'
'Avg. CNDSR(C) OUT CW TEMP A/B'	'CONDENSER(A) PRESSURE'
'Avg. CNDSR(C) OUT CW TEMP A/B'	'CONDENSER(B) PRESSURE'
'Avg. CNDSR(C) OUT CW TEMP A/B'	'CONDENSER(C) PRESSURE'
'Avg. CNDSR(C) OUT CW TEMP A/B'	'HTR 1A DRAIN TEMP'
'Avg. CNDSR(C) OUT CW TEMP A/B'	'HTR 1B DRAIN TEMP'
'Avg. CNDSR(C) OUT CW TEMP A/B'	'HTR 1C DRAIN TEMP'
'Avg. CNDSR(C) OUT CW TEMP A/B'	'HTR 2A DRAIN TEMP'
'Avg. CNDSR(C) OUT CW TEMP A/B'	'HTR 2B DRAIN TEMP'
'Avg. CNDSR(C) OUT CW TEMP A/B'	'HTR 2C DRAIN TEMP'
'Avg. CNDSR(C) OUT CW TEMP A/B'	'TRB HEAT RATE GROSS DESIGN'
'Avg. CNDSR(C) OUT CW TEMP C/D'	'Avg. CNDSR(C)IN CW TEMP L A/B'
'Avg. CNDSR(C) OUT CW TEMP C/D'	'Avg. CNDSR(C)IN CW TEMP L C/D'

'Avg. CNDSR(C) OUT CW TEMP C/D'	'CONDENSER(A) PRESSURE'
'Avg. CNDSR(C) OUT CW TEMP C/D'	'CONDENSER(B) PRESSURE'
'Avg. CNDSR(C) OUT CW TEMP C/D'	'CONDENSER(C) PRESSURE'
'Avg. CNDSR(C) OUT CW TEMP C/D'	'HTR 1A DRAIN TEMP'
'Avg. CNDSR(C) OUT CW TEMP C/D'	'HTR 1B DRAIN TEMP'
'Avg. CNDSR(C) OUT CW TEMP C/D'	'HTR 1C DRAIN TEMP'
'Avg. CNDSR(C) OUT CW TEMP C/D'	'HTR 2A DRAIN TEMP'
'Avg. CNDSR(C) OUT CW TEMP C/D'	'HTR 2B DRAIN TEMP'
'Avg. CNDSR(C) OUT CW TEMP C/D'	'HTR 2C DRAIN TEMP'
'Avg. CNDSR(C) OUT CW TEMP C/D'	'TRB HEAT RATE GROSS DESIGN'
'Avg. CNDSR(C)IN CW TEMP L A/B'	'Avg. CNDSR(C)IN CW TEMP L C/D'
'Avg. CNDSR(C)IN CW TEMP L A/B'	'CONDENSER(A) PRESSURE'
'Avg. CNDSR(C)IN CW TEMP L A/B'	'CONDENSER(B) PRESSURE'
'Avg. CNDSR(C)IN CW TEMP L A/B'	'CONDENSER(C) PRESSURE'
'Avg. CNDSR(C)IN CW TEMP L A/B'	'HTR 1A DRAIN TEMP'
'Avg. CNDSR(C)IN CW TEMP L A/B'	'HTR 1B DRAIN TEMP'
'Avg. CNDSR(C)IN CW TEMP L A/B'	'HTR 1C DRAIN TEMP'
'Avg. CNDSR(C)IN CW TEMP L A/B'	'HTR 2A DRAIN TEMP'
'Avg. CNDSR(C)IN CW TEMP L A/B'	'HTR 2B DRAIN TEMP'
'Avg. CNDSR(C)IN CW TEMP L A/B'	'HTR 2C DRAIN TEMP'
'Avg. CNDSR(C)IN CW TEMP L A/B'	'TRB HEAT RATE GROSS DESIGN'
'Avg. CNDSR(C)IN CW TEMP L C/D'	'CONDENSER(A) PRESSURE'
'Avg. CNDSR(C)IN CW TEMP L C/D'	'CONDENSER(B) PRESSURE'
'Avg. CNDSR(C)IN CW TEMP L C/D'	'CONDENSER(C) PRESSURE'
'Avg. CNDSR(C)IN CW TEMP L C/D'	'HTR 1A DRAIN TEMP'
'Avg. CNDSR(C)IN CW TEMP L C/D'	'HTR 1B DRAIN TEMP'
'Avg. CNDSR(C)IN CW TEMP L C/D'	'HTR 1C DRAIN TEMP'
'Avg. CNDSR(C)IN CW TEMP L C/D'	'HTR 2A DRAIN TEMP'
'Avg. CNDSR(C)IN CW TEMP L C/D'	'HTR 2B DRAIN TEMP'
'Avg. CNDSR(C)IN CW TEMP L C/D'	'HTR 2C DRAIN TEMP'
'Avg. CNDSR(C)IN CW TEMP L C/D'	'TRB HEAT RATE GROSS DESIGN'
'CONDENSER(A) PRESSURE'	'CONDENSER(B) PRESSURE'
'CONDENSER(A) PRESSURE'	'CONDENSER(C) PRESSURE'
'CONDENSER(A) PRESSURE'	'HTR 1A DRAIN TEMP'
'CONDENSER(A) PRESSURE'	'HTR 1B DRAIN TEMP'
'CONDENSER(A) PRESSURE'	'HTR 1C DRAIN TEMP'
'CONDENSER(A) PRESSURE'	'HTR 2A DRAIN TEMP'
'CONDENSER(A) PRESSURE'	'HTR 2B DRAIN TEMP'
'CONDENSER(A) PRESSURE'	'HTR 2C DRAIN TEMP'
'CONDENSER(A) PRESSURE'	'TRB HEAT RATE GROSS DESIGN'
'CONDENSER(B) PRESSURE'	'CONDENSER(C) PRESSURE'
'CONDENSER(B) PRESSURE'	'HTR 1A DRAIN TEMP'
'CONDENSER(B) PRESSURE'	'HTR 1B DRAIN TEMP'
'CONDENSER(B) PRESSURE'	'HTR 1C DRAIN TEMP'
'CONDENSER(B) PRESSURE'	'HTR 2A DRAIN TEMP'
'CONDENSER(B) PRESSURE'	'HTR 2B DRAIN TEMP'
'CONDENSER(B) PRESSURE'	'HTR 2C DRAIN TEMP'
'CONDENSER(B) PRESSURE'	'TRB HEAT RATE GROSS DESIGN'
'CONDENSER(C) PRESSURE'	'HTR 1A DRAIN TEMP'

'CONDENSER(C) PRESSURE'	'HTR 1B DRAIN TEMP'
'CONDENSER(C) PRESSURE'	'HTR 1C DRAIN TEMP'
'CONDENSER(C) PRESSURE'	'HTR 2A DRAIN TEMP'
'CONDENSER(C) PRESSURE'	'HTR 2B DRAIN TEMP'
'CONDENSER(C) PRESSURE'	'HTR 2C DRAIN TEMP'
'CONDENSER(C) PRESSURE'	'TRB HEAT RATE GROSS DESIGN'
'FW PMP B FW SP VOL'	'HTR 6B DRAIN TEMP'
'HTR 1A DRAIN TEMP'	'HTR 1B DRAIN TEMP'
'HTR 1A DRAIN TEMP'	'HTR 1C DRAIN TEMP'
'HTR 1A DRAIN TEMP'	'HTR 2A DRAIN TEMP'
'HTR 1A DRAIN TEMP'	'HTR 2B DRAIN TEMP'
'HTR 1A DRAIN TEMP'	'HTR 2C DRAIN TEMP'
'HTR 1A DRAIN TEMP'	'TRB HEAT RATE GROSS DESIGN'
'HTR 1B DRAIN TEMP'	'HTR 1C DRAIN TEMP'
'HTR 1B DRAIN TEMP'	'HTR 2A DRAIN TEMP'
'HTR 1B DRAIN TEMP'	'HTR 2B DRAIN TEMP'
'HTR 1B DRAIN TEMP'	'HTR 2C DRAIN TEMP'
'HTR 1B DRAIN TEMP'	'TRB HEAT RATE GROSS DESIGN'
'HTR 1C DRAIN TEMP'	'HTR 2A DRAIN TEMP'
'HTR 1C DRAIN TEMP'	'HTR 2B DRAIN TEMP'
'HTR 1C DRAIN TEMP'	'HTR 2C DRAIN TEMP'
'HTR 1C DRAIN TEMP'	'TRB HEAT RATE GROSS DESIGN'
'HTR 2A DRAIN TEMP'	'HTR 2B DRAIN TEMP'
'HTR 2A DRAIN TEMP'	'HTR 2C DRAIN TEMP'
'HTR 2A DRAIN TEMP'	'TRB HEAT RATE GROSS DESIGN'
'HTR 2B DRAIN TEMP'	'HTR 2C DRAIN TEMP'
'HTR 2B DRAIN TEMP'	'TRB HEAT RATE GROSS DESIGN'
'HTR 5A DRAIN TEMP'	'HTR 5B DRAIN TEMP'
'HTR DRN PMP A EFF'	'HTR DRN PMP A EFF DEV'
'HTR DRN PMP B EFF'	'HTR DRN PMP B EFF DEV'

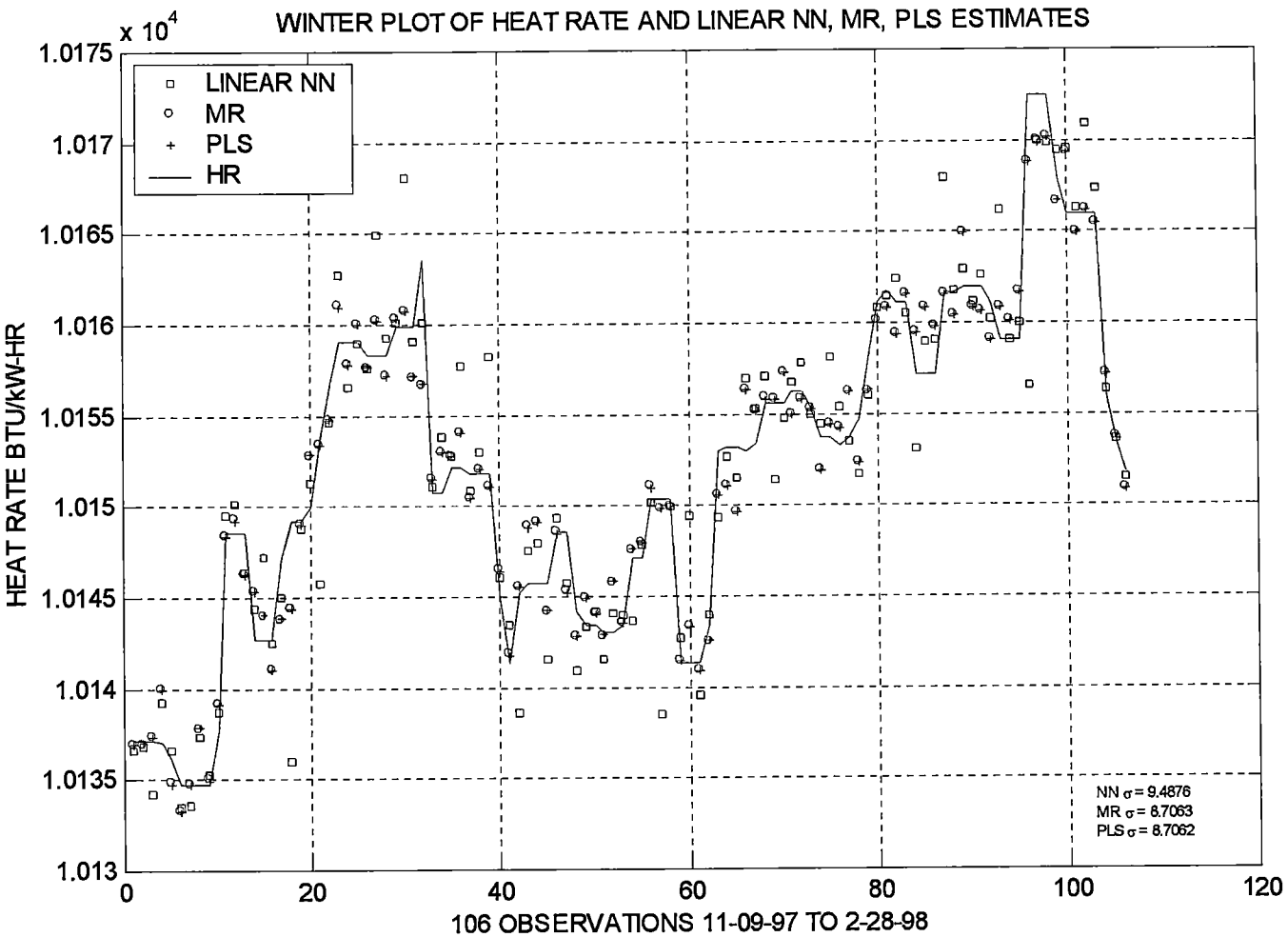

Figure A-67 Winter plot of heat rate (and linear NN, MR, PLS estimates)

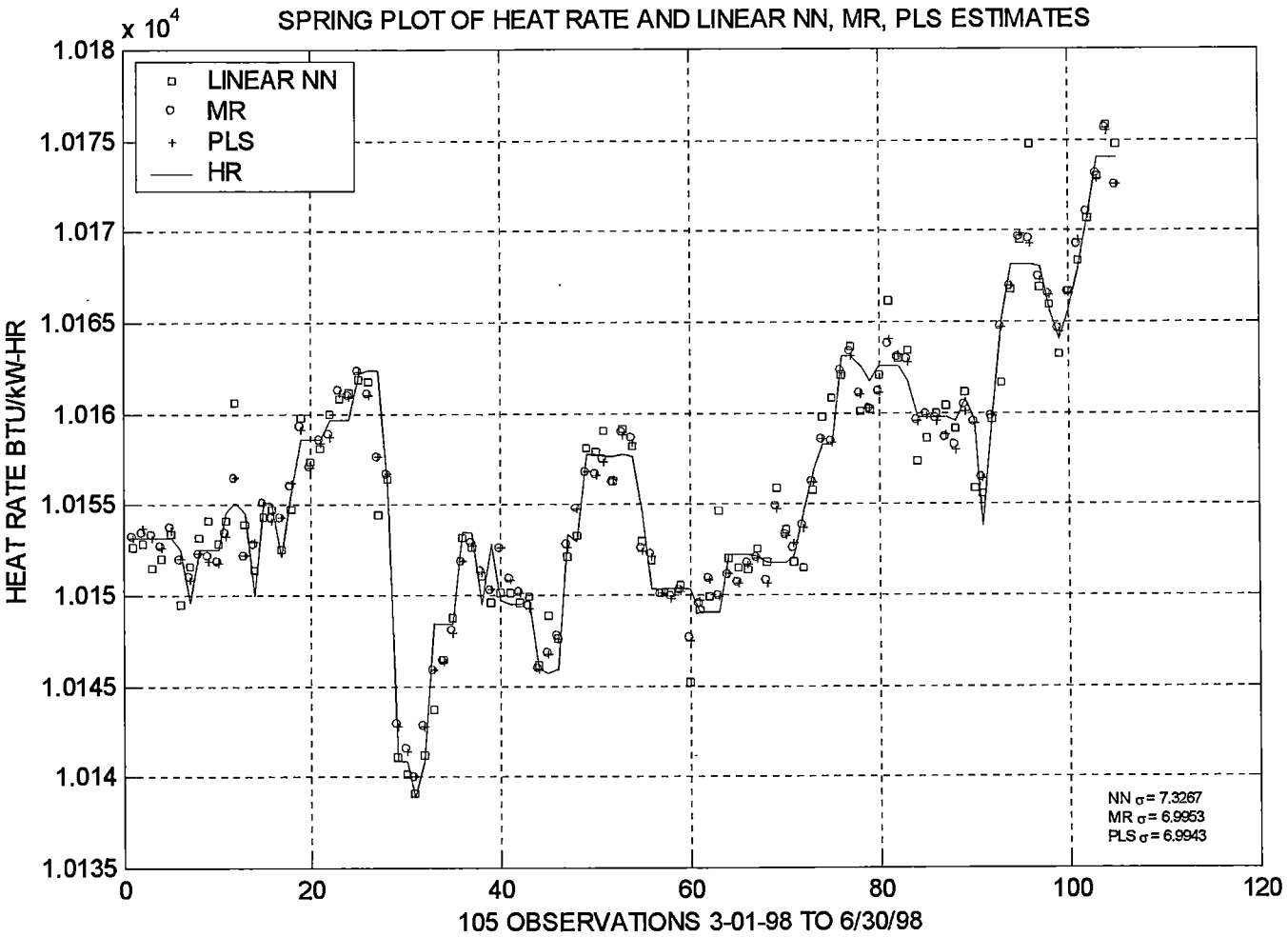

Figure A-68 Spring plot of heat rate (and linear NN, MR, PLS estimates)

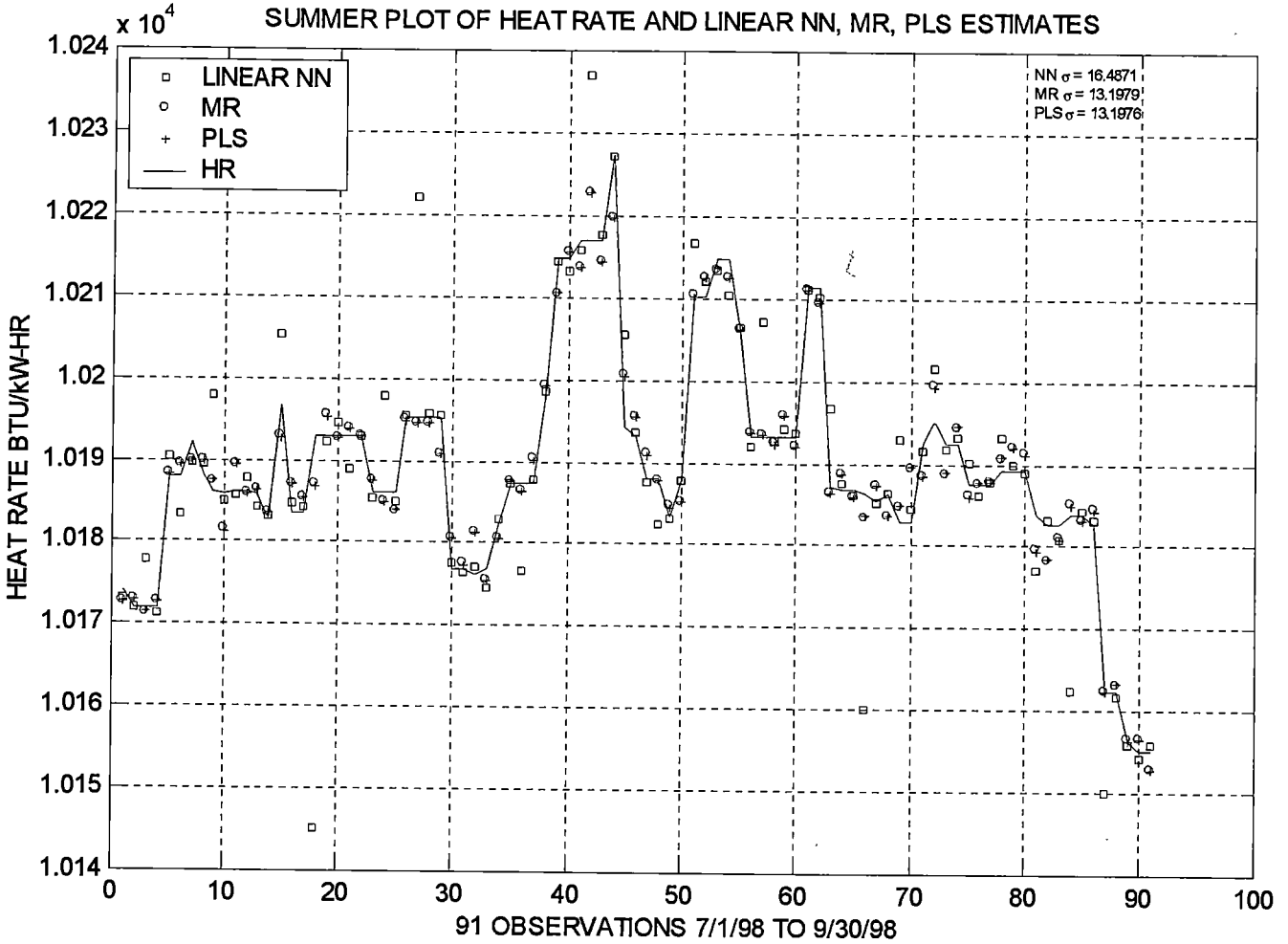

Figure A-69 Summer plot of heat rate (and linear NN, MR, PLS estimates)

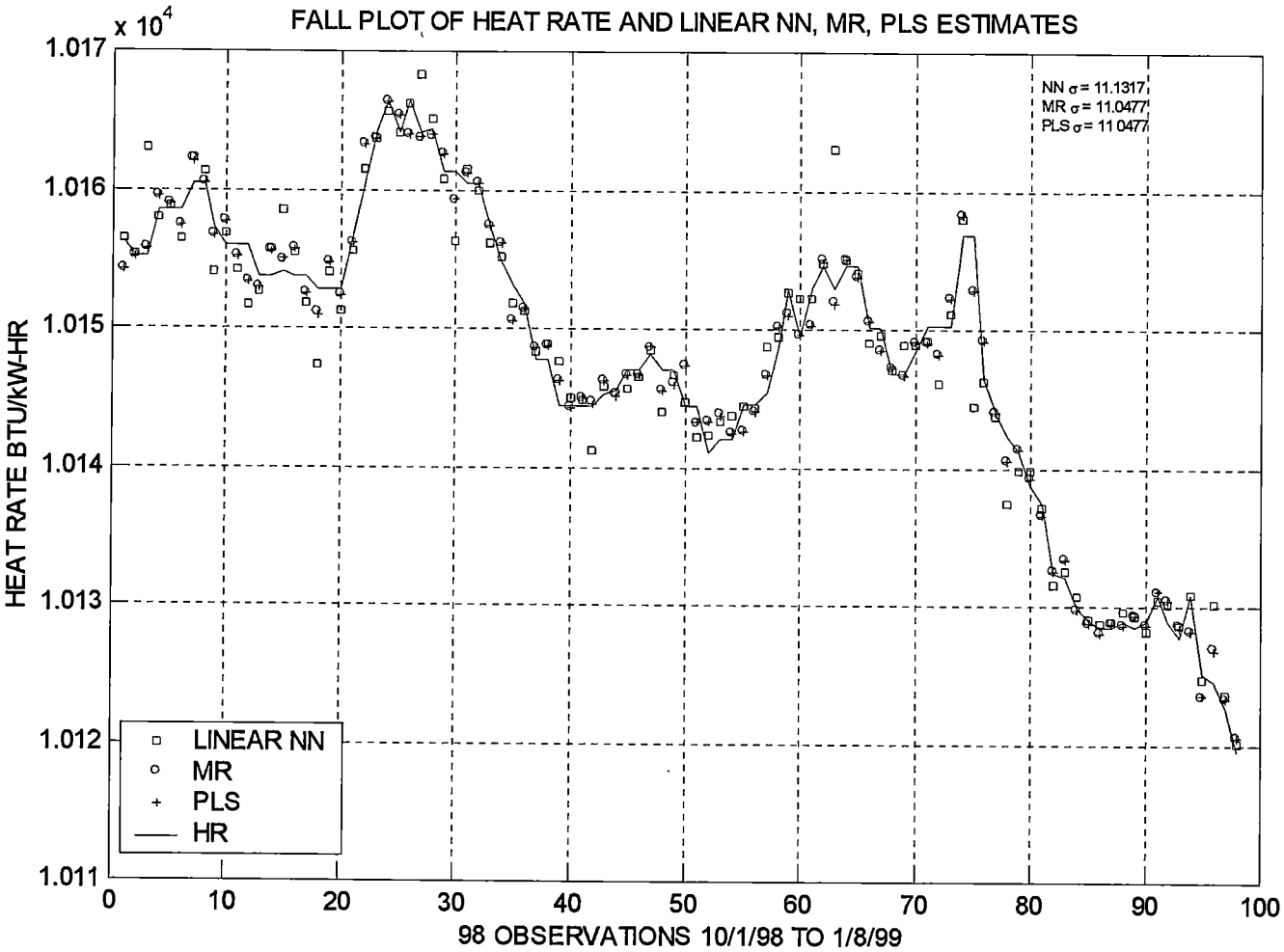

Figure A-70 Fall plot of heat rate (and linear NN, MR, PLS estimates)

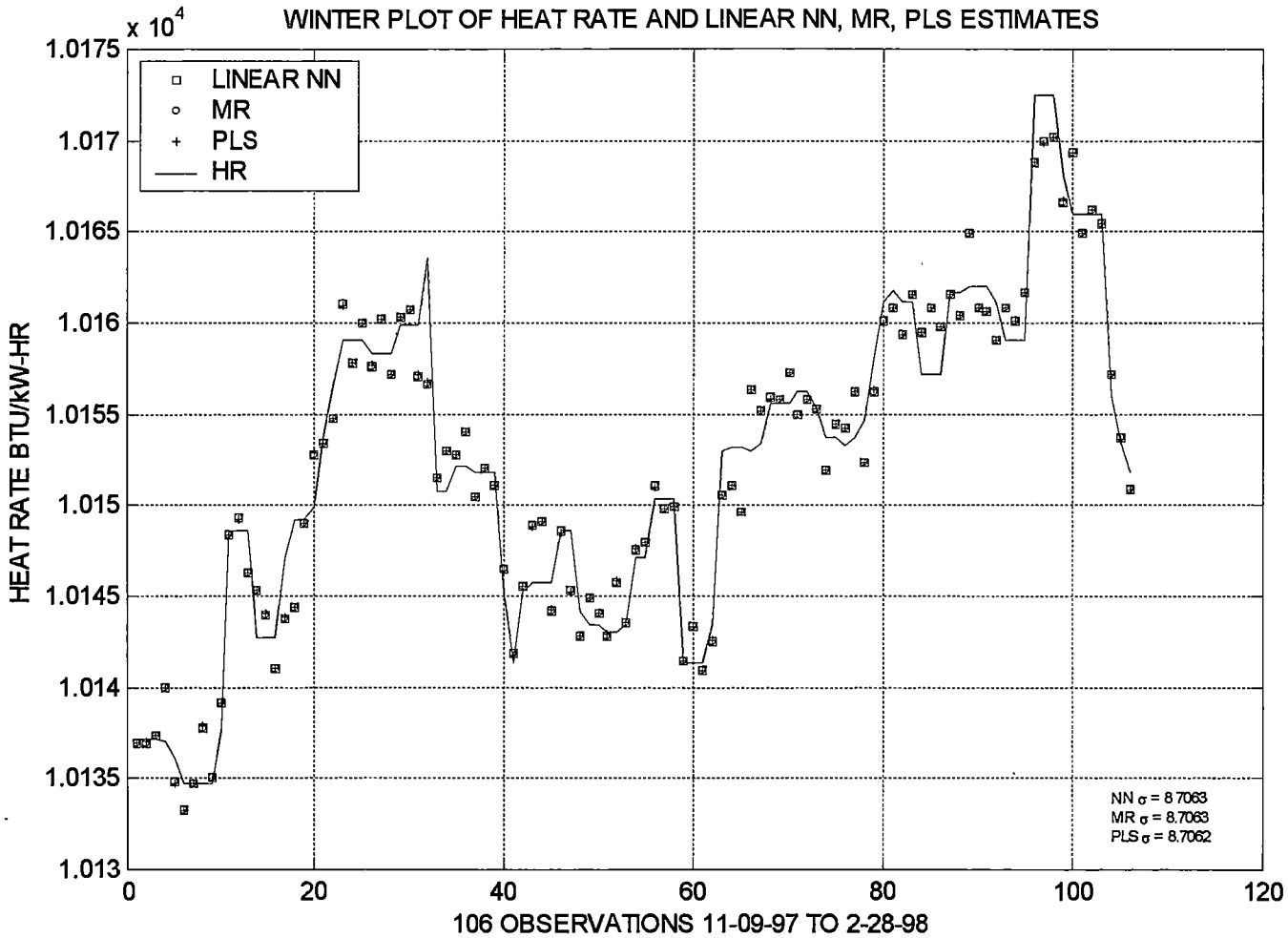

Figure A-71 Winter plot of heat rate (and linear NN (no VV points), MR, PLS estimates)

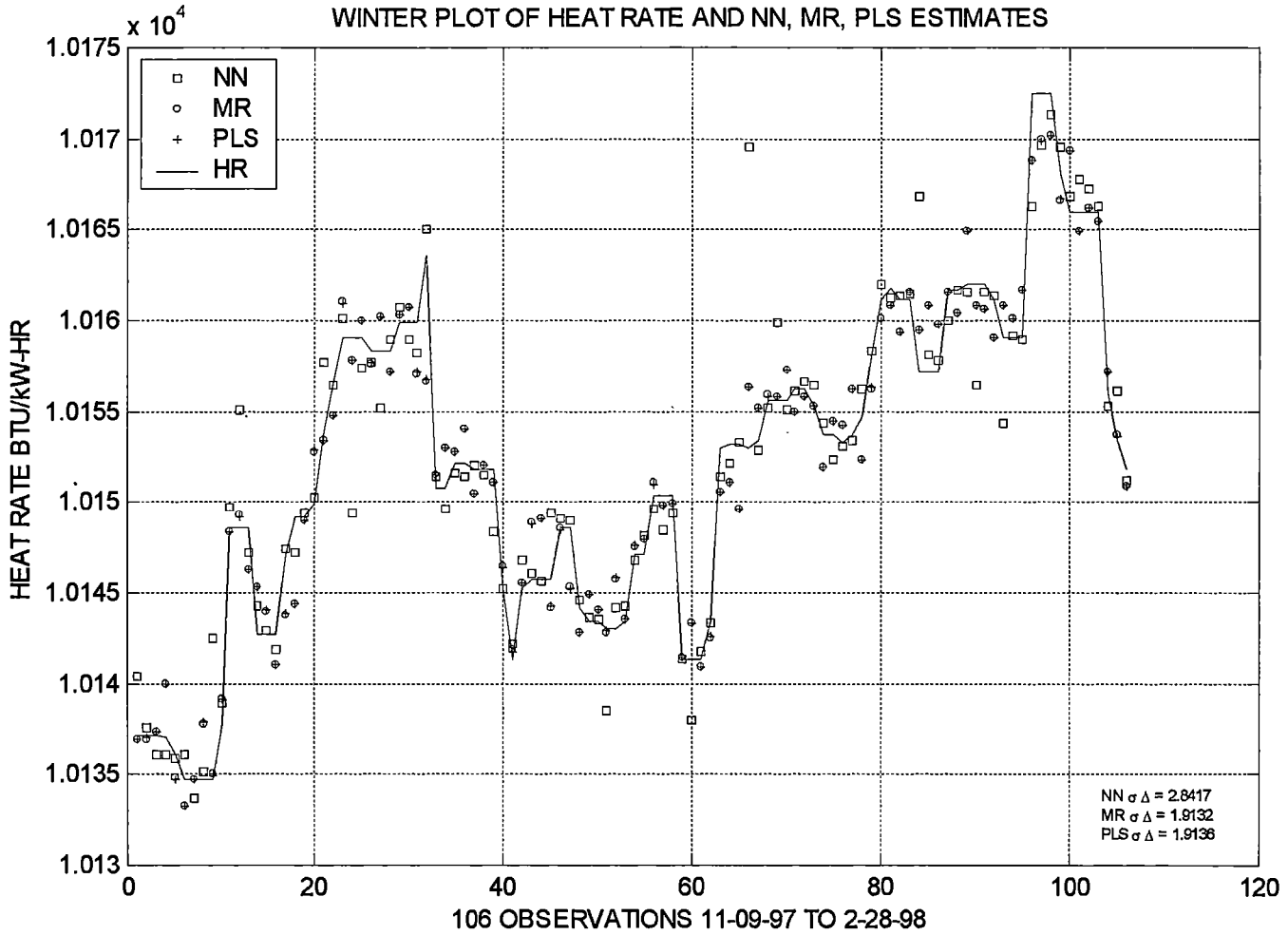

Figure A-72 Winter plot of heat rate (and NN, MR, PLS estimates)

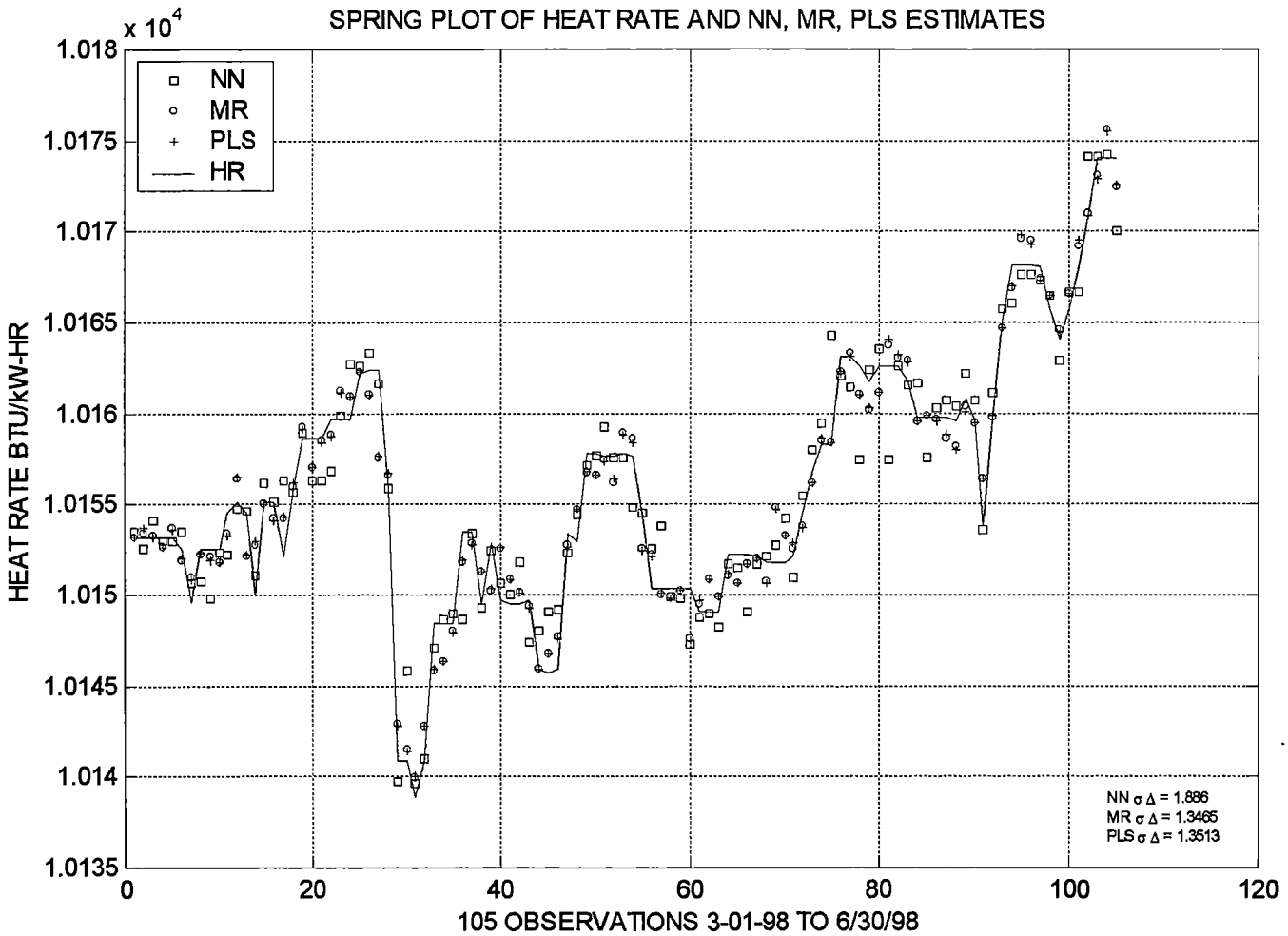

Figure A-73 Spring plot of heat rate (and NN, MR, PLS estimates)

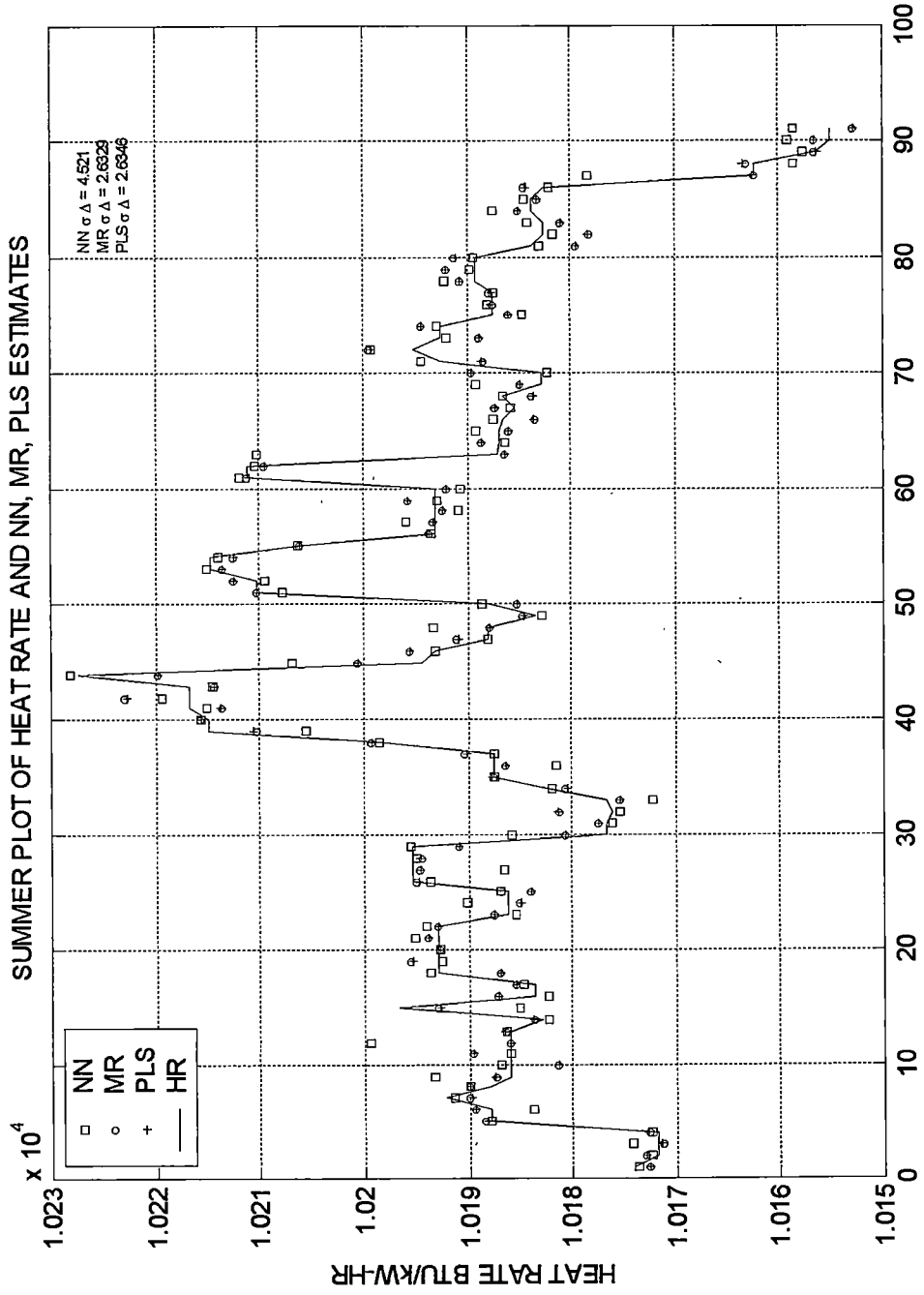

Figure A-74 Summer plot of heat rate (and NN, MR, PLS estimates)

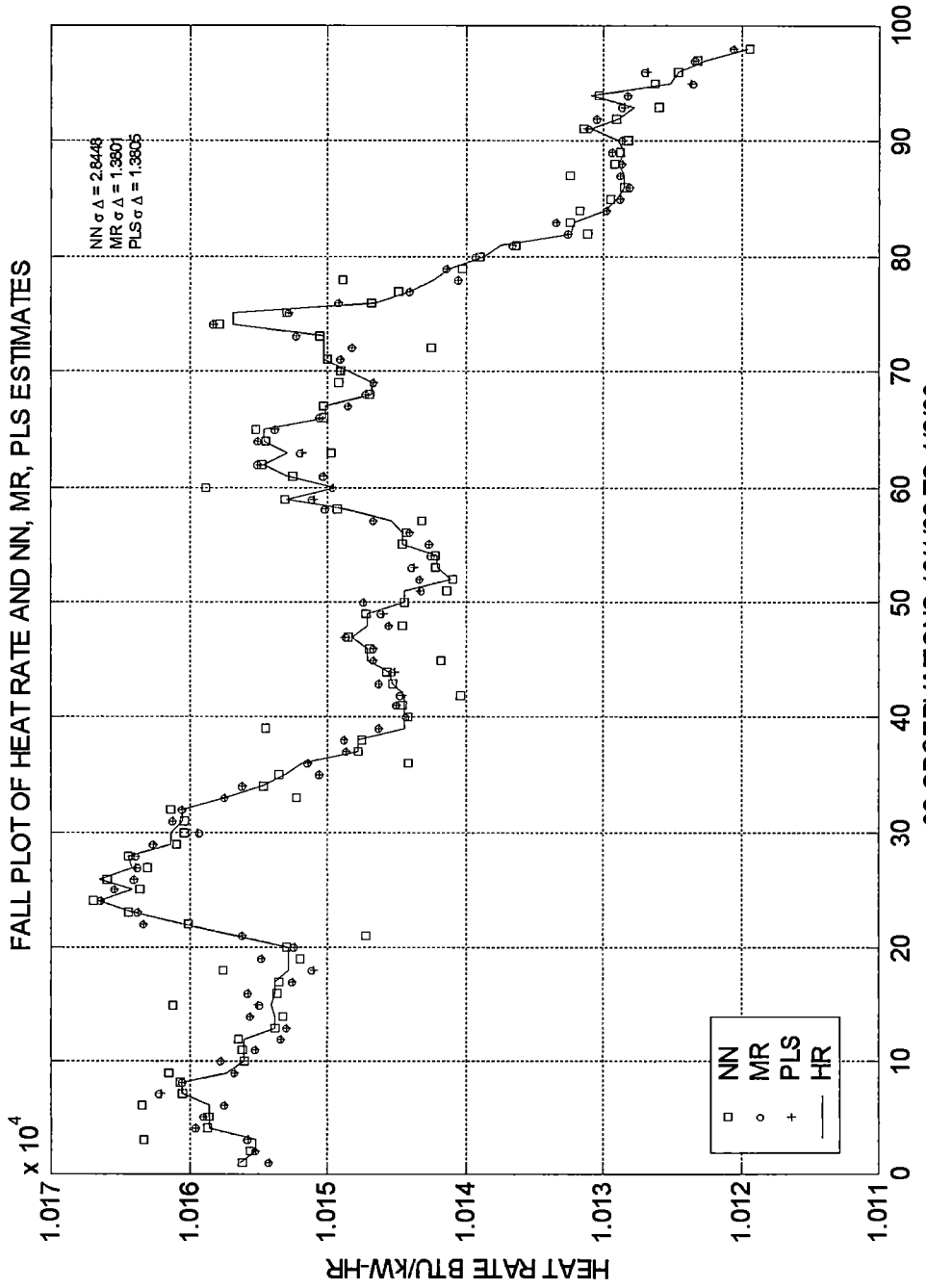

Figure A-75 Fall plot of heat rate (and NN, MR, PLS estimates)

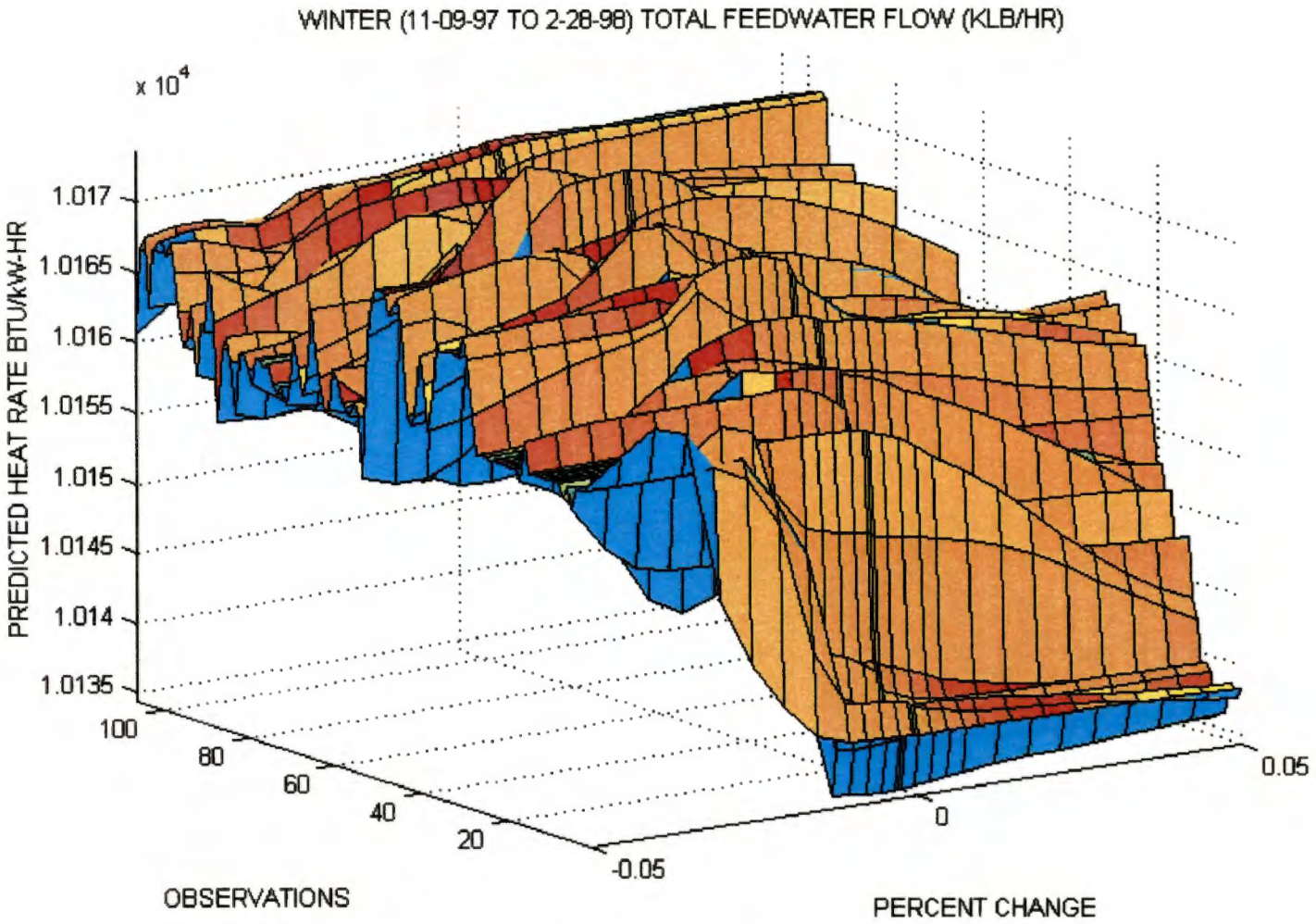

Figure A-77 Winter plot, NN estimate of heat rate

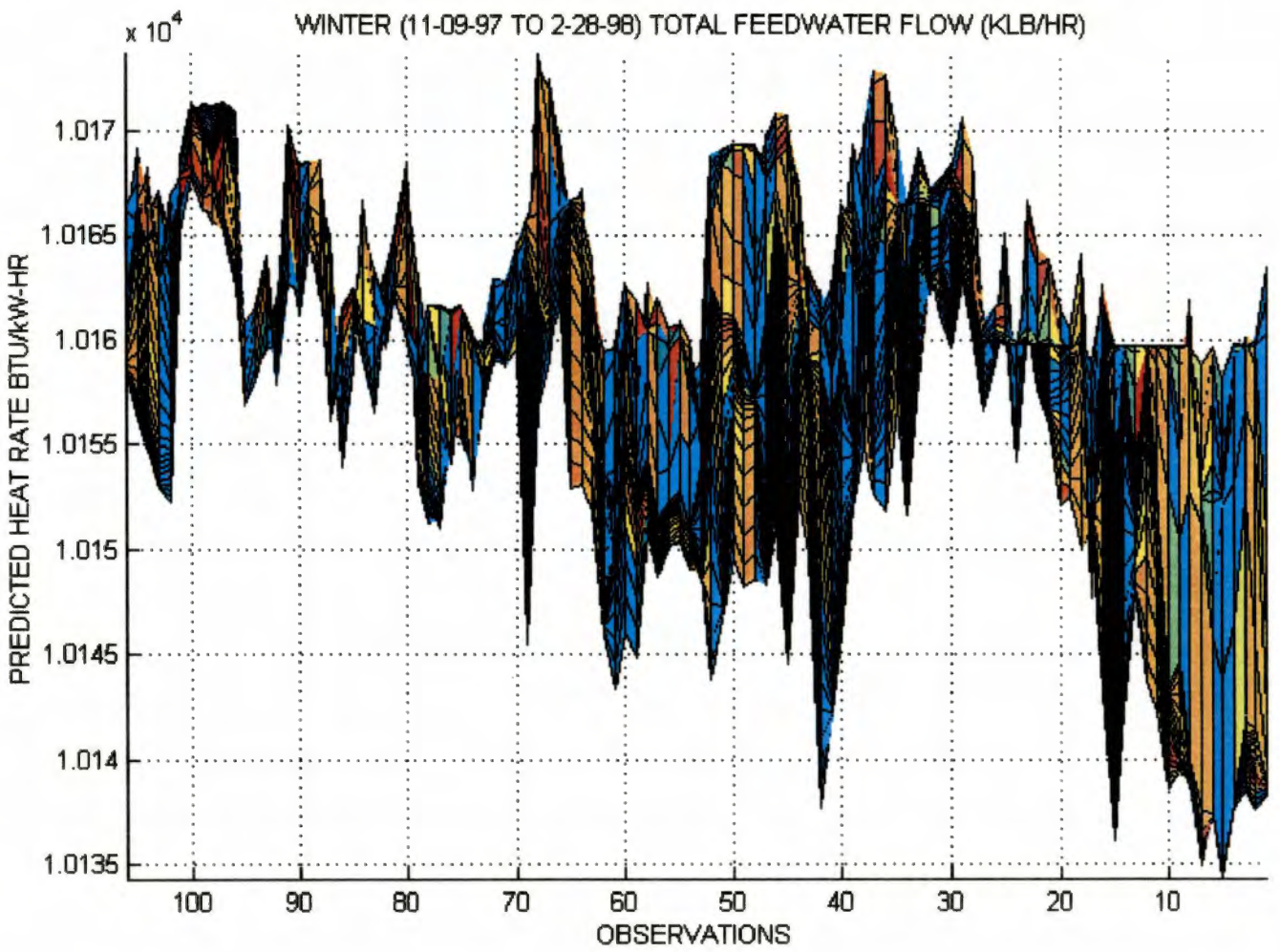

Figure A-78 Figure A-77 rotated

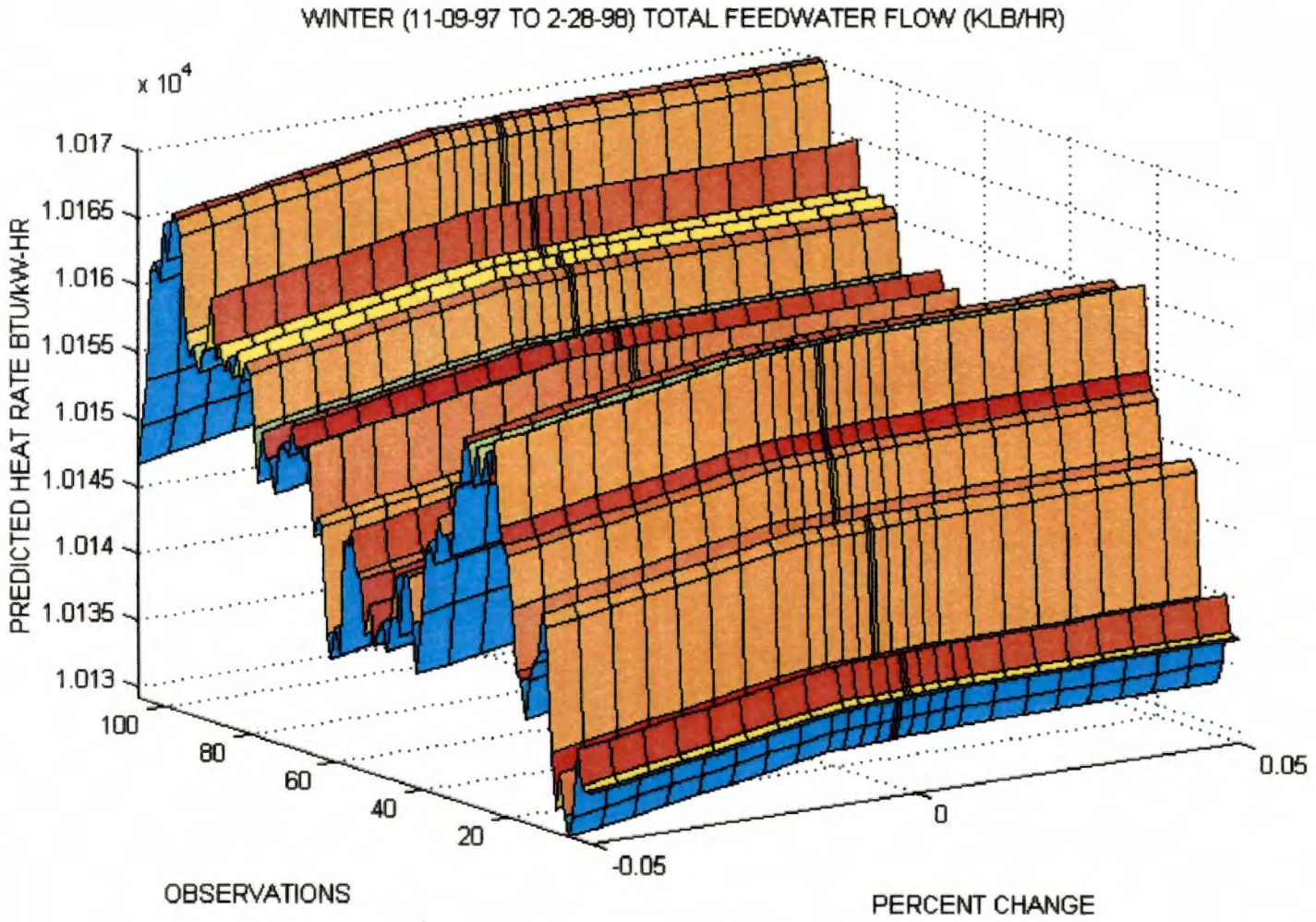

Figure A-79 Winter plot, MR estimate of heat rate

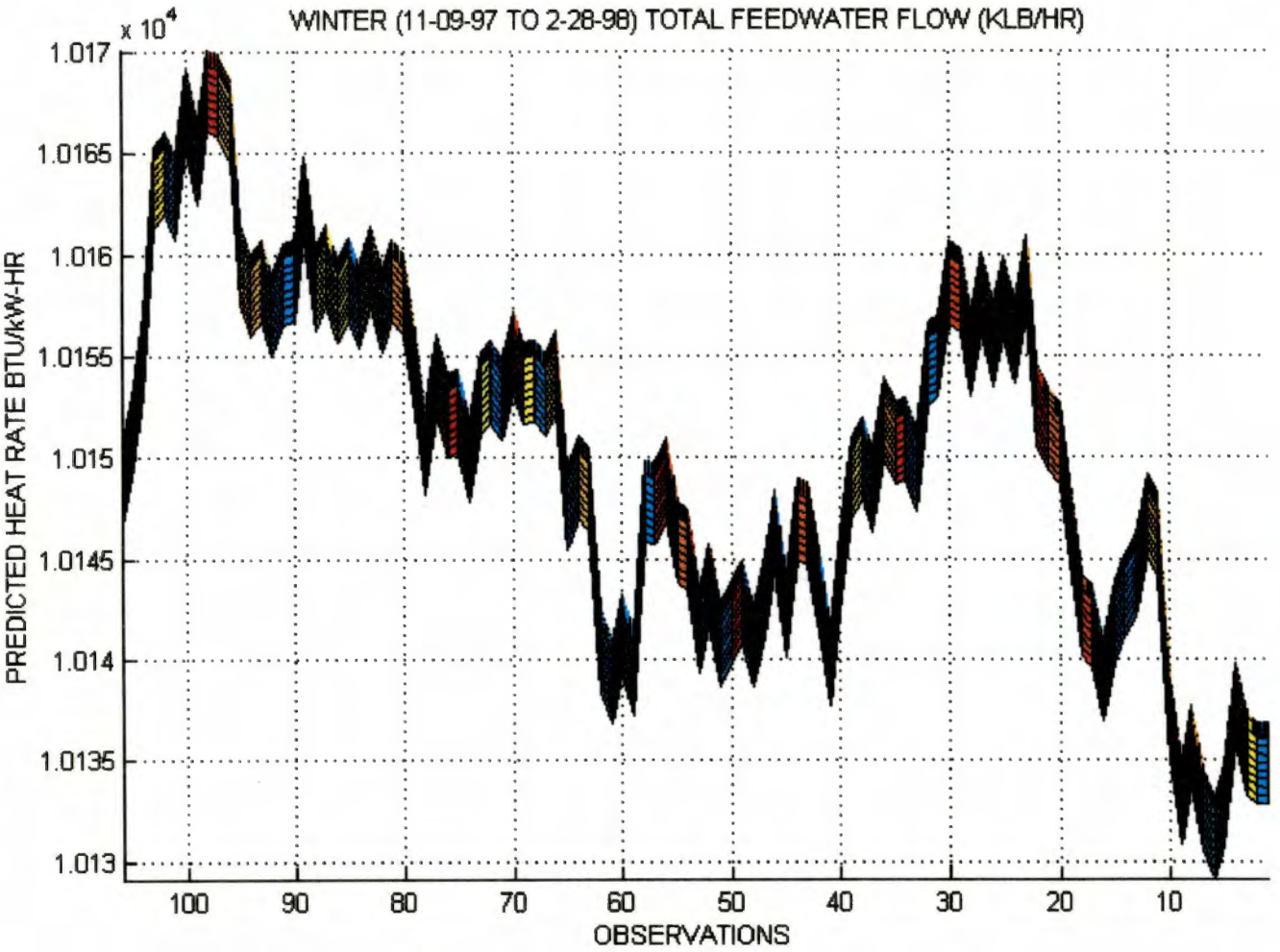

Figure A-80 Figure A-79 rotated

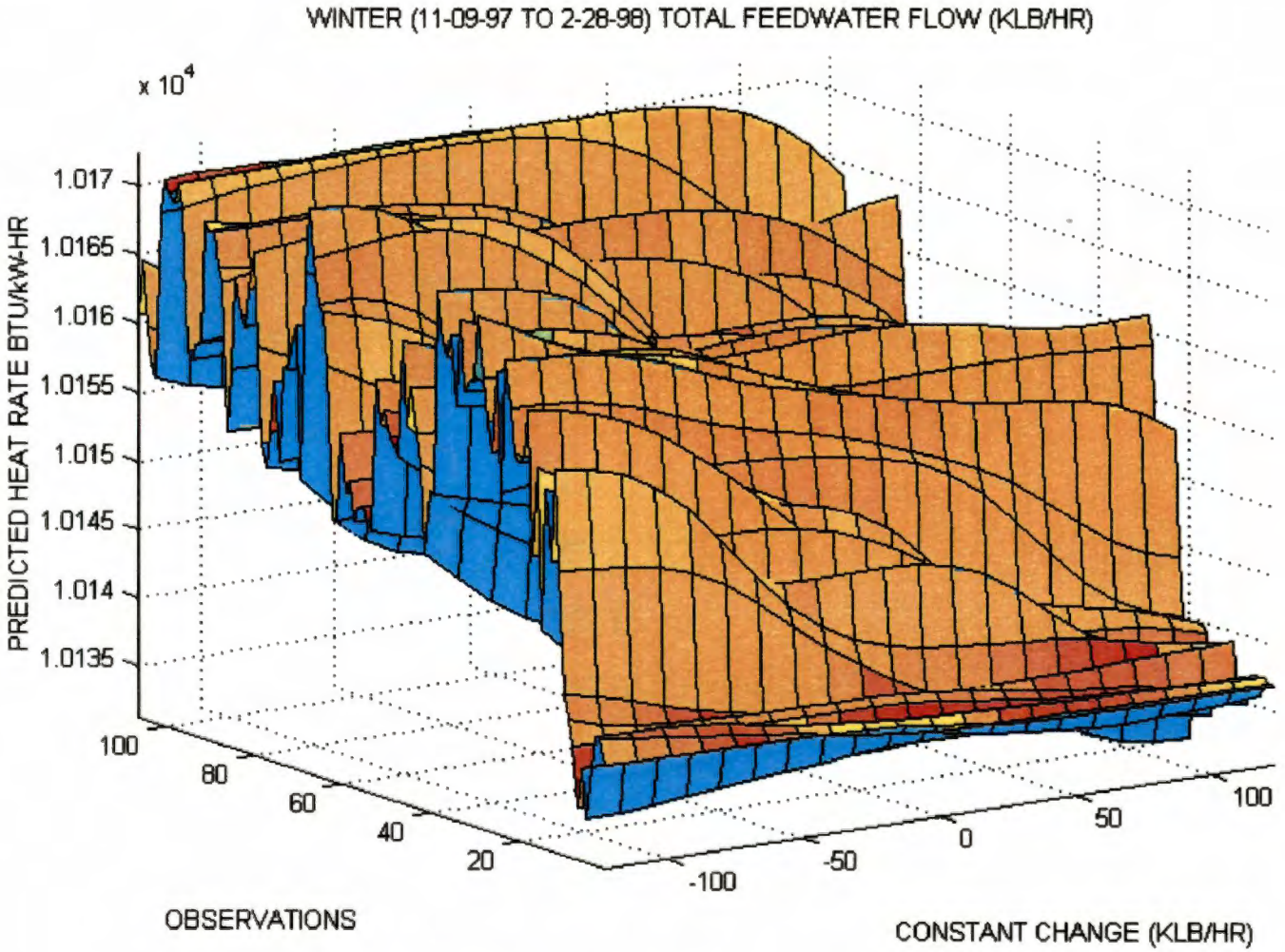

Figure A-81 Winter plot, NN estimate of heat rate (for constant change)

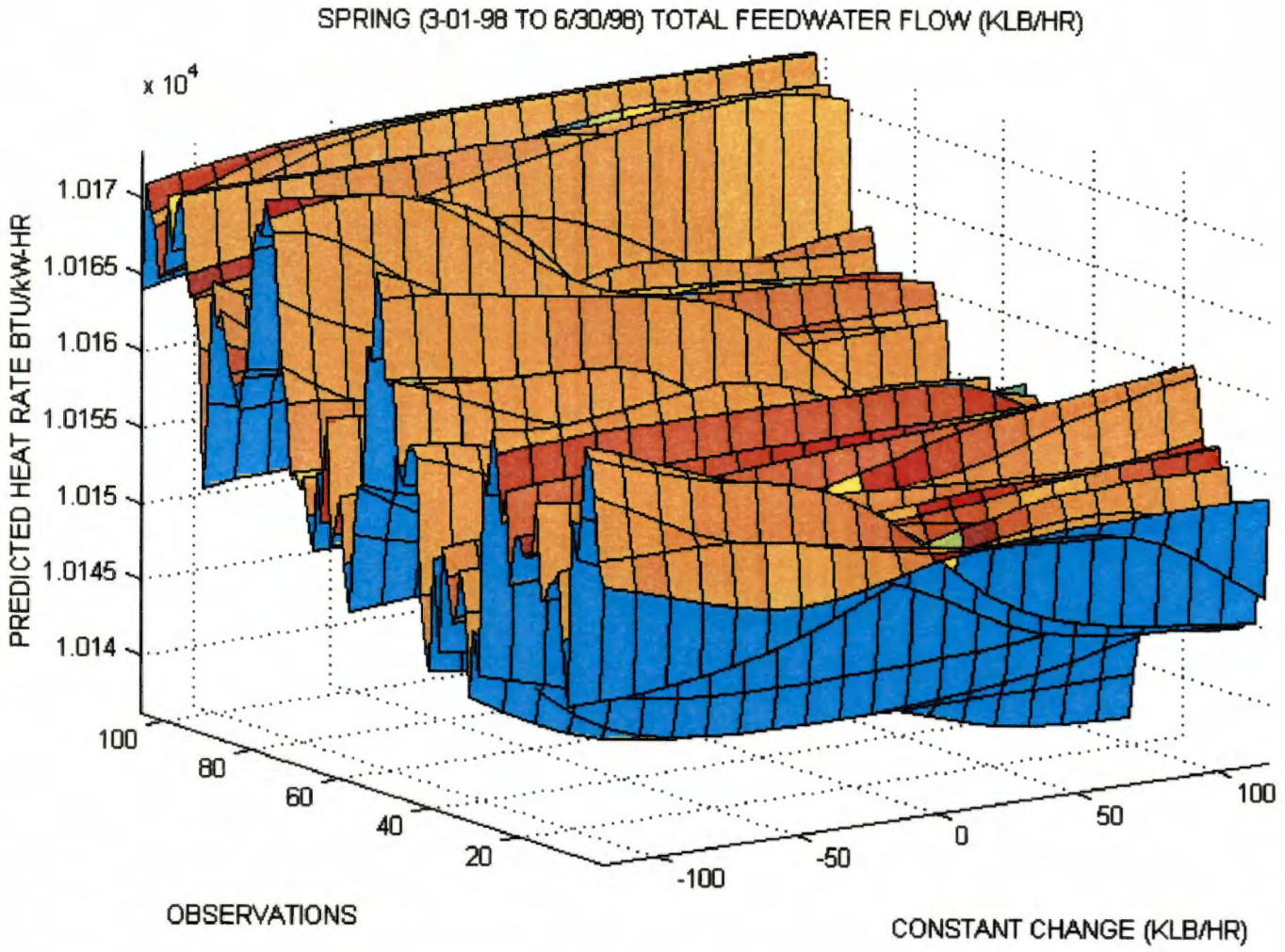

Figure A-82 Spring plot, NN estimate of heat rate (for constant change)

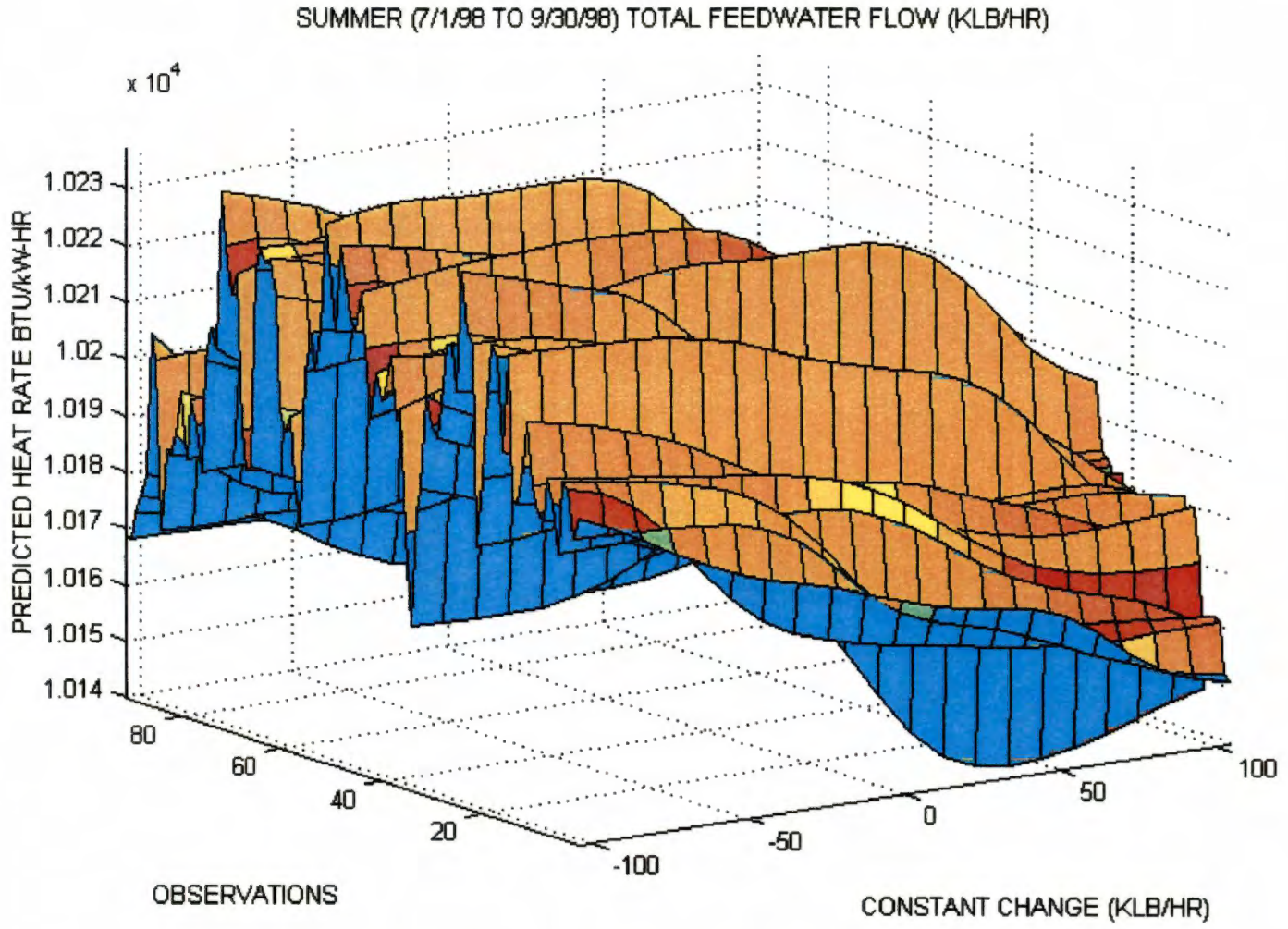

Figure A-83 Summer plot, NN estimate of heat rate (for constant change)

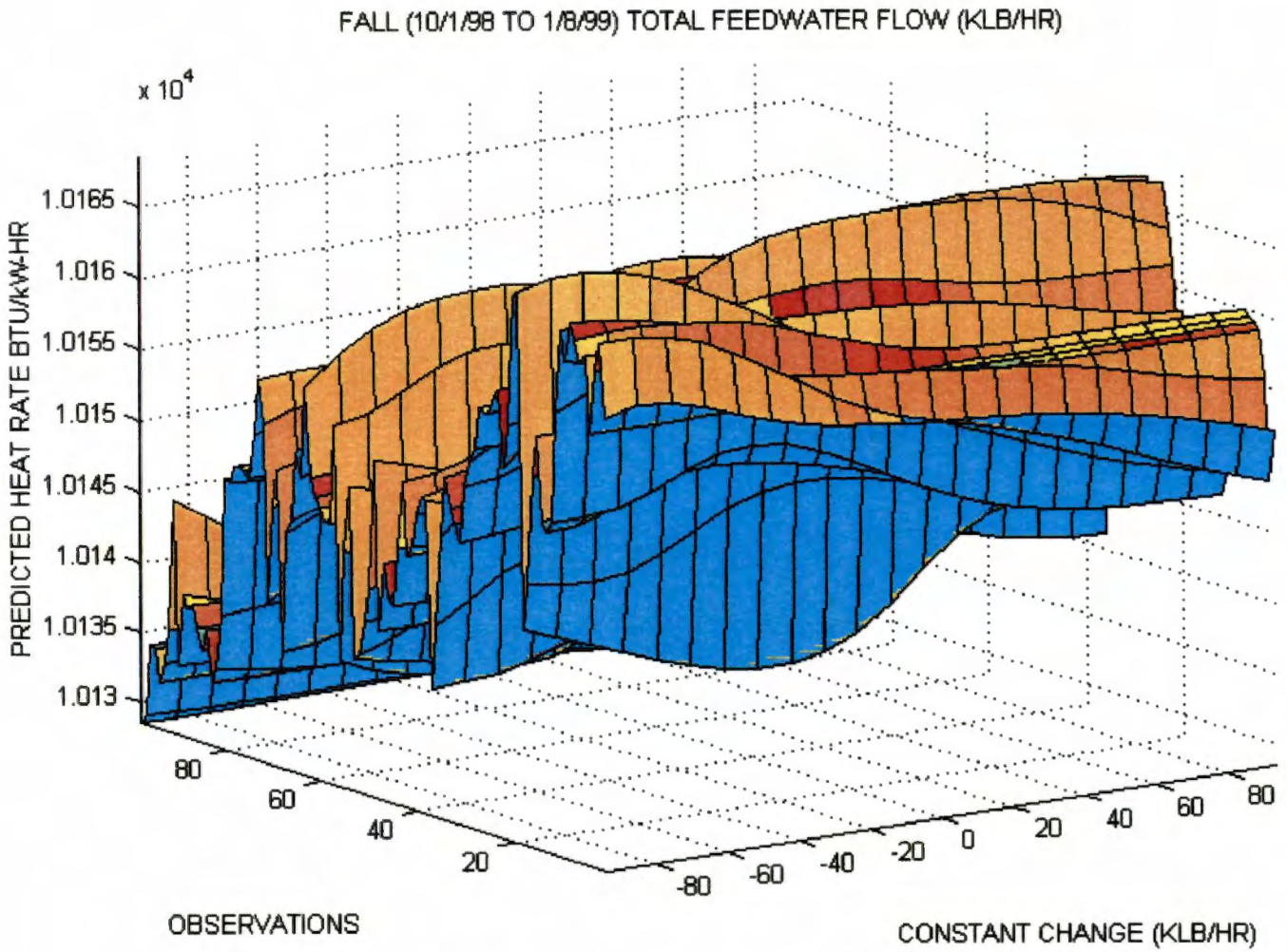

Figure A-84 Fall plot, NN estimate of heat rate (for constant change)

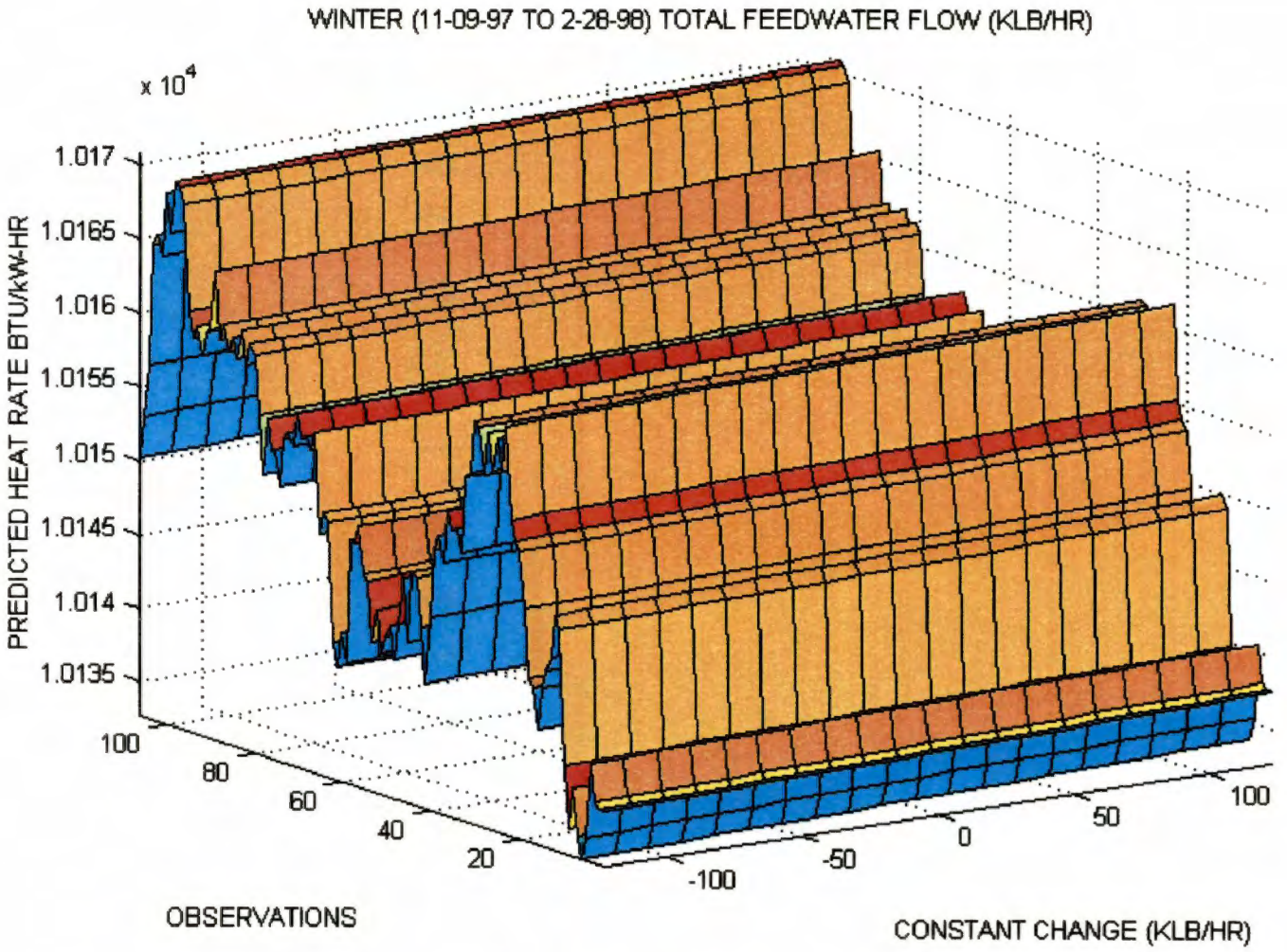

Figure A-85 Winter plot, MR estimate of heat rate (for constant change)

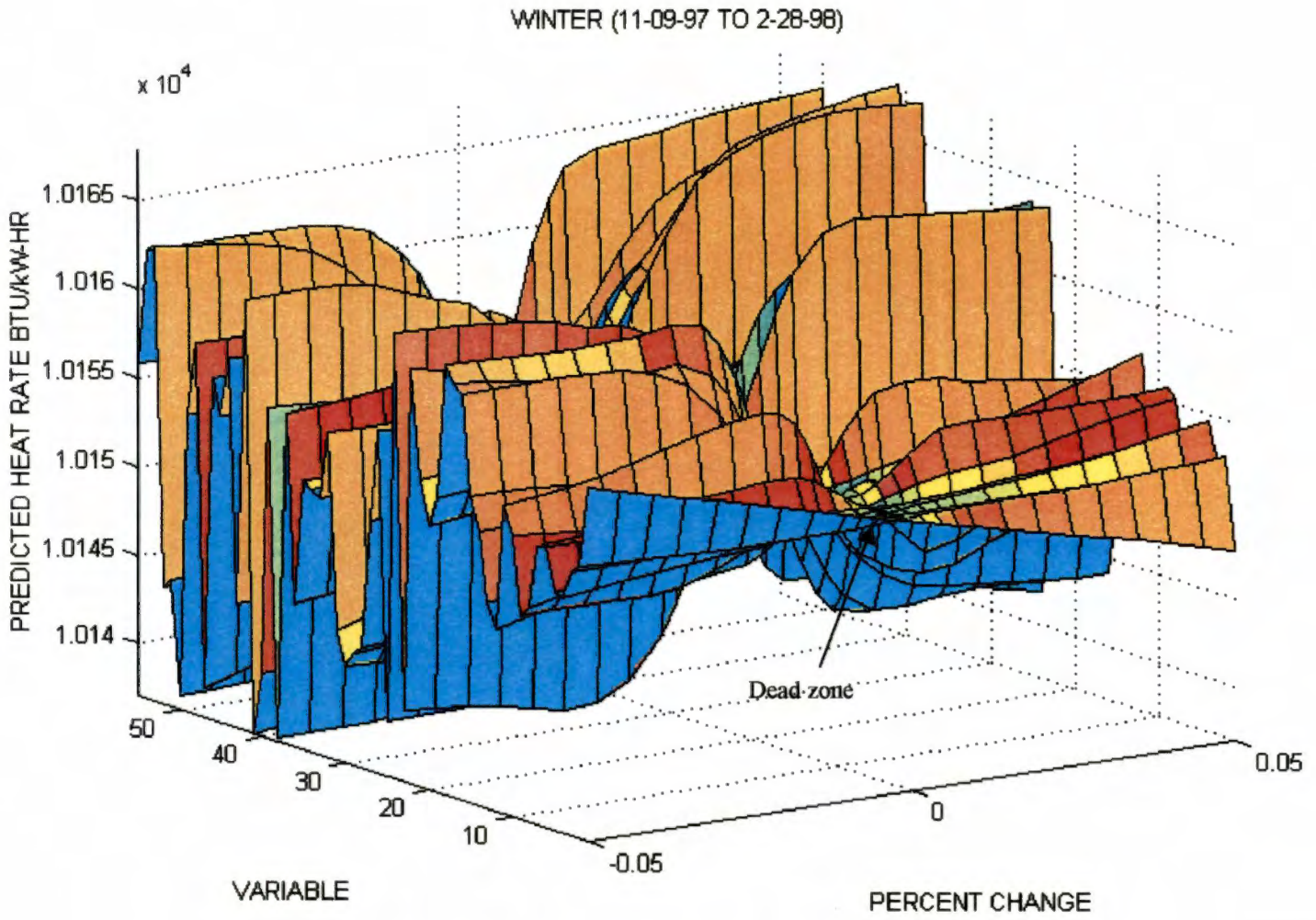

Figure A-86 Winter plot, NN estimate of heat rate for all variables 1-55

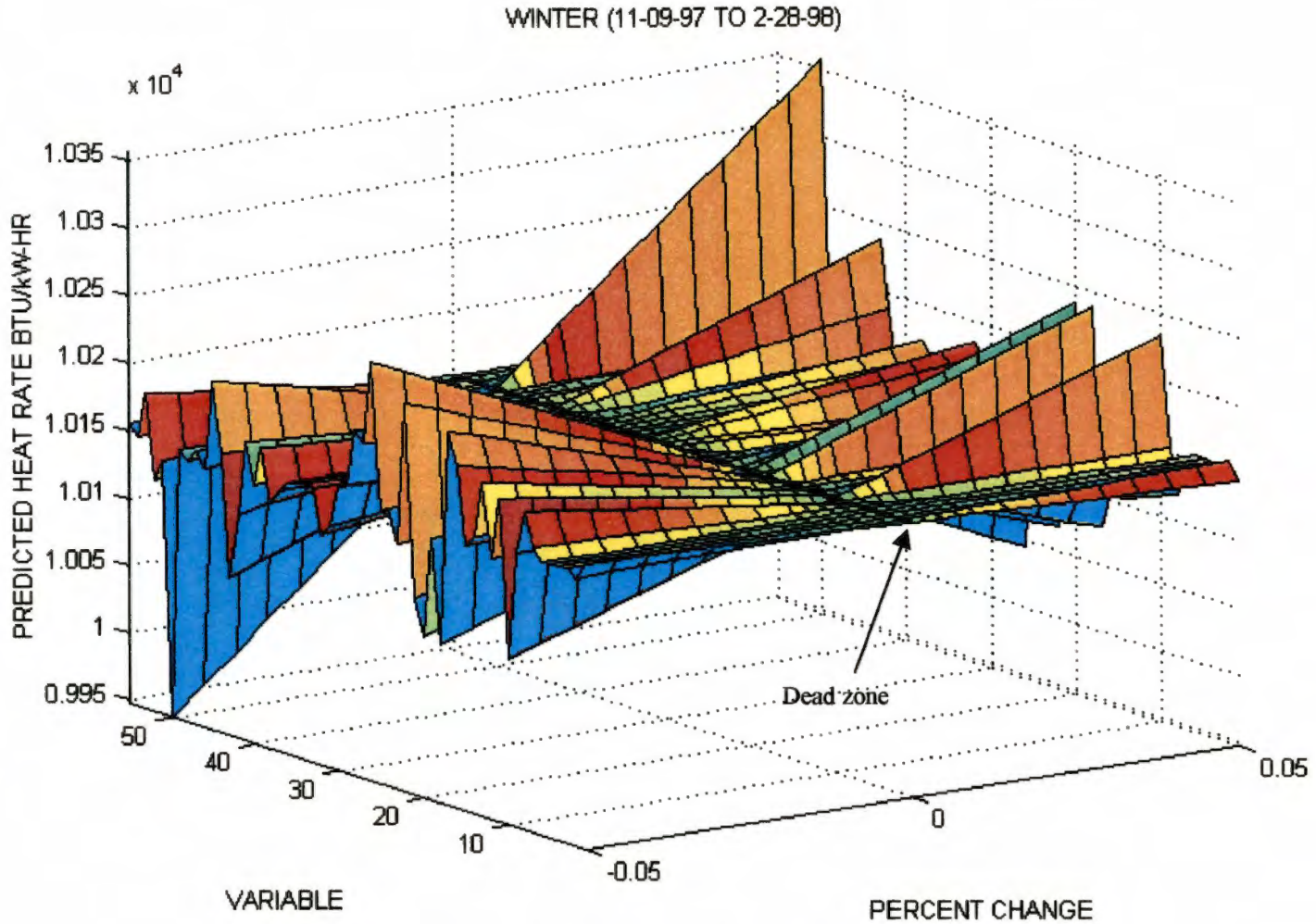

Figure A-87 Winter plot, MR estimate of heat rate for all variables 1-55

Table A-14 Variable ranking, from least to most important

	For a -0.5% perturbation (NN)	For a -0.5% perturbation (MR)	For a 0.5% perturbation (NN)	For a 0.5% perturbation (MR)
1	'FW PUMP(A) SPEED'	'FWPT B HP SV CHEST TEMP'	'TRB HEAT RATE GROSS UNCORR'	'TRB HEAT RATE GROSS UNCORR'
2	'FWPT A HP STEAM INLET'	'FW PUMP(B) SPEED'	'MSR B 2ND STG RHTG STM PRESS'	'FW PUMP(A) SPEED'
3	'TOTAL FEEDWATER FLOW'	'FW PUMP(A) PRESSURE'	'MSR D 2ND STG RHTG STM PRESS'	'TURB STM CHEST PRESS'
4	'FEEDWATER PRESS 2'	'MAIN STEAM PRESSURE (A)'	'FWPT B HP STEAM INLET'	'FW PUMP(B) HEAD'
5	'FW PUMP(A) PRESSURE'	'DRN PUMP(B) ABS DIS PRES'	'FW TO SG PRESS'	'FEEDWATER PRESS 1'
6	'FW PUMP(B) PRESSURE'	'FW PUMP(A) HEAD'	'MSR C 2ND STG RHTG STM PRESS'	'FW PUMP(B) PRESSURE'
7	'RH A 2ND STG STM OUT FLW'	'DRN PUMP(A) ABS DIS PRES'	'TURB STM CHEST PRESS'	'RH A 2ND STG STM OUT FLW'
8	'S/G PRESSURE 1'	'FW PMP A IN ENTH'	'TOTAL STEAM FLOW'	'HTR DRN PUMP(B) DEL PRES'
9	'FW PUMP(A) HEAD'	'FEEDWATER PRESS 2'	'FW PMP A IN ENTH'	'TURB DIFF EXPANSION'
10	'FWPT B HP SV CHEST TEMP'	'CONDENSATE PUMP(B) FLOW'	'S/G 2 ENERGY REMOVAL RATE'	'HTR DRN PUMP(A) DEL PRES'
11	'TOTAL FEED FLOW'	'TOTAL FEED FLOW'	'MAIN STEAM PRESSURE (A)'	'FW TO SG PRESS'
12	'MSR A 2ND STG RHTG STM PRESS'	'MSR D 2ND STG RHTG STM PRESS'	'TURB DIFF EXPANSION'	'MSR B 2ND STG RHTG STM PRESS'
13	'CONDENSATE PUMP(A) FLOW'	'MSR C 1ST STG RHTG STM PRESS'	'DRN PUMP(B) ABS DIS PRES'	'TRB HEAT RATE NET CORR'
14	'FW PUMP(B) FLOW'	'HTR DRAIN PUMP(A) FLOW'	'TRB HEAT RATE NET CORR'	'MSR C 2ND STG RHTG STM PRESS'
15	'HTR DRN PUMP(B) DEL PRES'	'CONDENSATE PUMP(A) FLOW'	'HTR DRAIN PUMP(B) FLOW'	'FP TURB(A) STEAM FLOW'
16	'S/G 1 ENERGY REMOVAL RATE'	'CNDR A CLEAN FACTOR 1'	'HTR DRAIN PUMP(A) FLOW'	'S/G 1 ENERGY REMOVAL RATE'
17	'CNDR A CLEAN FACTOR 1'	'HTR DRAIN PUMP(B) FLOW'	'HTR DRN PUMP(A) DEL PRES'	'MSR A 2ND STG RHTG STM PRESS'
18	'MSR C 1ST STG RHTG STM PRESS'	'TOTAL STEAM FLOW'	'FEEDWATER PRESS 1'	'S/G PRESSURE 1'
19	'AVG MAIN STEAM PRESSURE'	'AVG MAIN STEAM PRESSURE'	'CONDENSATE PUMP(C) FLOW'	'FW PUMP(B) FLOW'
20	'FW PUMP(B) SPEED'	'FWPT A HP STEAM INLET'	'TOTAL DRAIN & COND FLOW'	'FW PUMP(A) FLOW'
21	'FW PUMP(A) FLOW'	'FW PMP A SUCT FLOW DP'	'FP TURB(A) STEAM FLOW'	'CNDR A CLEAN FACTOR 2'
22	'SG 1 LEVEL'	'HTR NO 2B TERM TEMP DIFF'	'DRN PUMP(A) ABS DIS PRES'	'TOTAL FEEDWATER FLOW'
23	'CONDENSATE PUMP(B) FLOW'	'CNDR B CLEAN FACTOR 1'	'CNDR B CLEAN FACTOR 2'	'SG 1 LEVEL'
24	'FW PUMP(B) HEAD'	'CNDR B CLEAN FACTOR 2'	'CNDR C CLEAN FACTOR 1'	'FWPT B HP STEAM INLET'
25	'CNDR C CLEAN FACTOR 2'	'HTR NO 2C TERM TEMP DIFF'	'CNDR A CLEAN FACTOR 2'	'TOTAL DRAIN & COND FLOW'
26	'FW PMP A SUCT FLOW DP'	'HTR NO 1B TERM TEMP DIFF'	'HTR NO 1B TERM TEMP DIFF'	'CNDR C CLEAN FACTOR 2'
27	'HTR NO 2C TERM TEMP DIFF'	'CONDENSATE PUMP(C) FLOW'	'HTR NO 2B TERM TEMP DIFF'	'S/G 2 ENERGY REMOVAL RATE'
28	'CNDR B CLEAN FACTOR 1'	'CNDR C CLEAN FACTOR 1'	'CNDR B CLEAN FACTOR 1'	'CNDR C CLEAN FACTOR 1'

	For a -0.5% perturbation (NN)	For a -0.5% perturbation (MR)	For a 0.5% perturbation (NN)	For a 0.5% perturbation (MR)
29	'HTR NO 2B TERM TEMP DIFF'	'S/G 2 ENERGY REMOVAL RATE'	'HTR NO 2C TERM TEMP DIFF'	'CONDENSATE PUMP(C) FLOW'
30	'HTR NO 1B TERM TEMP DIFF'	'CNDR C CLEAN FACTOR 2'	'FW PMP A SUCT FLOW DP'	'HTR NO 1B TERM TEMP DIFF'
31	'CNDR A CLEAN FACTOR 2'	'TOTAL DRAIN & COND FLOW'	'CNDR C CLEAN FACTOR 2'	'HTR NO 2C TERM TEMP DIFF'
32	'CNDR C CLEAN FACTOR 1'	'FWPT B HP STEAM INLET'	'CONDENSATE PUMP(B) FLOW'	'CNDR B CLEAN FACTOR 2'
33	'CNDR B CLEAN FACTOR 2'	'SG 1 LEVEL'	'SG 1 LEVEL'	'CNDR B CLEAN FACTOR 1'
34	'DRN PUMP(A) ABS DIS PRES'	'TOTAL FEEDWATER FLOW'	'FW PUMP(B) HEAD'	'HTR NO 2B TERM TEMP DIFF'
35	'FP TURB(A) STEAM FLOW'	'CNDR A CLEAN FACTOR 2'	'FW PUMP(A) FLOW'	'FW PMP A SUCT FLOW DP'
36	'CONDENSATE PUMP(C) FLOW'	'FW PUMP(A) FLOW'	'MSR C 1ST STG RHTG STM PRESS'	'FWPT A HP STEAM INLET'
37	'TOTAL DRAIN & COND FLOW'	'FW PUMP(B) FLOW'	'FW PUMP(B) SPEED'	'AVG MAIN STEAM PRESSURE'
38	'TRB HEAT RATE NET CORR'	'S/G PRESSURE 1'	'AVG MAIN STEAM PRESSURE'	'TOTAL STEAM FLOW'
39	'FEEDWATER PRESS 1'	'MSR A 2ND STG RHTG STM PRESS'	'CNDR A CLEAN FACTOR 1'	'HTR DRAIN PUMP(B) FLOW'
40	'HTR DRAIN PUMP(A) FLOW'	'S/G 1 ENERGY REMOVAL RATE'	'S/G 1 ENERGY REMOVAL RATE'	'CNDR A CLEAN FACTOR 1'
41	'HTR DRN PUMP(A) DEL PRES'	'FP TURB(A) STEAM FLOW'	'HTR DRN PUMP(B) DEL PRES'	'CONDENSATE PUMP(A) FLOW'
42	'HTR DRAIN PUMP(B) FLOW'	'MSR C 2ND STG RHTG STM PRESS'	'FW PUMP(B) FLOW'	'HTR DRAIN PUMP(A) FLOW'
43	'MAIN STEAM PRESSURE (A)'	'TRB HEAT RATE NET CORR'	'CONDENSATE PUMP(A) FLOW'	'MSR C 1ST STG RHTG STM PRESS'
44	'DRN PUMP(B) ABS DIS PRES'	'MSR B 2ND STG RHTG STM PRESS'	'FWPT B HP SV CHEST TEMP'	'MSR D 2ND STG RHTG STM PRESS'
45	'TURB DIFF EXPANSION'	'FW TO SG PRESS'	'TOTAL FEED FLOW'	'TOTAL FEED FLOW'
46	'S/G 2 ENERGY REMOVAL RATE'	'HTR DRN PUMP(A) DEL PRES'	'FW PUMP(A) HEAD'	'CONDENSATE PUMP(B) FLOW'
47	'FW PMP A IN ENTH'	'TURB DIFF EXPANSION'	'S/G PRESSURE 1'	'FEEDWATER PRESS 2'
48	'TOTAL STEAM FLOW'	'HTR DRN PUMP(B) DEL PRES'	'RH A 2ND STG STM OUT FLW'	'FW PMP A IN ENTH'
49	'MSR C 2ND STG RHTG STM PRESS'	'RH A 2ND STG STM OUT FLW'	'MSR A 2ND STG RHTG STM PRESS'	'DRN PUMP(A) ABS DIS PRES'
50	'TURB STM CHEST PRESS'	'FW PUMP(B) PRESSURE'	'FW PUMP(B) PRESSURE'	'FW PUMP(A) HEAD'
51	'FWPT B HP STEAM INLET'	'FEEDWATER PRESS 1'	'FEEDWATER PRESS 2'	'DRN PUMP(B) ABS DIS PRES'
52	'FW TO SG PRESS'	'FW PUMP(B) HEAD'	'FW PUMP(A) PRESSURE'	'MAIN STEAM PRESSURE (A)'
53	'MSR D 2ND STG RHTG STM PRESS'	'TURB STM CHEST PRESS'	'TOTAL FEEDWATER FLOW'	'FW PUMP(A) PRESSURE'
54	'MSR B 2ND STG RHTG STM PRESS'	'FW PUMP(A) SPEED'	'FWPT A HP STEAM INLET'	'FW PUMP(B) SPEED'
55	'TRB HEAT RATE GROSS UNCORR'	'TRB HEAT RATE GROSS UNCORR'	'FW PUMP(A) SPEED'	'FWPT B HP SV CHEST TEMP'

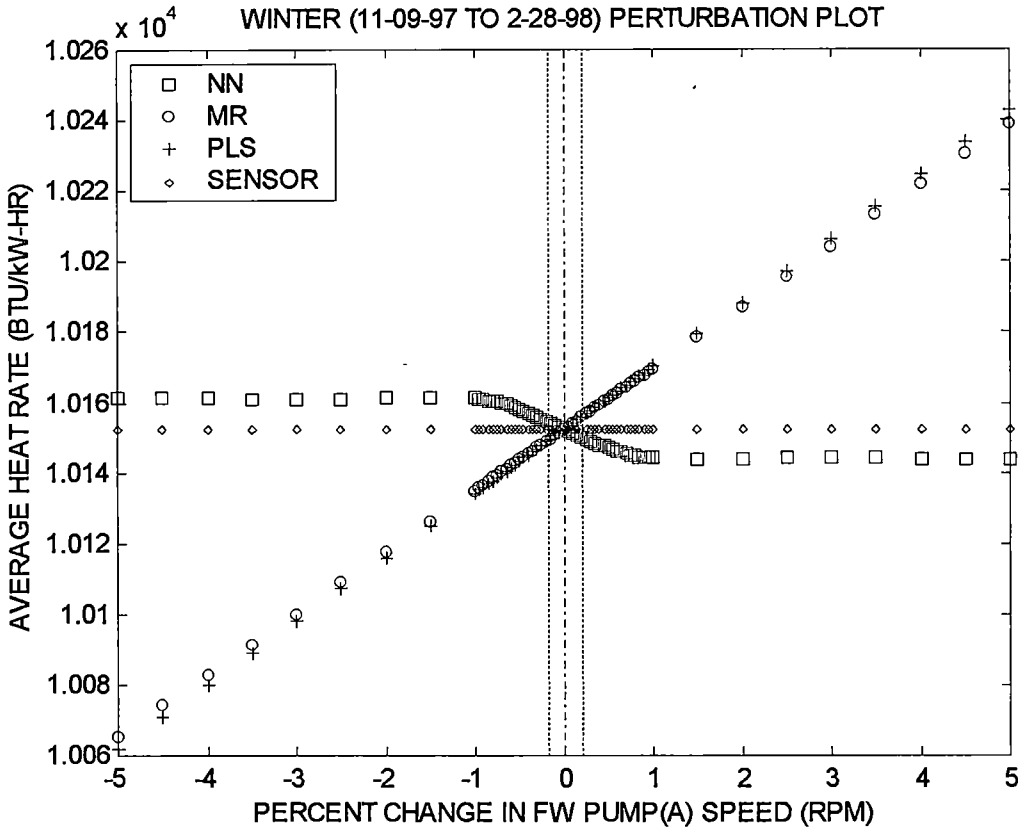

Figure A-87 Winter perturbation plot, FW Pump(A) Speed

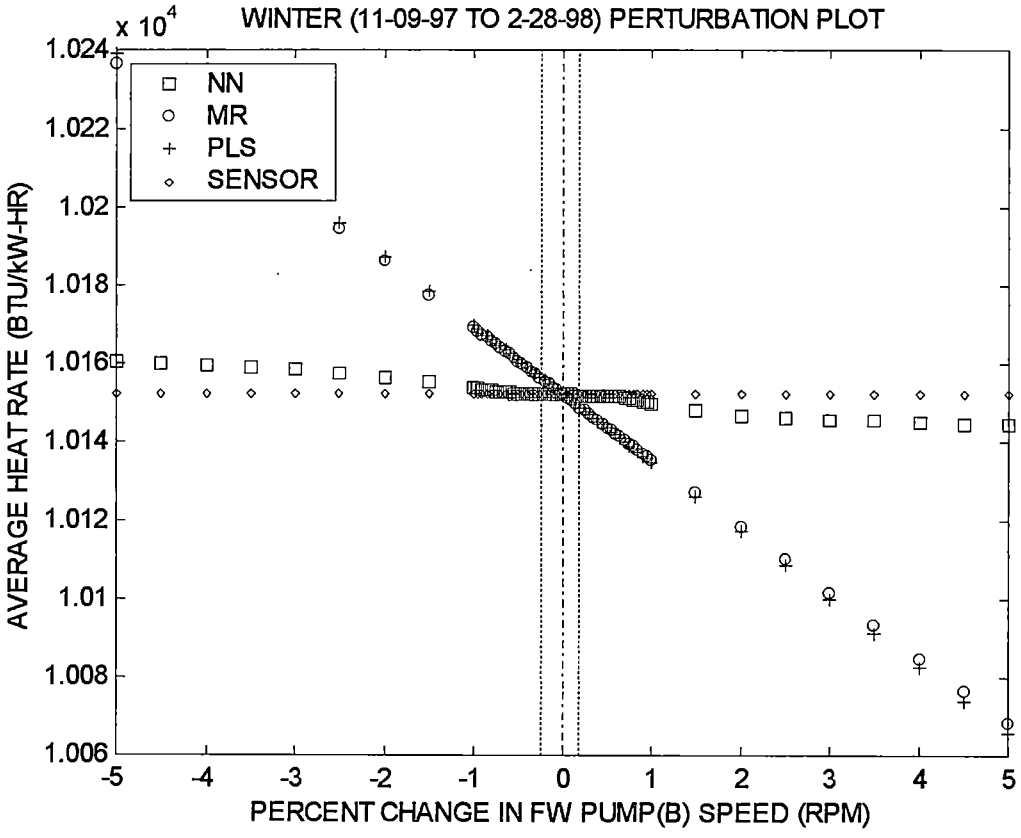

Figure A-88 Winter perturbation plot, FW Pump(B) Speed

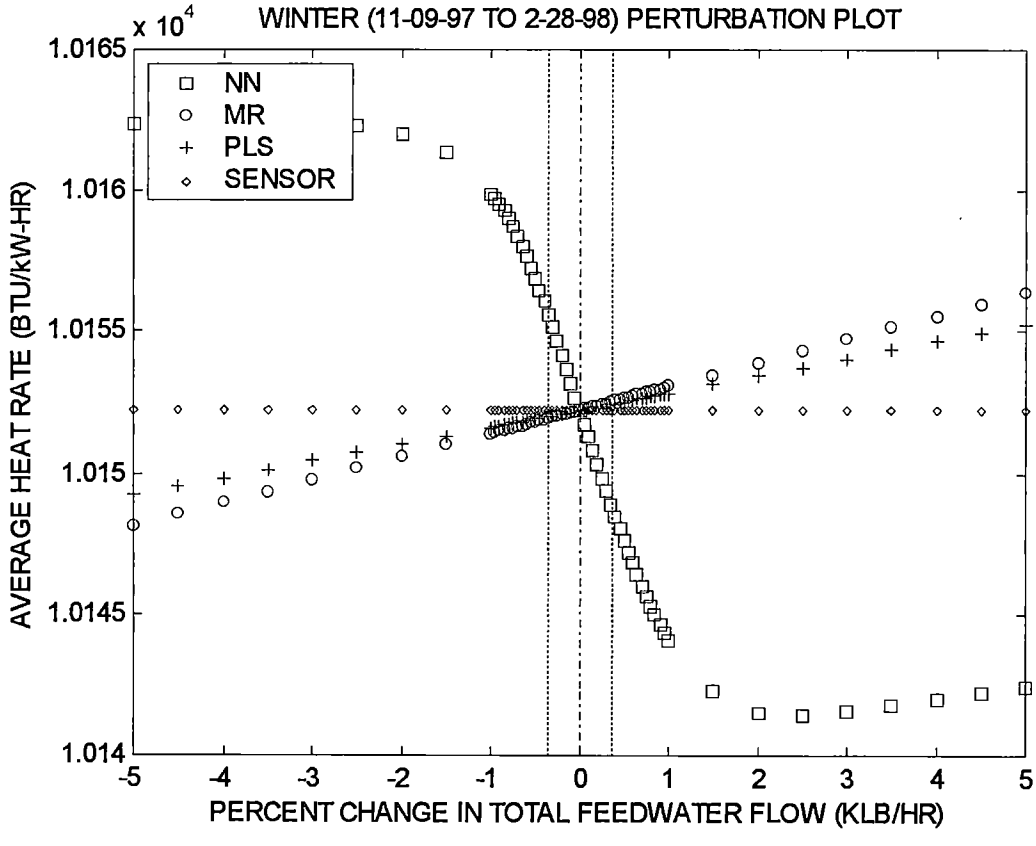

Figure A-89 Winter perturbation plot, Total Feedwater Flow

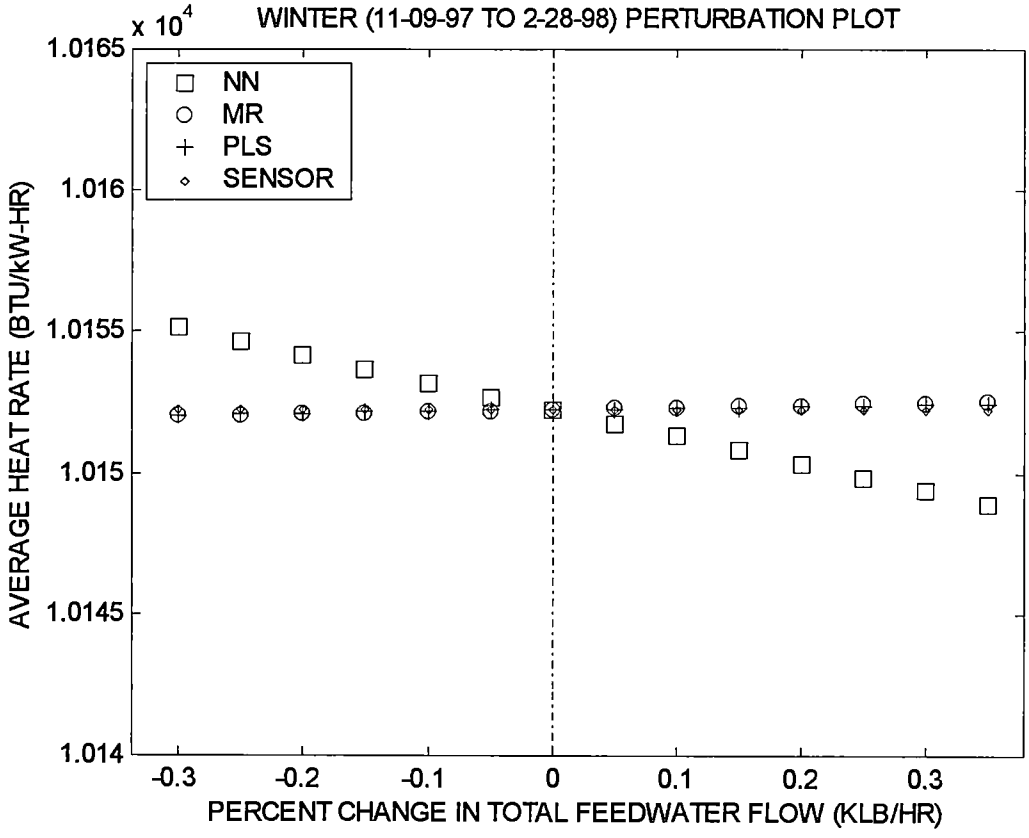

Figure A-90 Winter perturbation plot, Total Feedwater Flow

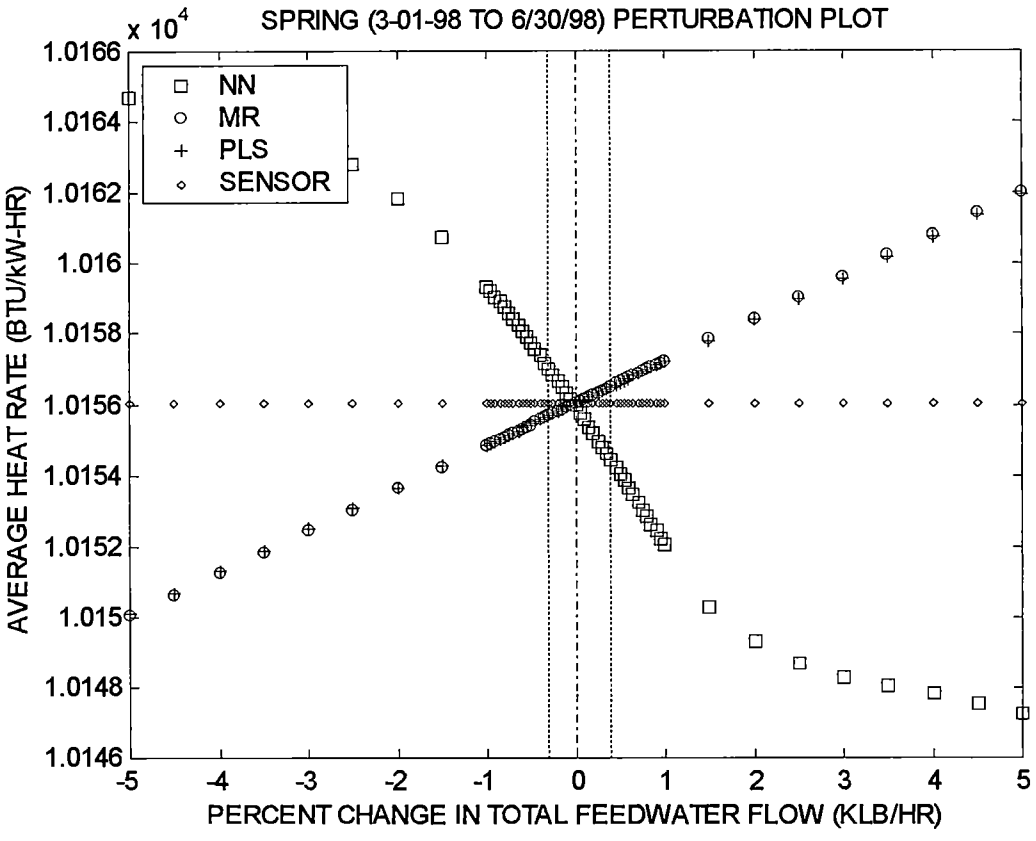

Figure A-91 Spring perturbation plot, Total Feedwater Flow

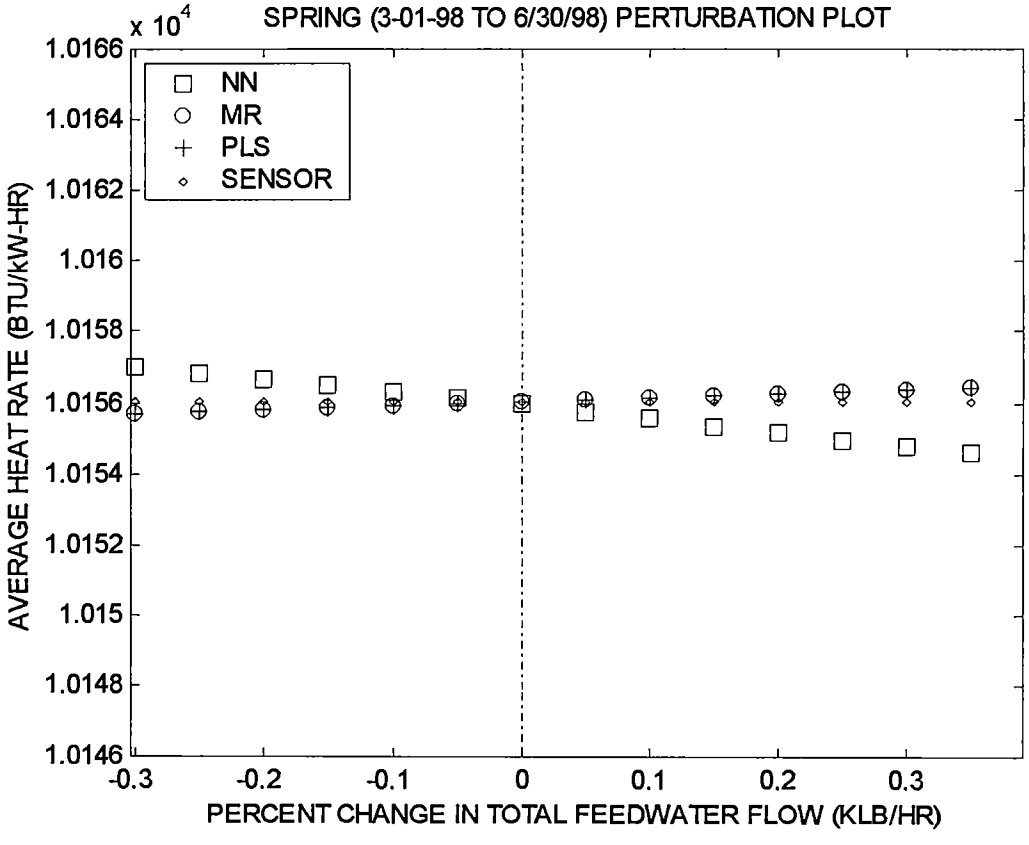

Figure A-92 Spring perturbation plot, Total Feedwater Flow

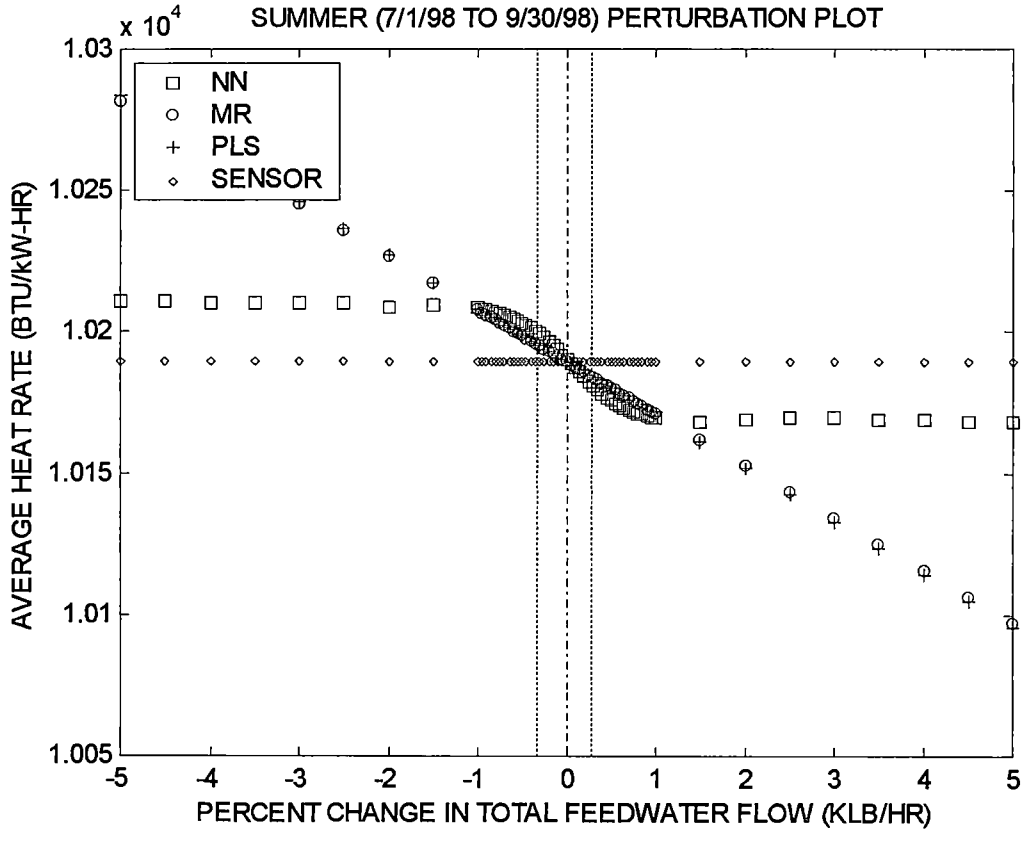

Figure A-93 Summer perturbation plot, Total Feedwater Flow

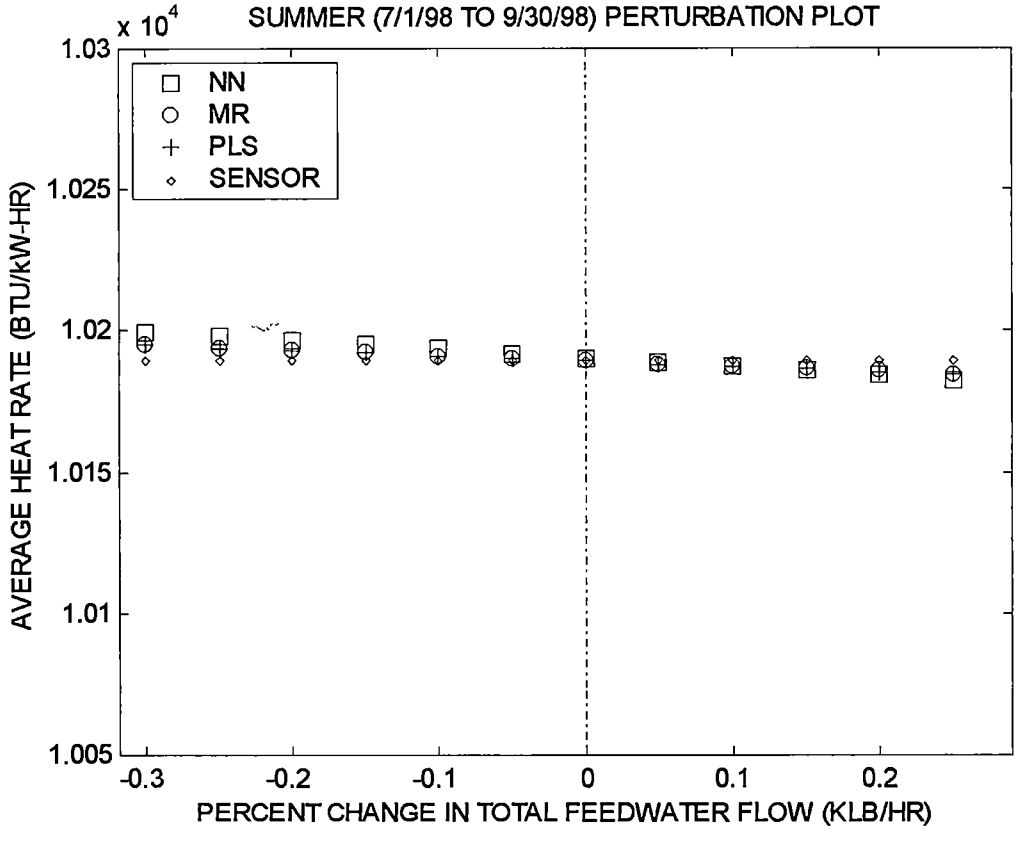

Figure A-94 Summer perturbation plot, Total Feedwater Flow

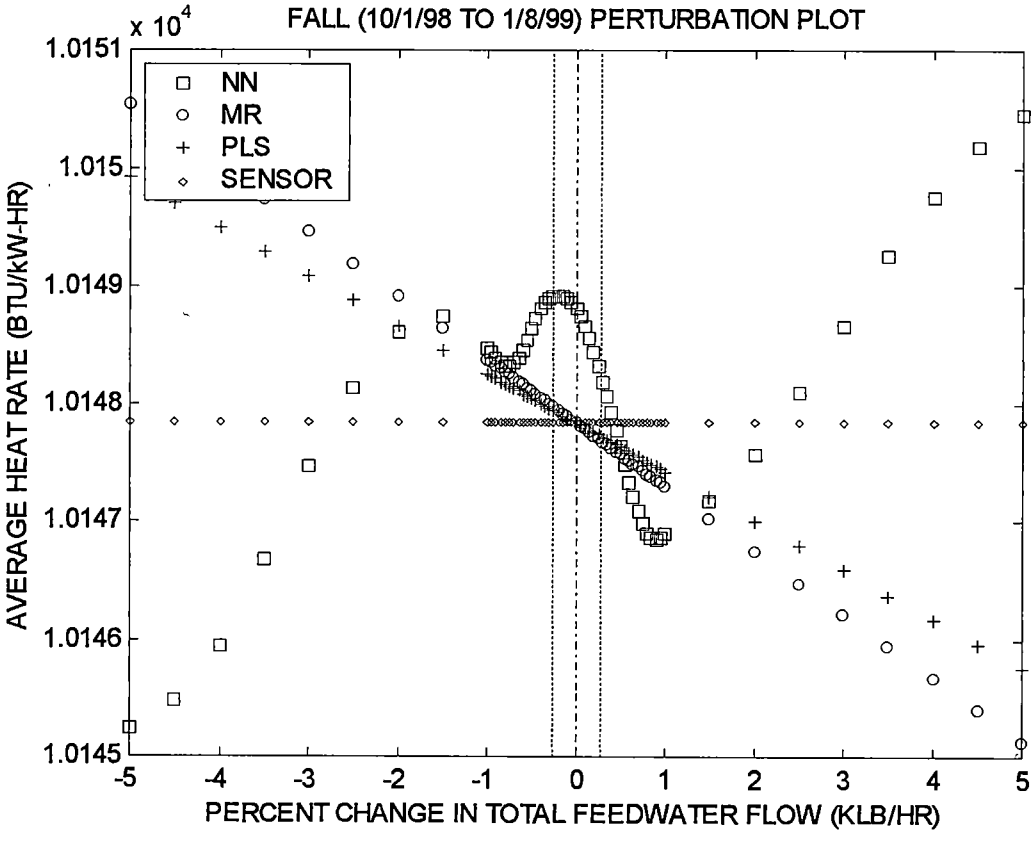

Figure A-95 Fall perturbation plot, Total Feedwater Flow

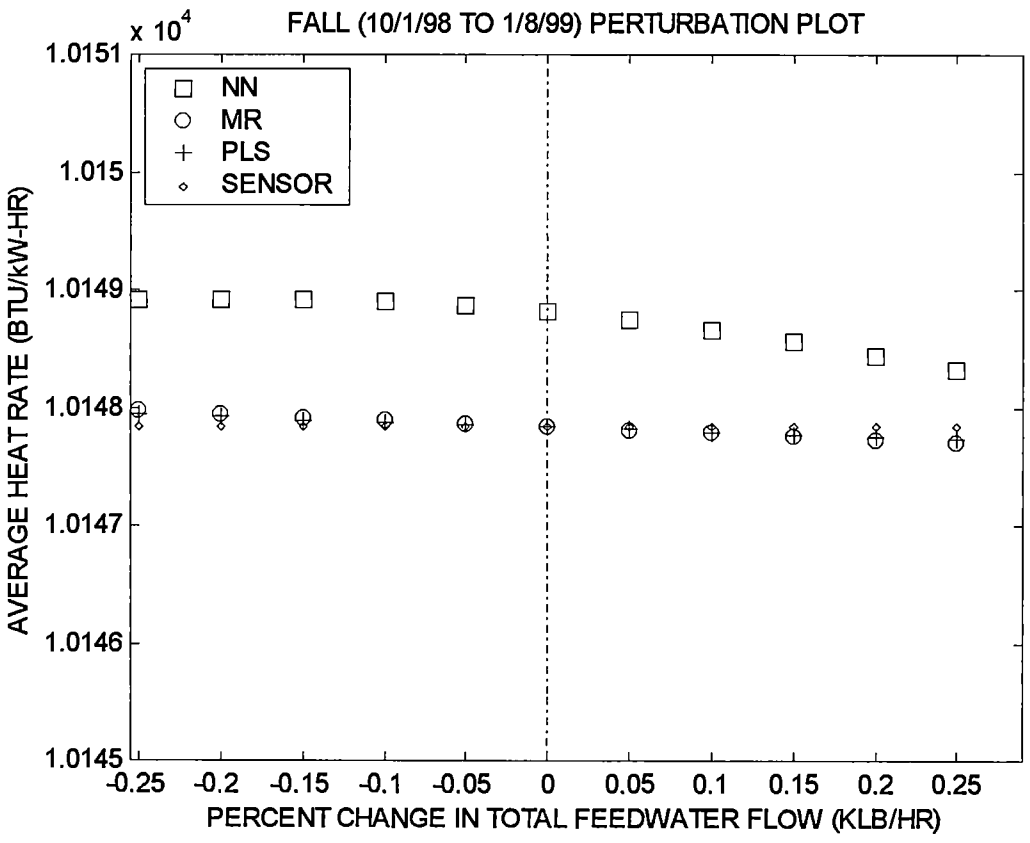

Figure A-96 Fall perturbation plot, Total Feedwater Flow

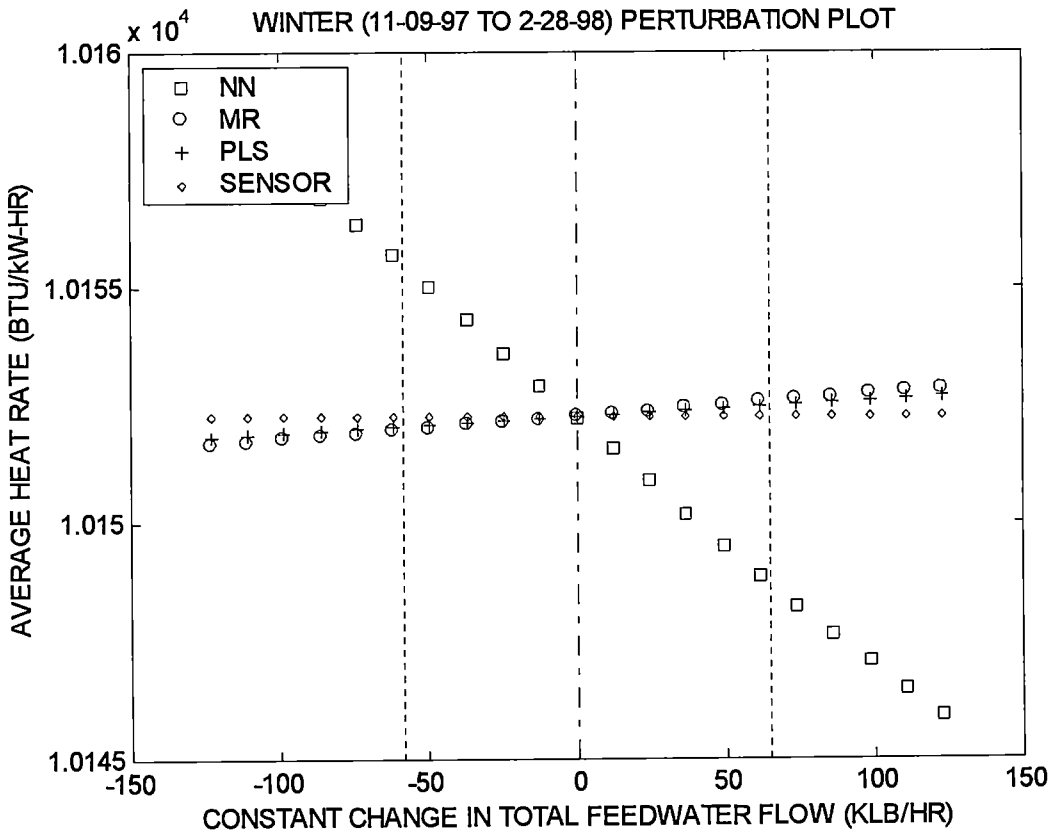

Figure A-97 Winter perturbation plot, Total Feedwater Flow (for constant change)

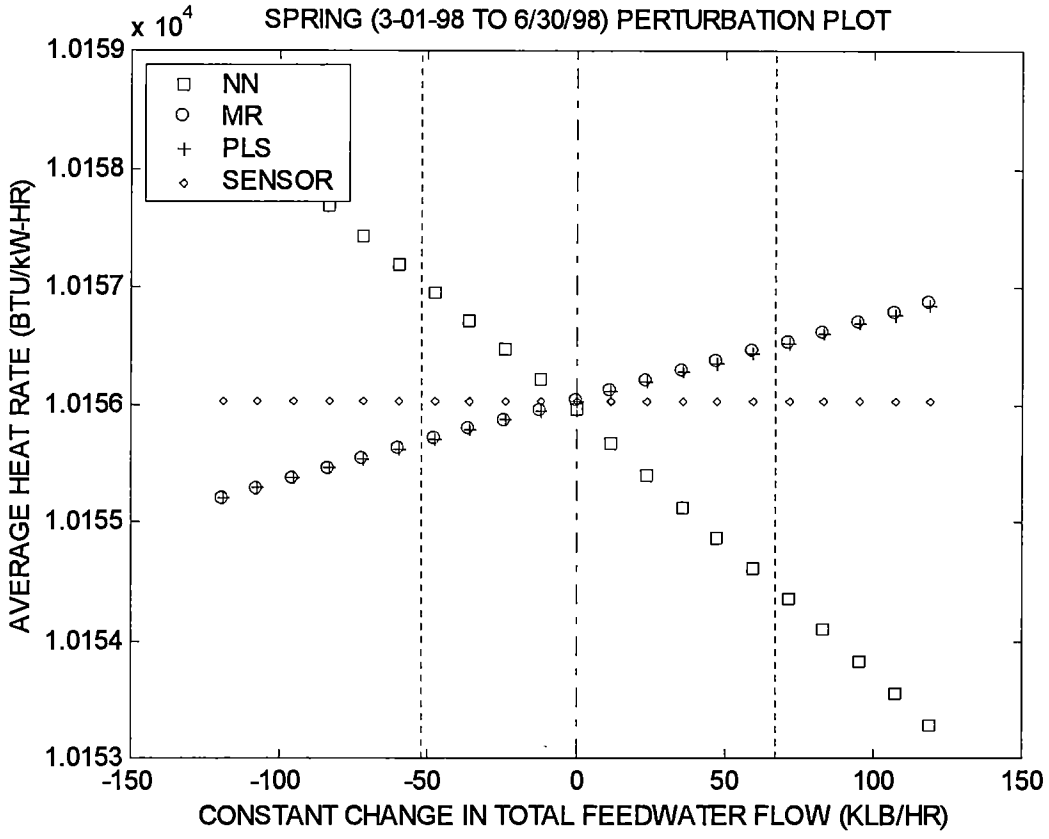

Figure A-98 Spring perturbation plot, Total Feedwater Flow (for constant change)

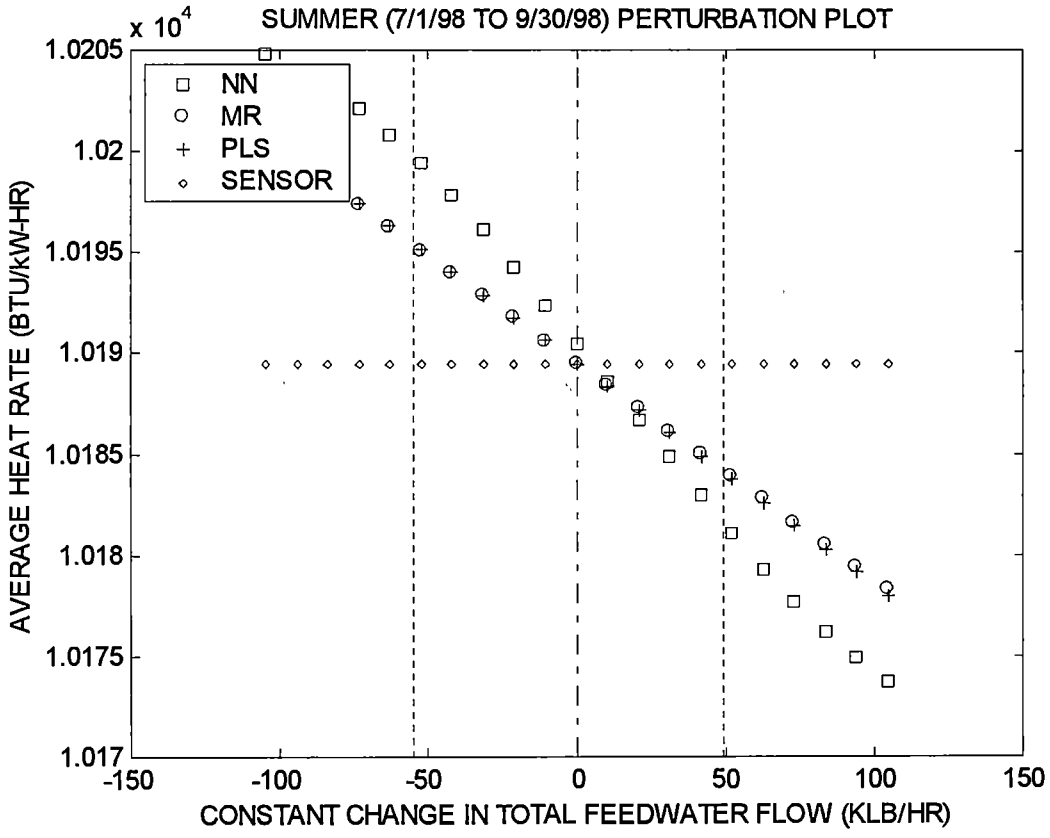

Figure A-99 Summer perturbation plot, Total Feedwater Flow (for constant change)

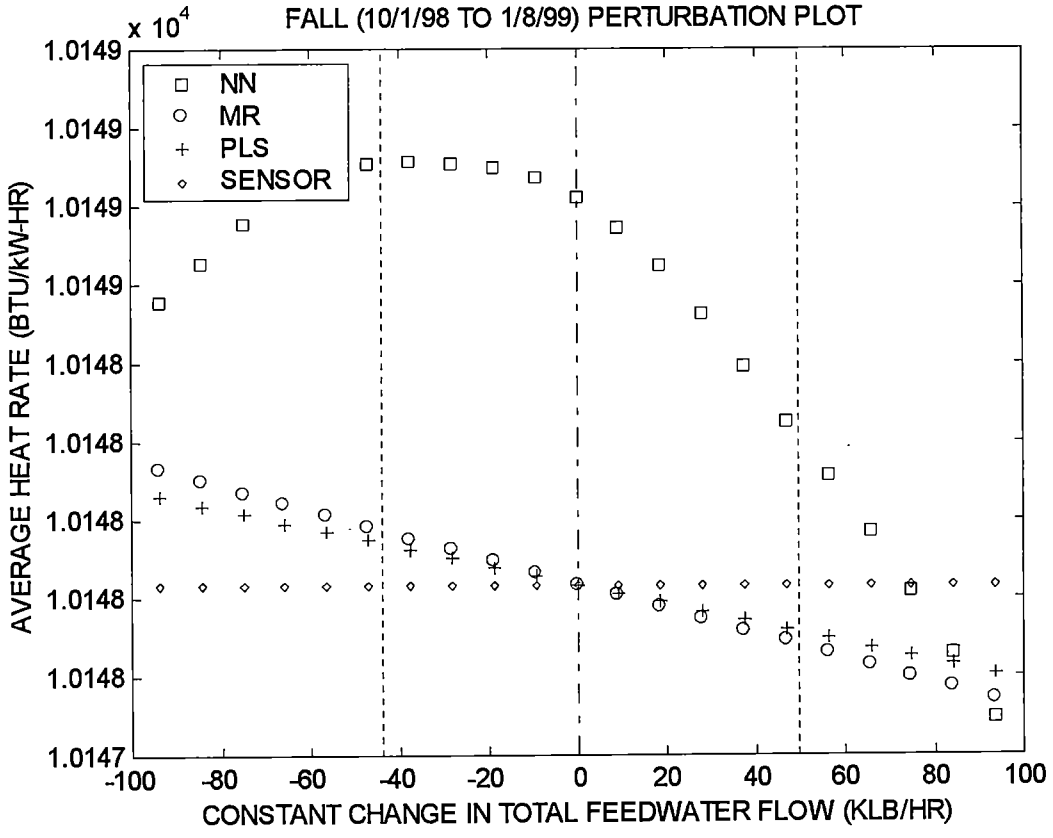

Figure A-100 Fall perturbation plot, Total Feedwater Flow (for constant change)

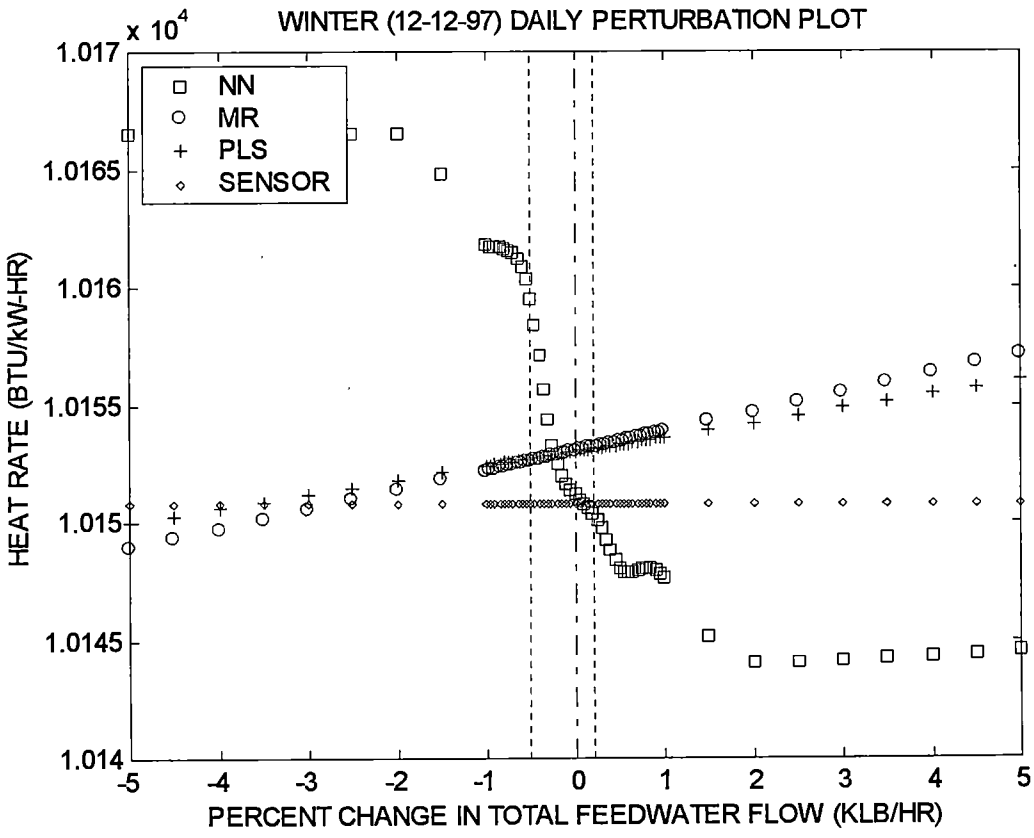

Figure A-101 Winter daily perturbation plot, Feedwater Flow (12-12-97)

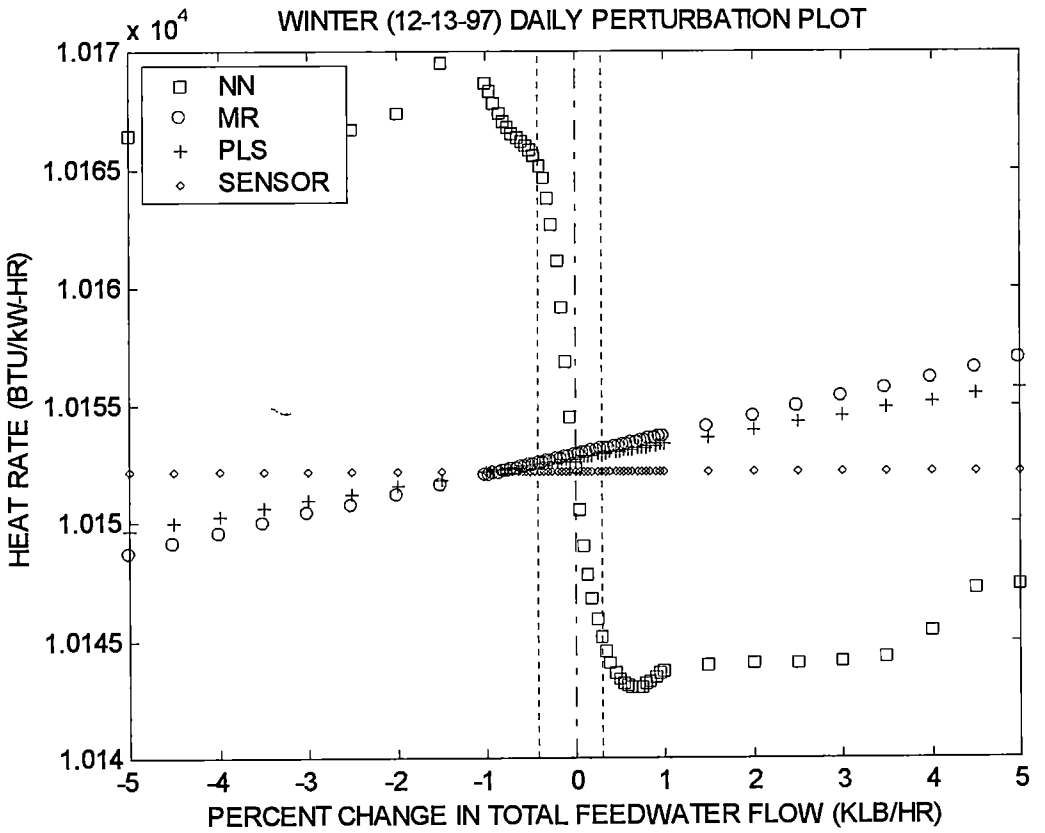

Figure A-102 Winter daily perturbation plot, Feedwater Flow (12-13-97)

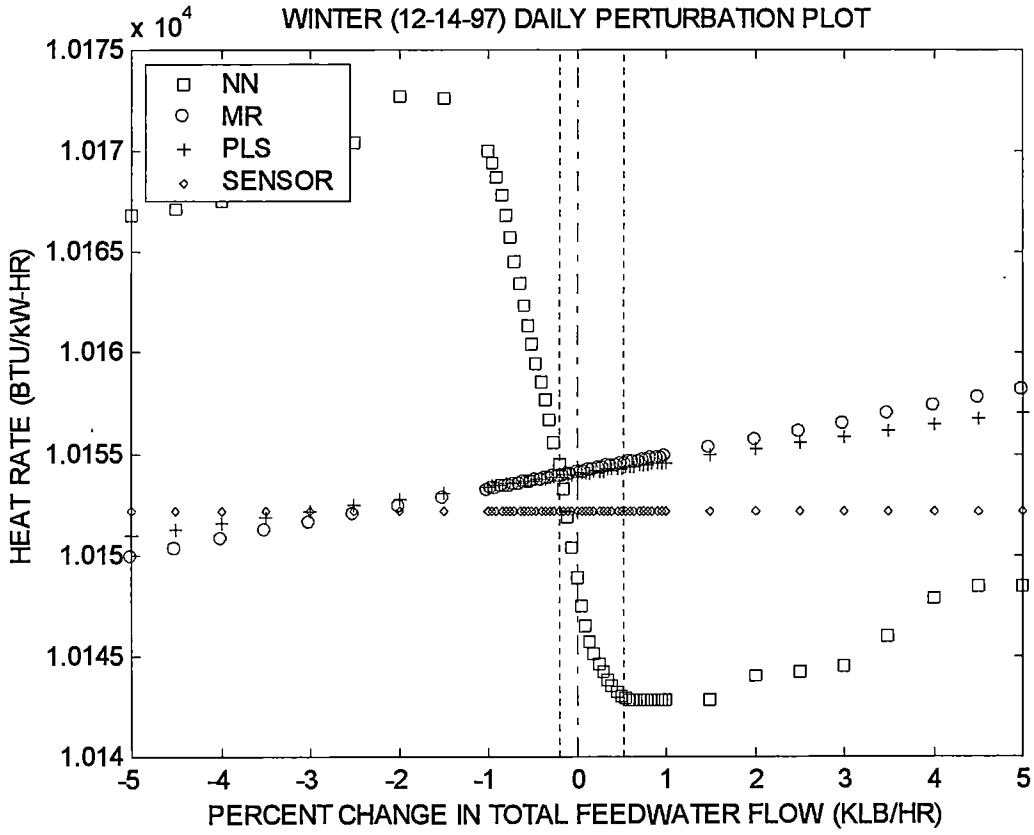

Figure A-103 Winter daily perturbation plot, Feedwater Flow (12-14-97)

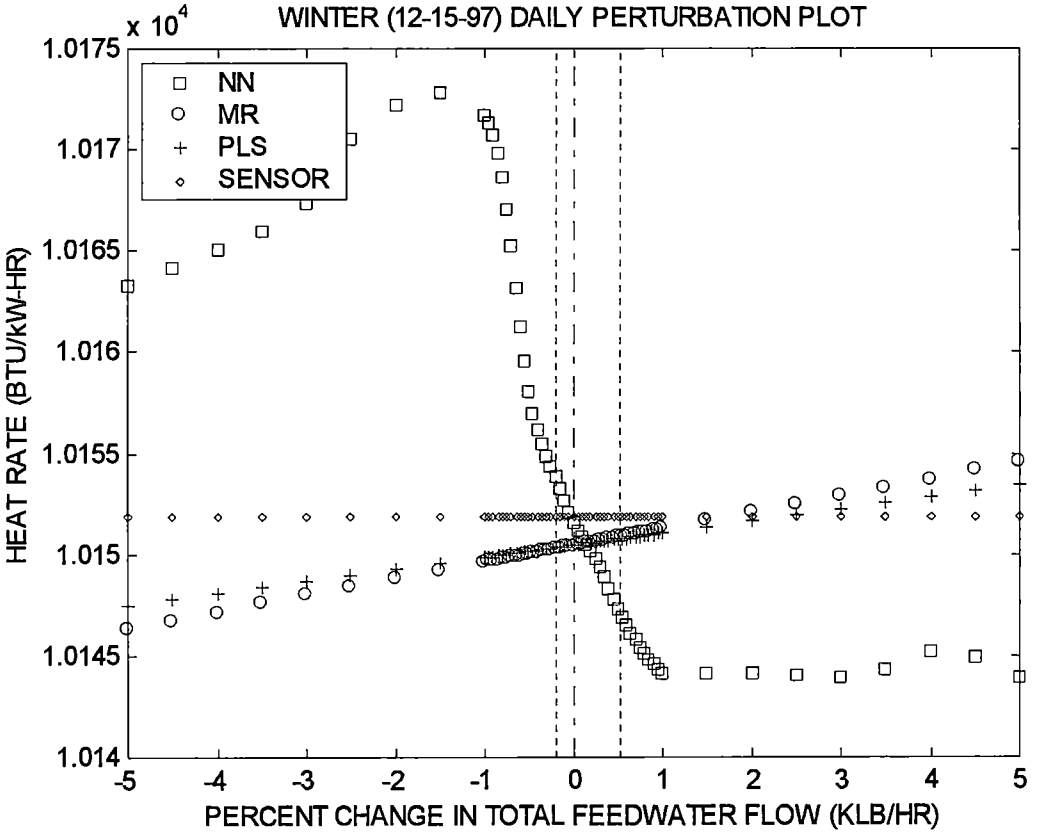

Figure A-104 Winter daily perturbation plot, Feedwater Flow (12-15-97)

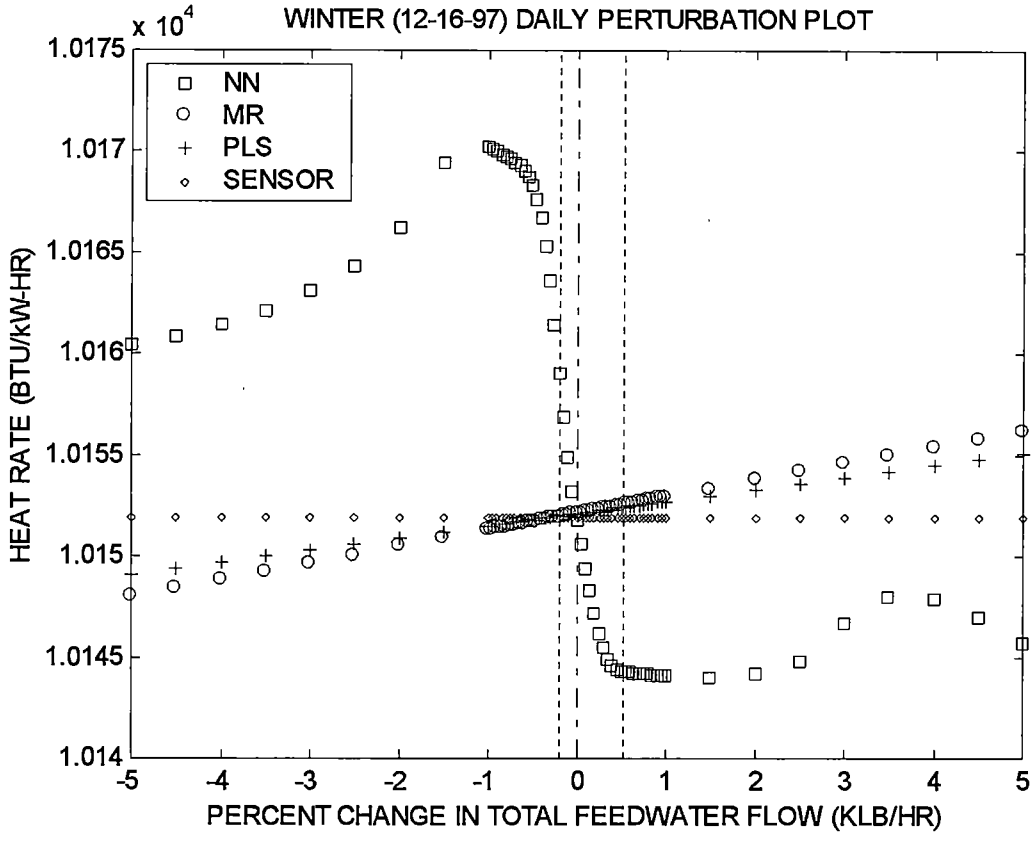

Figure A-105 Winter daily perturbation plot, Feedwater Flow (12-16-97)

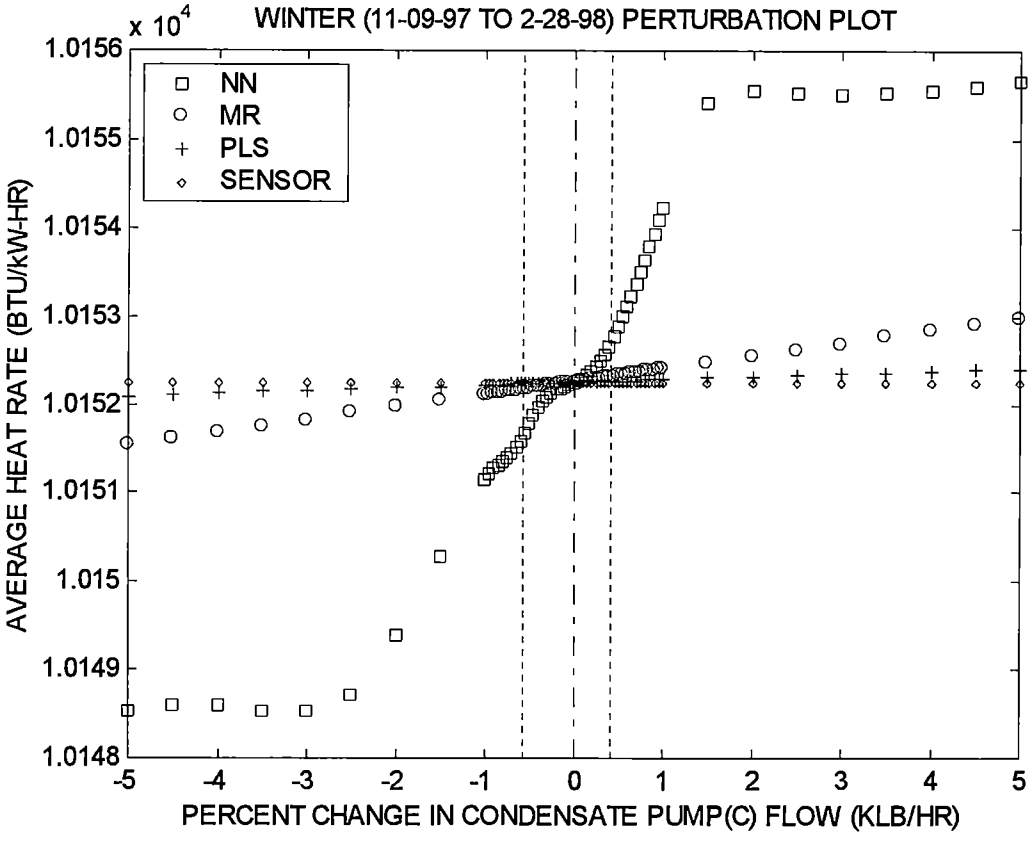

Figure A-106 Winter perturbation plot, Condensate pump(C) flow

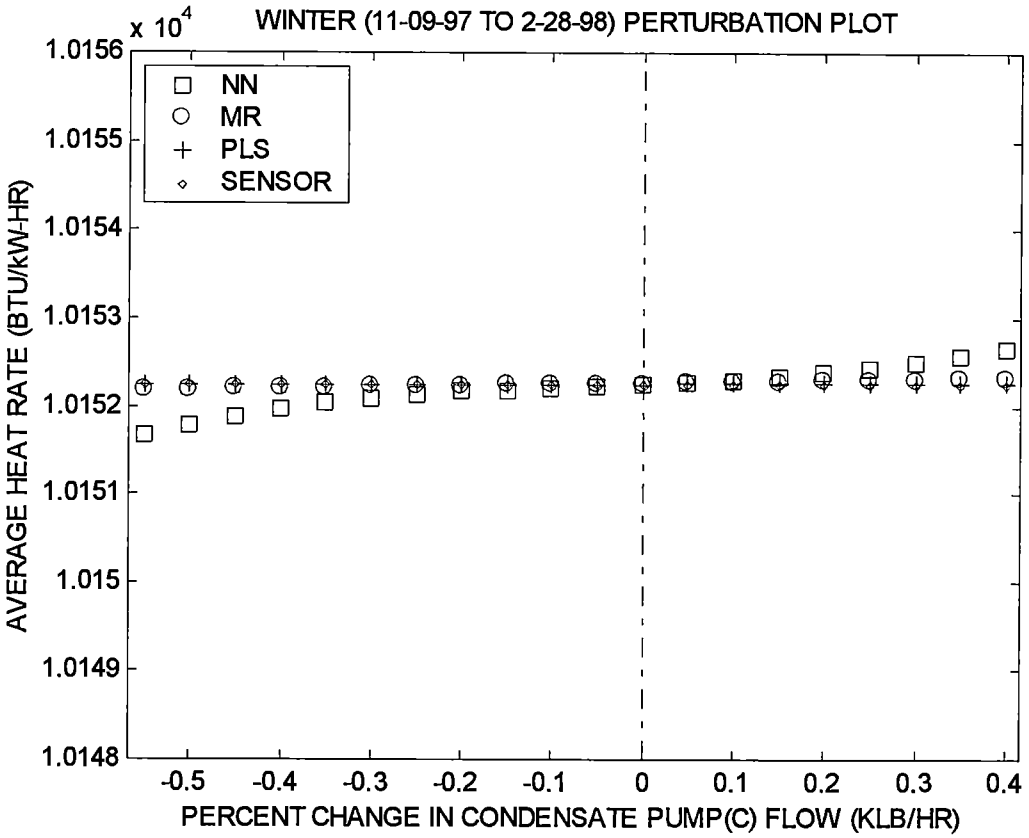

Figure A-107 Winter perturbation plot, Condensate pump(C) flow

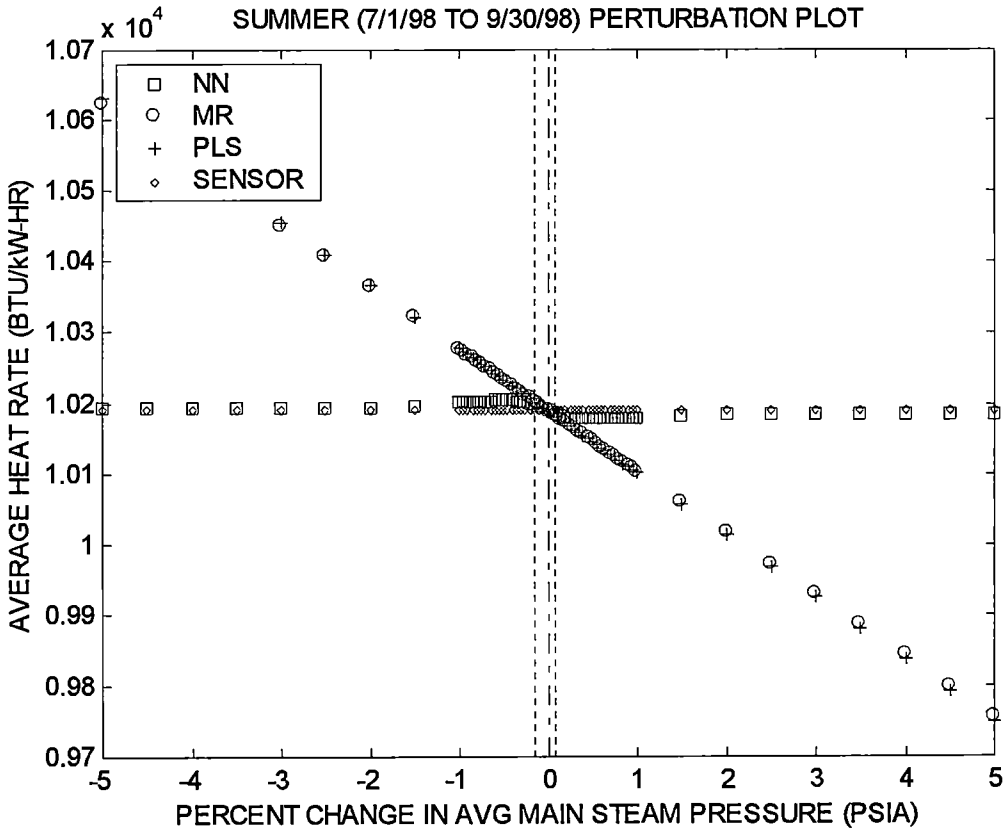

Figure A-108 Summer perturbation plot, Avg Main Steam Pressure

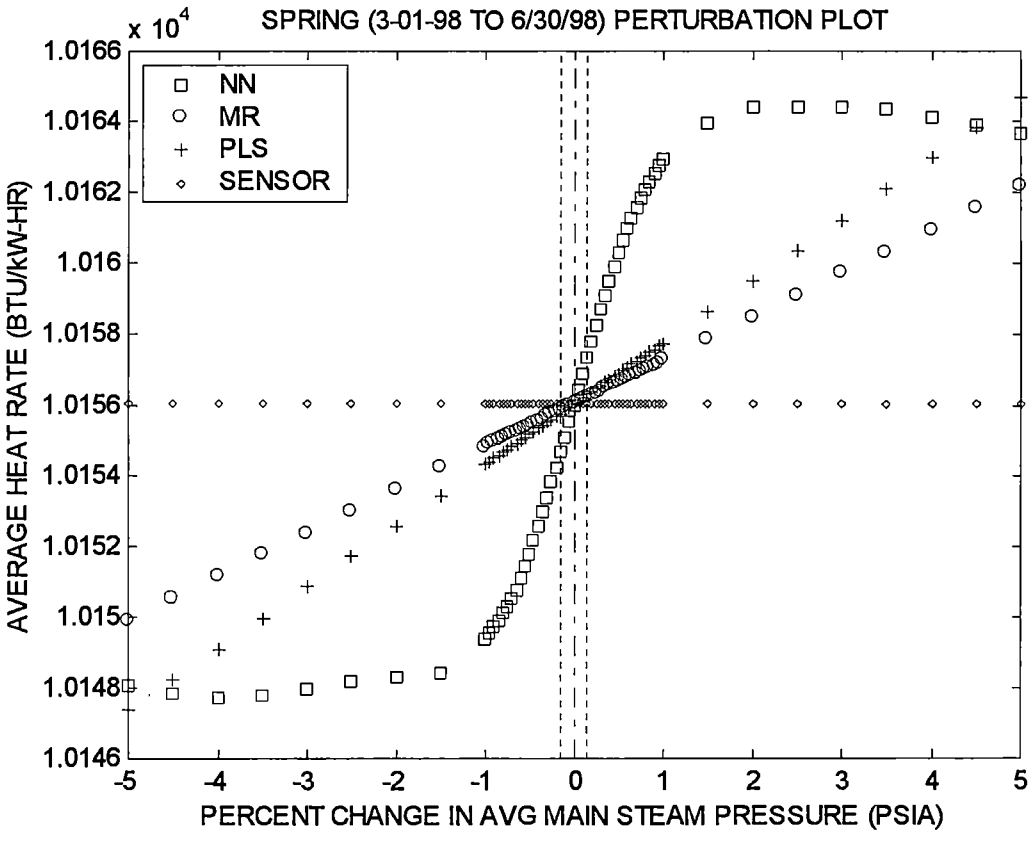

Figure A-109 Spring perturbation plot, Avg Main Steam Pressure

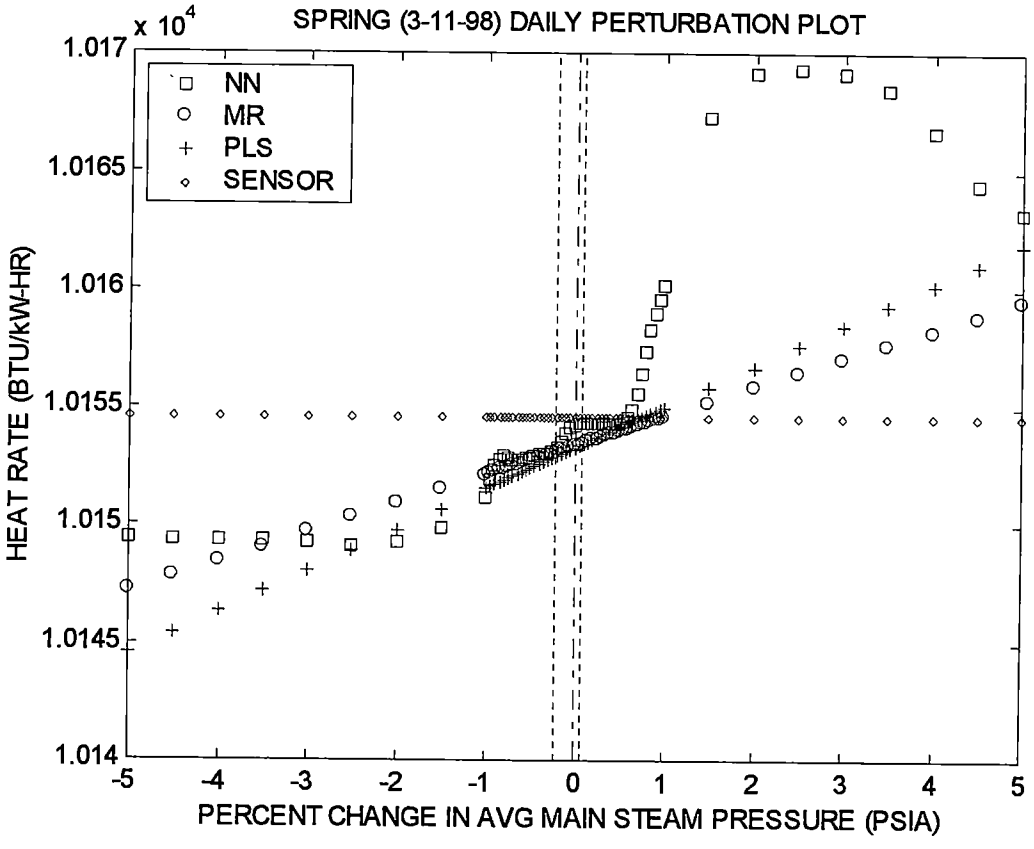

Figure A-110 Spring daily perturbation plot, Avg Main Steam Pressure (3-11-98)

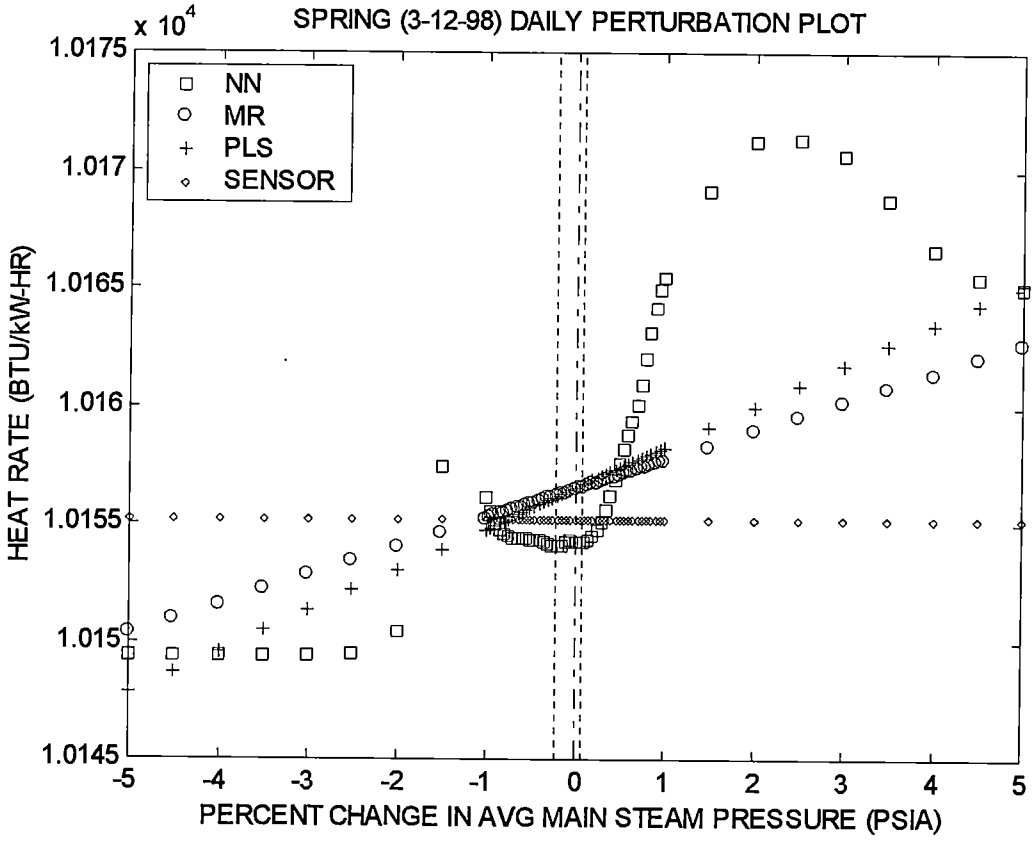

Figure A-111 Spring daily perturbation plot, Avg Main Steam Pressure (3-12-98)

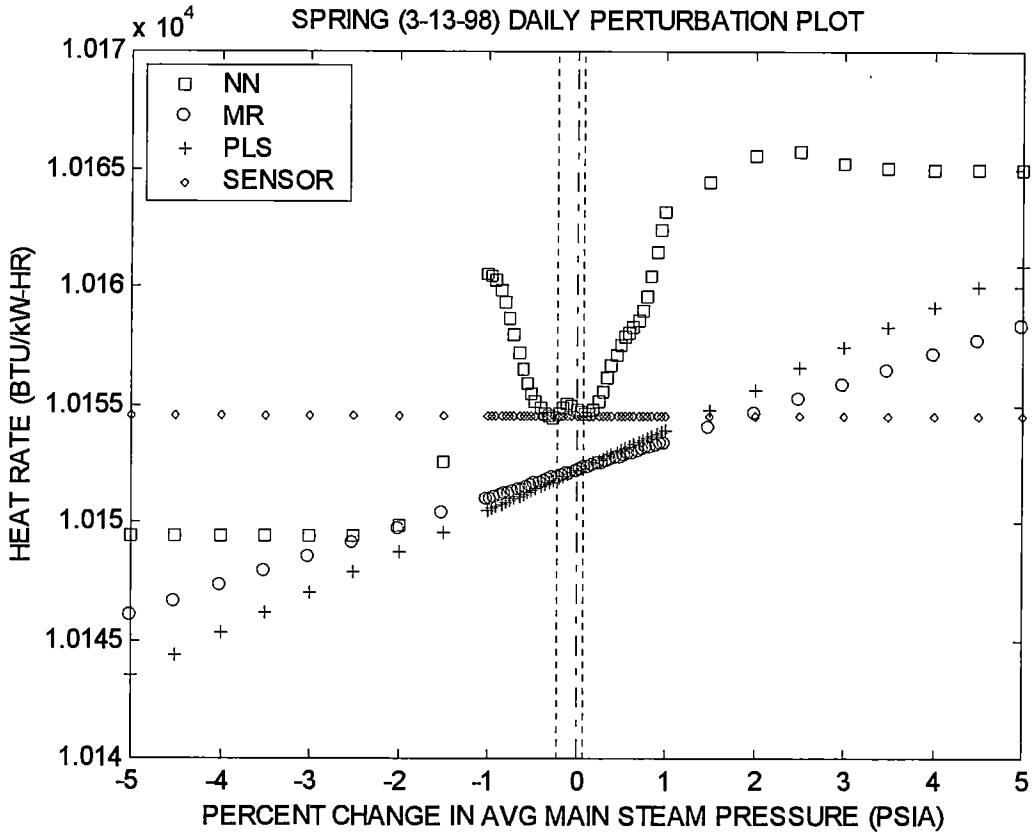

Figure A-112 Spring daily perturbation plot, Avg Main Steam Pressure (3-13-98)

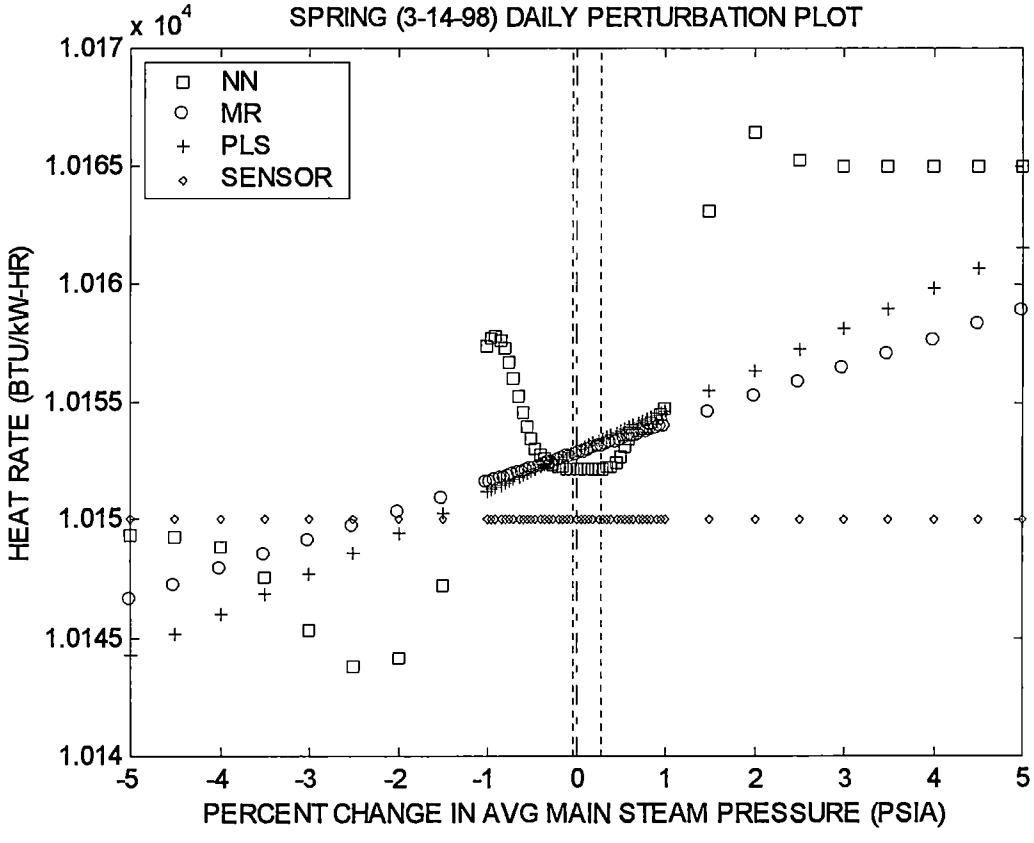

Figure A-113 Spring daily perturbation plot, Avg Main Steam Pressure (3-14-98)

Table A-15 Comparison of Plant Parameters to Reference Directionalities

SETPOINT VARIABLE	Y–CORRECT DIRECTIONALITY N-WRONG DIRECTIONALITY NN | MR/PLS								REFERENCE DIRECTIONALITY[1]	COMMENTS
	W	Sp	S	F	W	Sp	S	F		
HTR NO 1B TERM TEMP DIFF	Y	Y	Y	N	Y	Y	N	Y	INCREASE	Modeling error
HTR NO 2B TERM TEMP DIFF	Y	N	Y	Y	N	Y	Y	Y	INCREASE	Modeling error
HTR NO 2C TERM TEMP DIFF	Y	Y	Y	Y	Y	Y	N	Y	INCREASE	Modeling error
AVG MAIN STEAM PRESSURE	Y	N	Y	Y	Y	N	Y	Y	DECREASE	Small range, modeling error
S/G PRESSURE 1	Y	Y	Y	Y	N	N	Y	N	DECREASE	
TRB HEAT RATE GROSS UNCORR	Y	Y	Y	Y	Y	Y	Y	Y	INCREASE	
TRB HEAT RATE NET CORR	Y	Y	Y	Y	Y	N	Y	Y	INCREASE	
TOTAL FEED FLOW	Y	Y	Y	Y	Y	Y	N	Y	DECREASE	
TOTAL FEEDWATER FLOW	Y	Y	Y	Y	N	N	Y	Y	DECREASE	
TOTAL STEAM FLOW	N	N	N	N	Y	N	Y	Y	DECREASE	Reference direction assumed, NN modeling error

[1] – Reference directionalities are defined as the increase/decrease of heat rate given an increase in magnitude of the predictor variable.

W – Winter; Sp – Spring; S – Summer; F - Fall

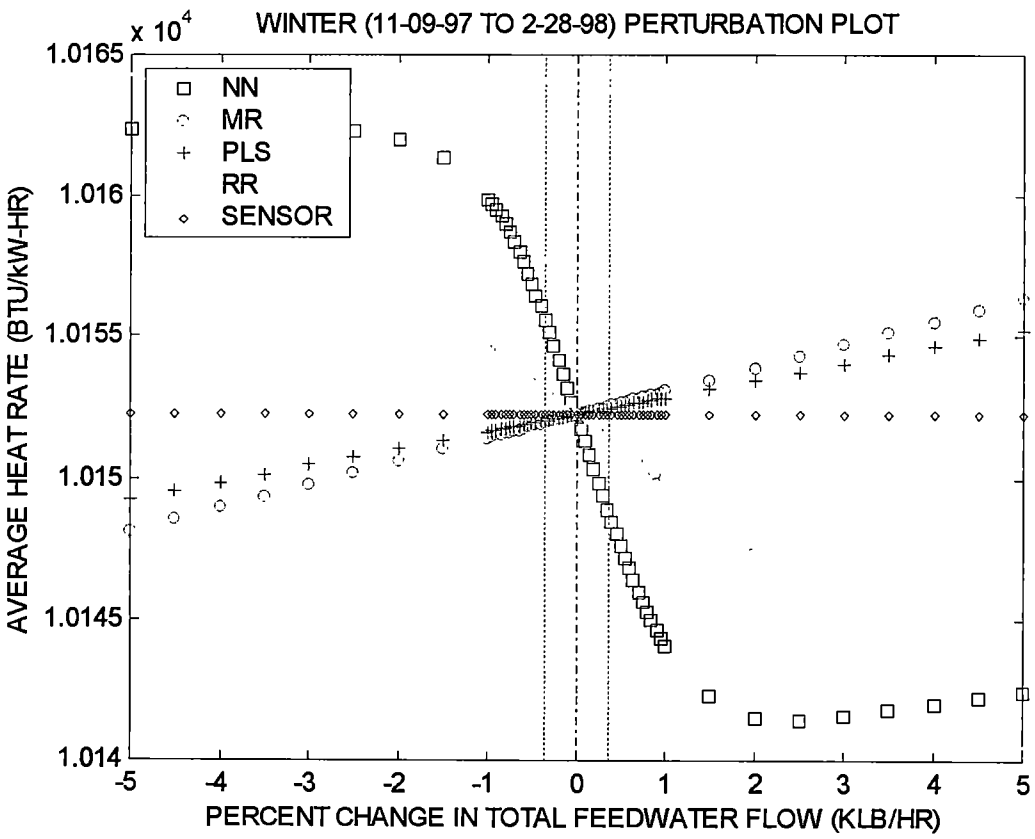

Figure A-114 Winter perturbation plot, Total Feedwater Flow, with Ridge Regression

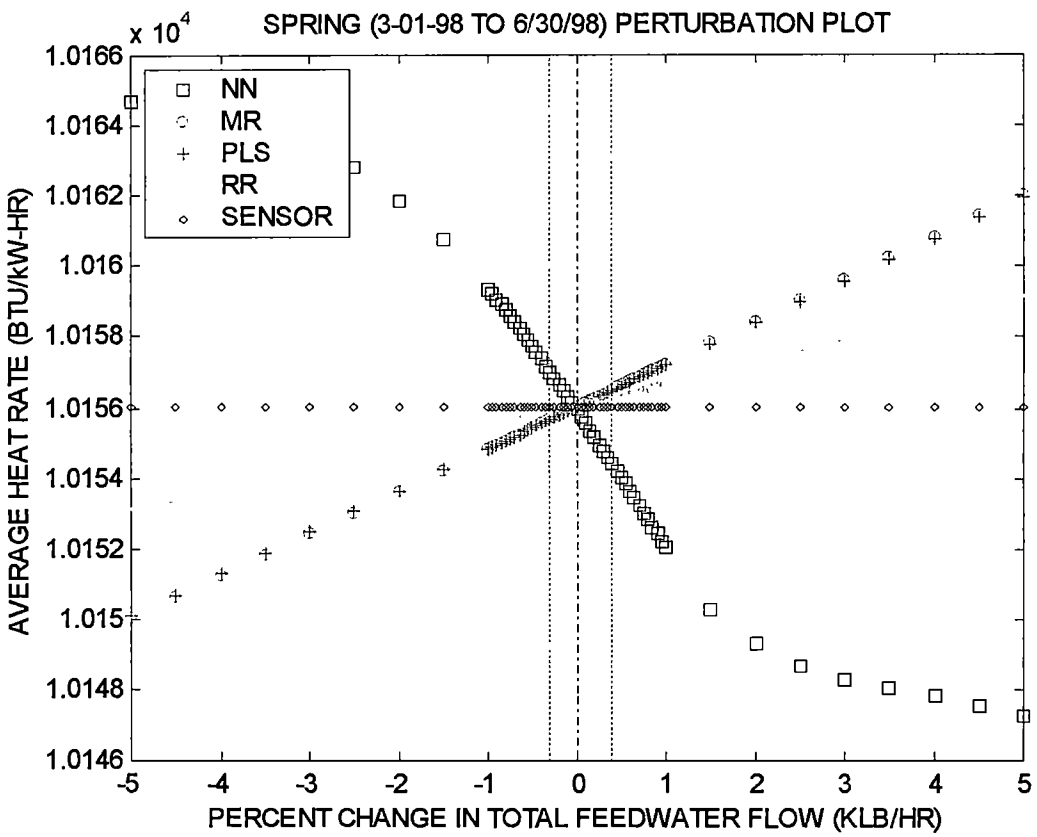

Figure A-115 Spring perturbation plot, Total Feedwater Flow, with Ridge Regression

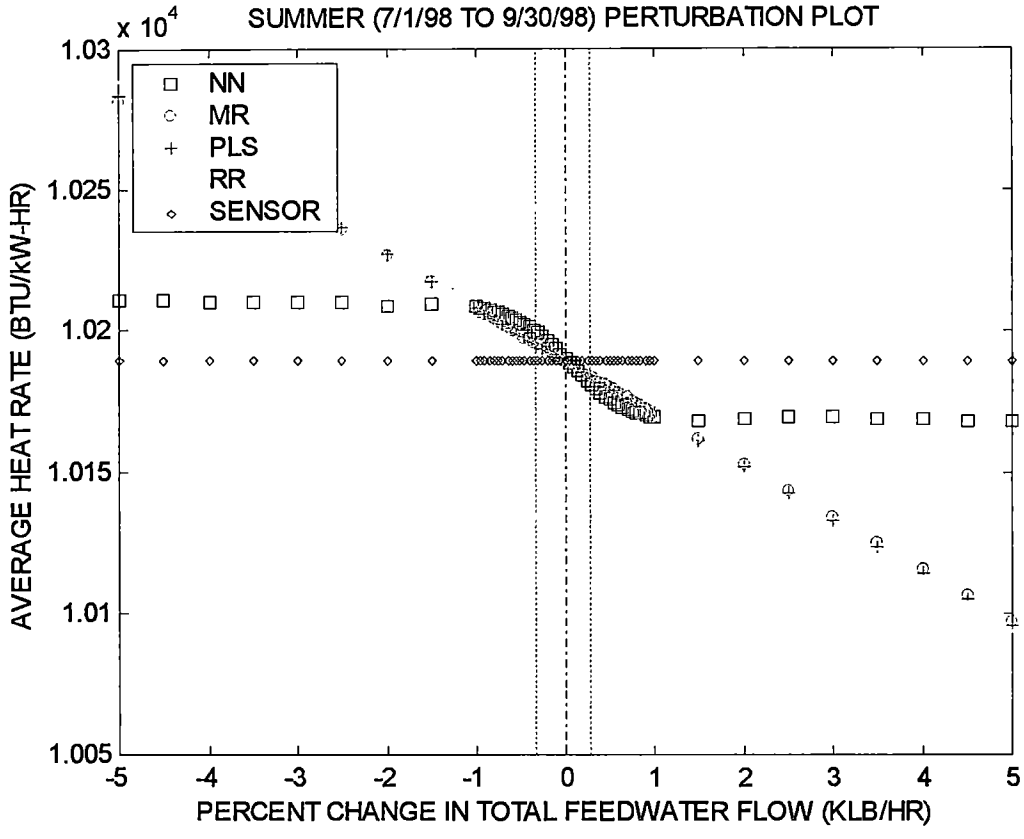

Figure A-116 Summer perturbation plot, Total Feedwater Flow, with Ridge Regression

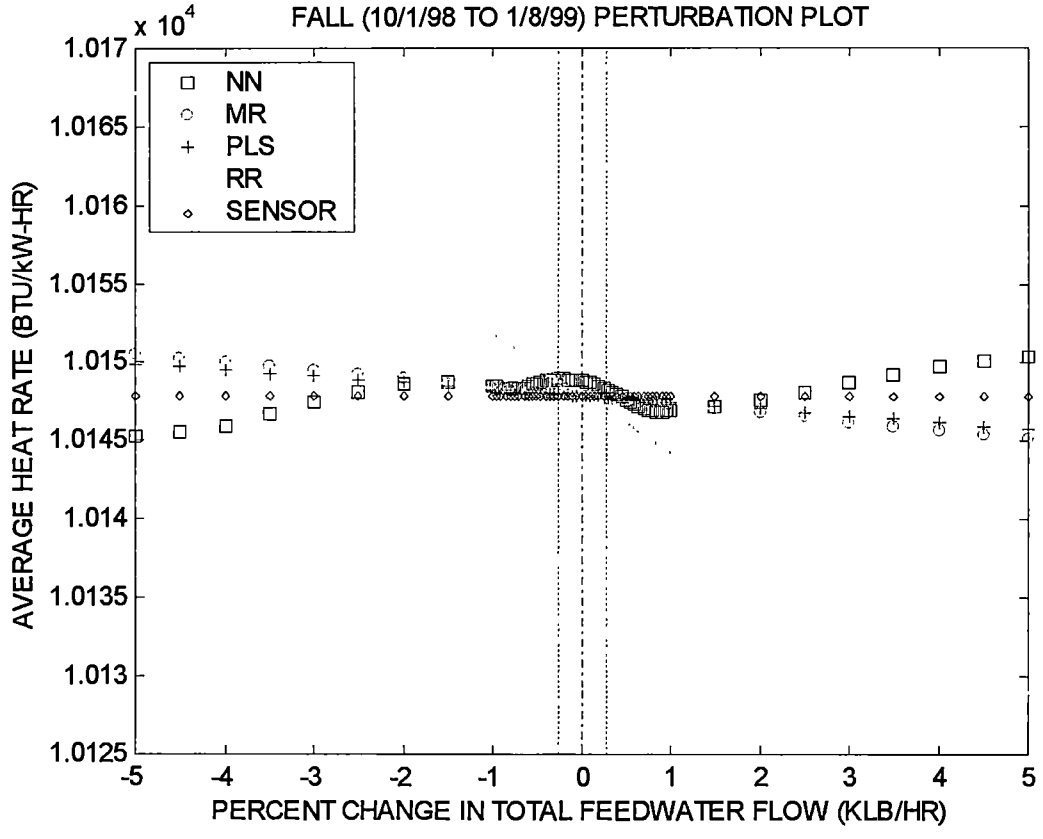

Figure A-117 Fall perturbation plot, Total Feedwater Flow, with Ridge Regression

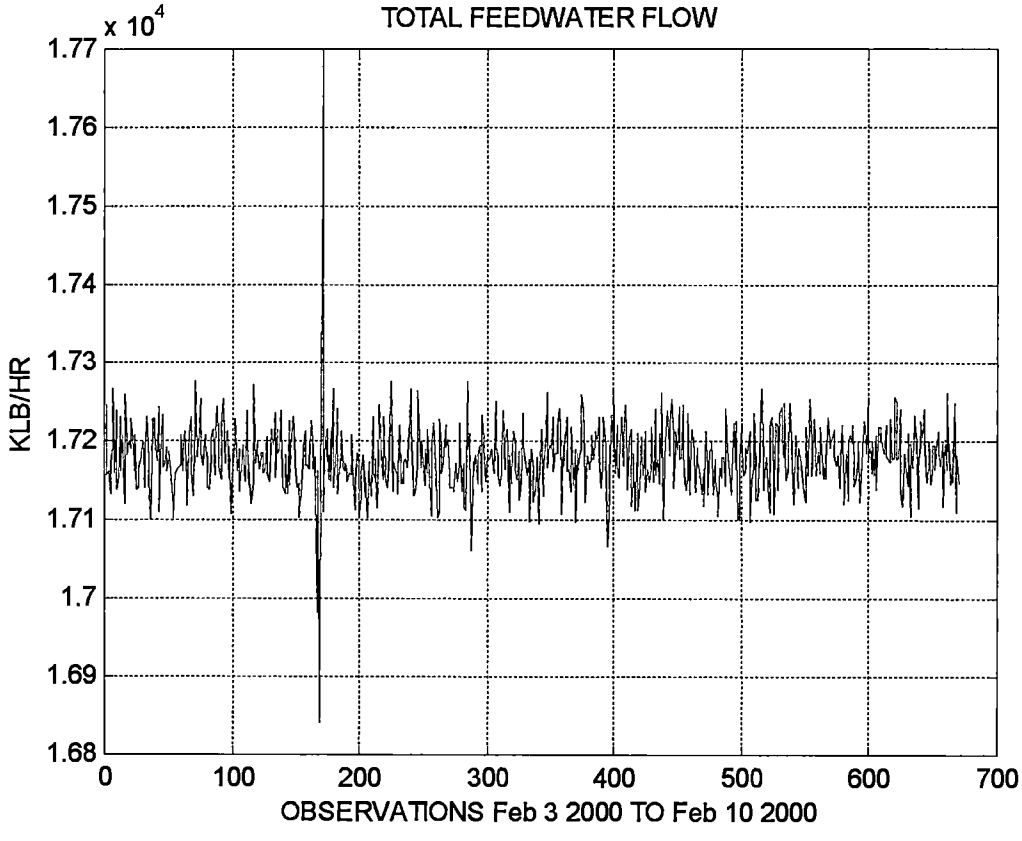

Figure A-118 Plot of Feedwater Flow 15 minute time separation between data points

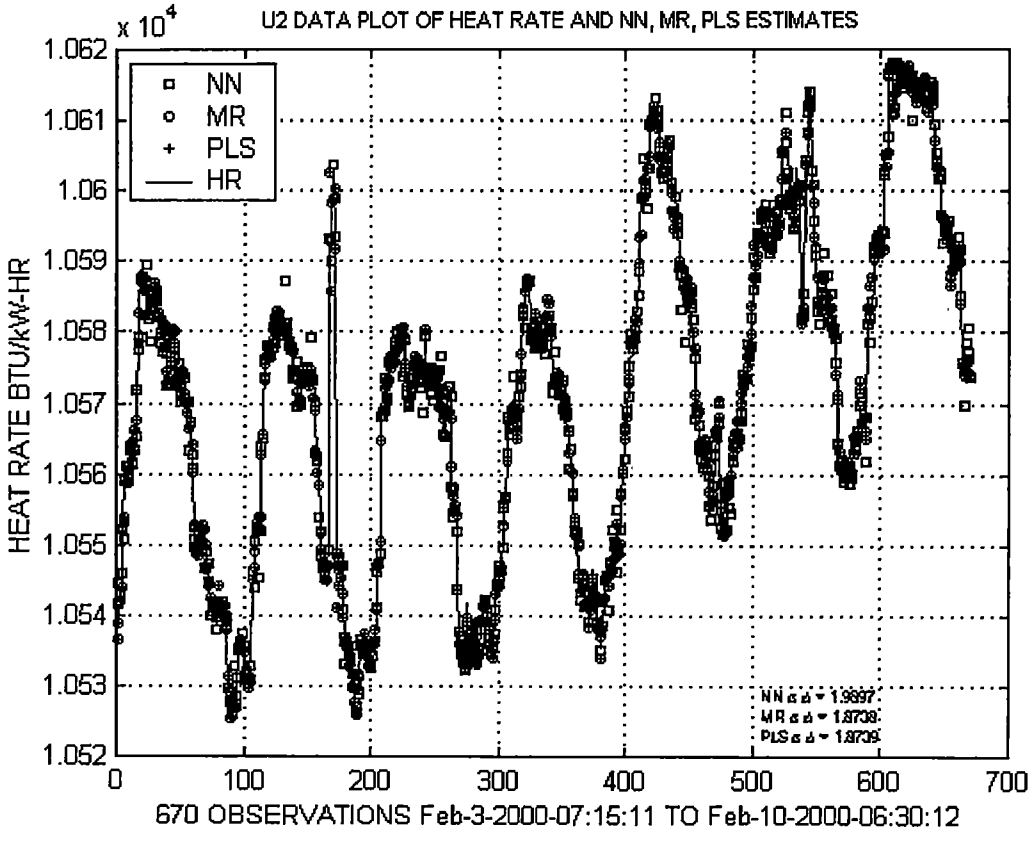

Figure A-119 Plot of NN, MR, PLS estimates, 15 minute time separation between data points

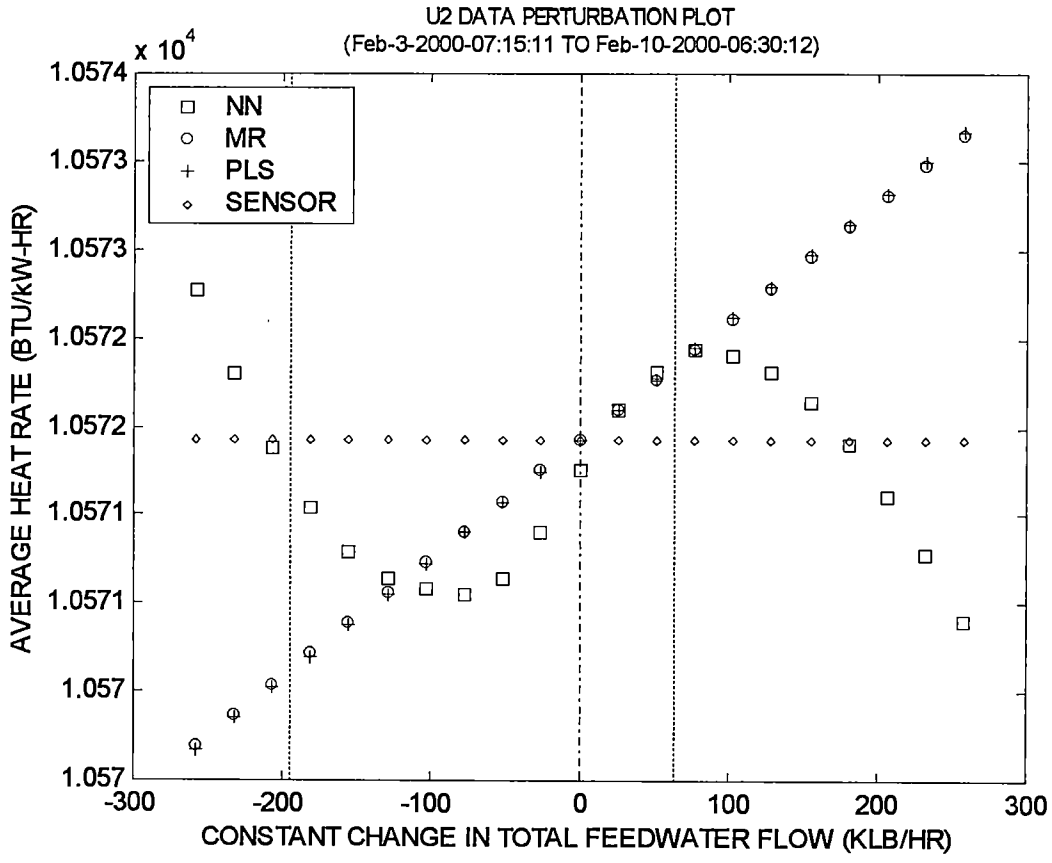

Figure A-120 Perturbation plot with U2 data, 15 minute time separation between data points, NN with LM training and 14 hidden nodes

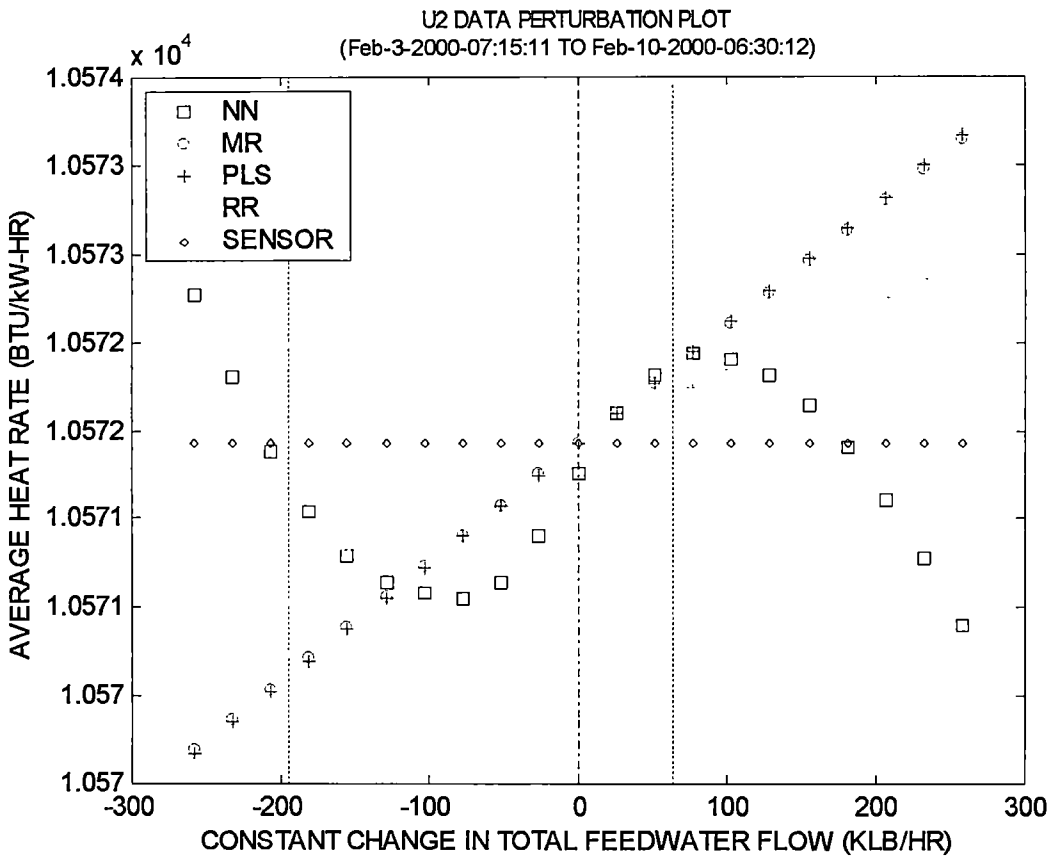

Figure A-121 Perturbation plot with U2 data, 15 minute time separation between data points, NN with LM training and 14 hidden nodes, Ridge Regression estimate

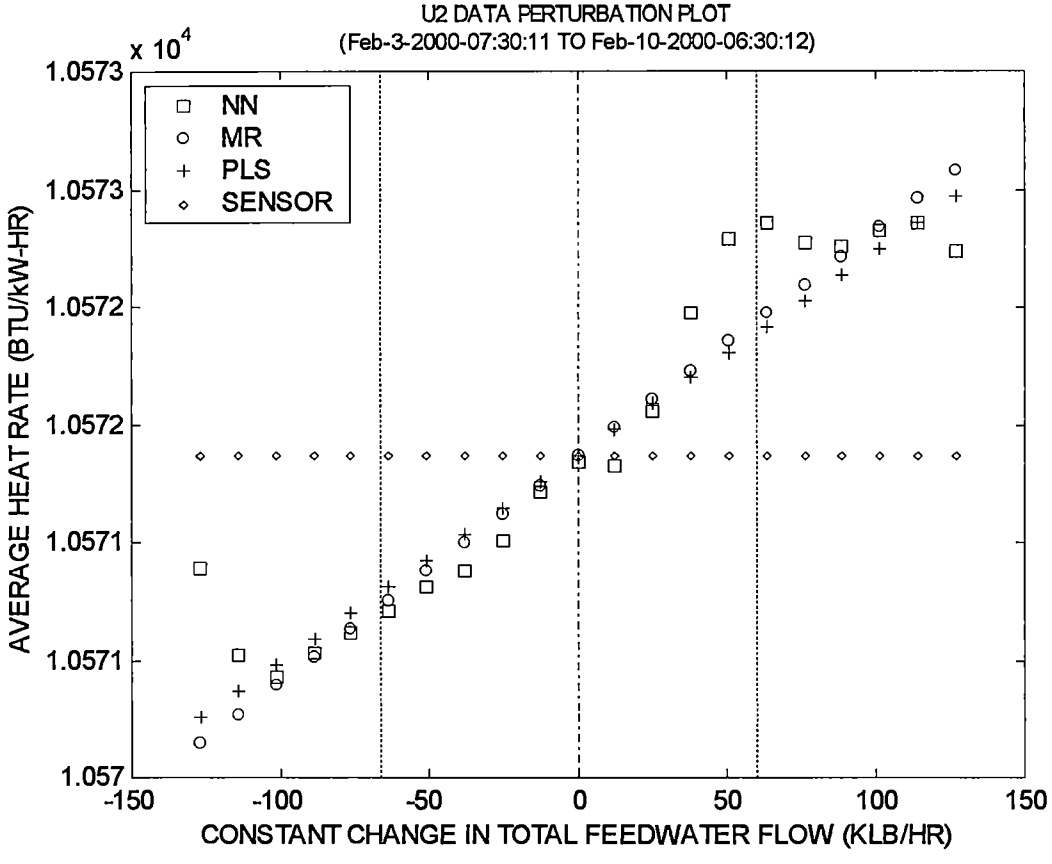

Figure A-122 Perturbation plot with U2 data, 30 minute time separation between data points, NN with LM training and 14 hidden nodes

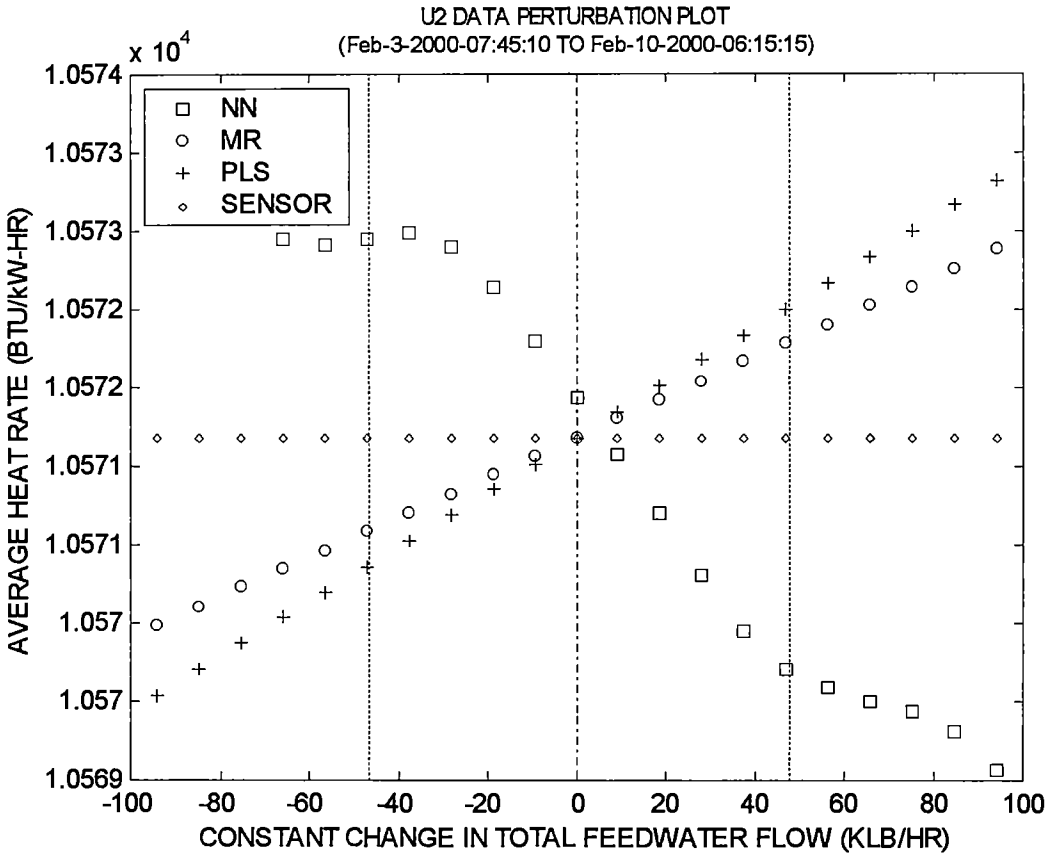

Figure A-123 Perturbation plot with U2 data, 45 minute time separation between data points, NN with LM training and 14 hidden nodes

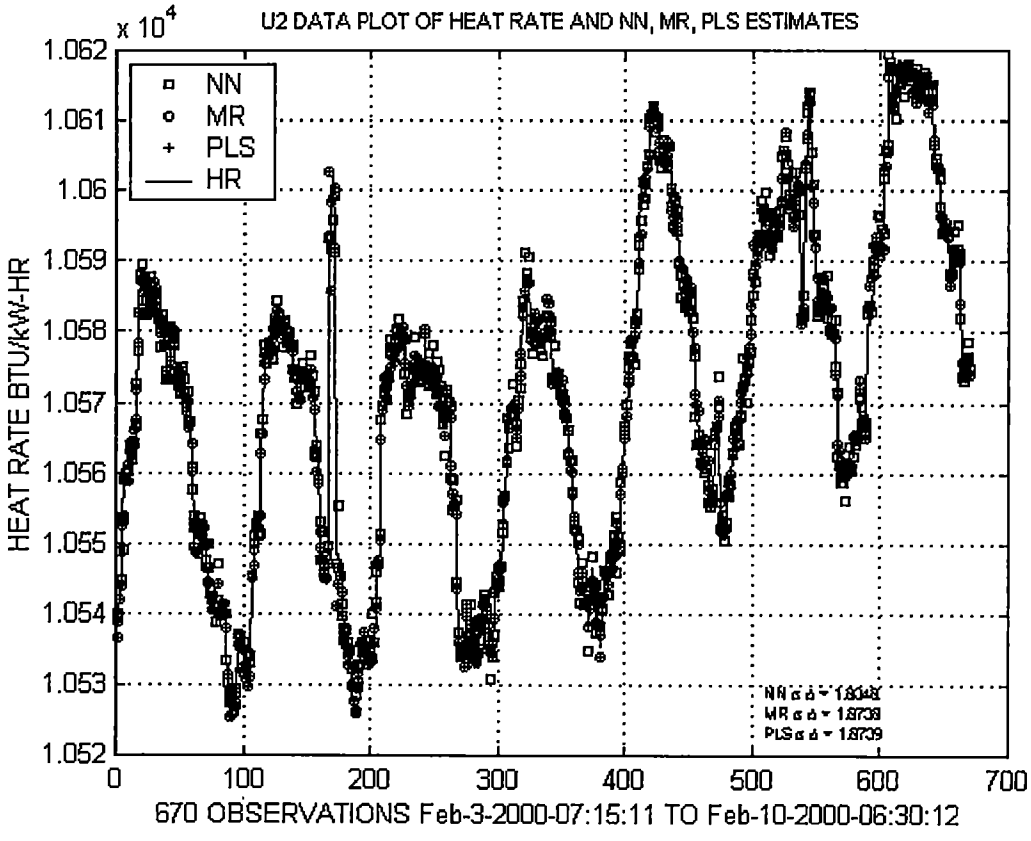

Figure A-124 Plot of NN, MR, PLS estimates, using an Elman neural network

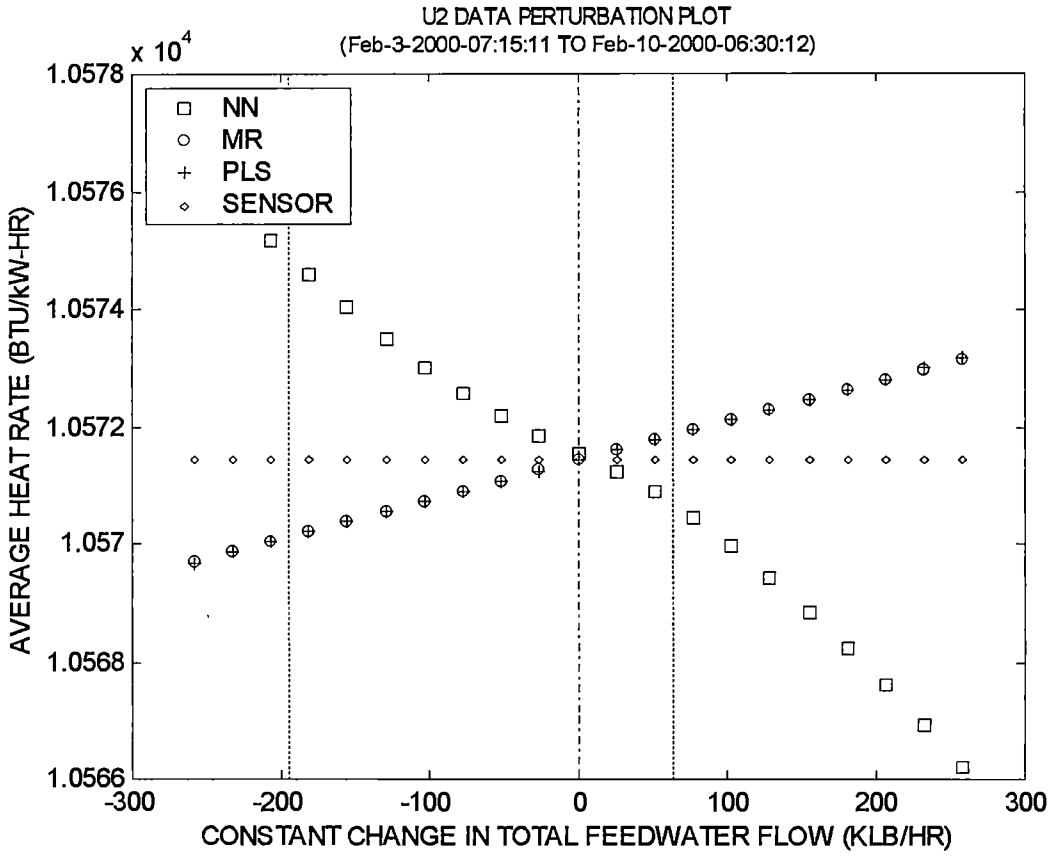

Figure A-125 Elman network, 1 delay, 15 minute time separation between data points, 18 nodes

Appendix B Listing of Computer Codes

File 1 Group_clean.m

```
% Group_clean
□
% variables with bad indicies identified in Group_info structure
□

□
%clear all
□
load ws
□
load Group
□
load Watts_Bar_variables_selected
□

□
% checks for exact match between Watts Bar and Palo Verde variables

temp = 0;
for i = 1:length(Group)

   [C,IA,IB] =
intersect(Group(i).selection_variable_description,Watts_Bar_variables_selected)
;
   temp = temp + length(IA);

end
disp('Watts Bar and Palo Verde selected variable descriptions')
disp(['Total number of exact matches is: ' num2str(temp)]);

% checks Group indices against ws.bad_indices
% removes questionable variables

l = 0;
Group_info = [];
Group_info.variables_bad_indices = [];
Group_info.variables_bi_group_number = [];
Group_info.variables_bi_index = [];
Group_info.total_number_variables_bi = [];


for i = 1:length(Group)
   k = 1;
   for j = 1:Group(i).number_of_variables

      temp1 = Group(i).ws_number(j);
      temp2 = Group(i).ws_index(j);

      [C1,IA1,IB1] = intersect(temp2, ws(temp1).bad_indices);
      [C2,IA2,IB2] = intersect(temp2, ws(temp1+1).bad_indices);

      if ~isempty(C1) | ~isempty(C2)
         Group_info.variables_bad_indices = [Group_info.variables_bad_indices
Group(i).ws_description(j)];
         Group_info.variables_bi_group_number =
[Group_info.variables_bi_group_number i];
         Group_info.variables_bi_index = [Group_info.variables_bi_index j];
```

```
         % Group(i).ws_description(j)
         k = k+1;
         l = l + 1;
      end

   end
end

% kill the variable and all its information from Group

for i = 1:length(Group)
   for j = 1:length(Group_info.variables_bad_indices)

      [C,IA,IB] = intersect(Group_info.variables_bad_indices(j),
Group(i).ws_description);
      if ~isempty(IB)
         % Group = delete_variable_from_Group(i,IB, Group);
      end

   end
   Group_info.total_number_variables_bi = 1;
end

% clean up

clear C C1 C2 IA IA1 IA2 IB IB1 IB2 Watts_Bar_variables_selected i j k l temp
temp1 temp2 ws
```

File 2 Group_state.m

```
% Group_state
□
% identifies variables as State or Control
□

□
%% state variables
□

□
Ginfo(1).variables_state = {
□
   'AVG CONDENSER PRESSURE       '
□
   'Avg. CNDSR(C)IN CW TEMP L A/B     '
   'Avg. CNDSR(C)IN CW TEMP L C/D     '
   'Avg. CNDSR(C) OUT CW TEMP A/B     '
   'Avg. CNDSR(C) OUT CW TEMP C/D     '
   'Avg. CNDSR(A)IN CW TEMP L A/B     '
   'Avg. CNDSR(A)IN CW TEMP L C/D     '
   'Avg. CNDSR(B)IN CW TEMP L A/B     '
   'Avg. CNDSR(B)IN CW TEMP L C/D     '
   'CNDS PMP A EFF                '
   'CNDS PMP B EFF                '
   'CNDS PMP C EFF                '
   'CONDENSER(A) PRESSURE        '
   'CONDENSER(B) PRESSURE        '
   'CONDENSER(C) PRESSURE        '
}';
```

```
Ginfo(2).variables_state = {
   'FW PMP A EFF                  '
   'FW PMP B EFF                  '
   %   'FW PMP A FW SP VOL            '
   'FW PMP B FW SP VOL            '
}';

Ginfo(3).variables_state = {
   'ELECTRICAL GENERATION'
   'THERMAL GENERATION'
   'HEAT RATE (GROSS)'
   'CORRECTED ELECTRICAL GENERATION'
   'HEAT RATE (CORRECTED)'
}';

Ginfo(4).variables_state = {
   'HTR 1A DRAIN TEMP             '
   'HTR 1B DRAIN TEMP             '
   'HTR 1C DRAIN TEMP             '
   'HTR 2A DRAIN TEMP             '
   'HTR 2B DRAIN TEMP             '
   'HTR 2C DRAIN TEMP             '
   'HTR 3A DRAIN TEMP             '
   'HTR 3B DRAIN TEMP             '
   'HTR 3C DRAIN TEMP             '
   'HTR 4A DRAIN TEMP             '
   'HTR 4B DRAIN TEMP             '
   'HTR 4C DRAIN TEMP             '
   'HTR 5A DRAIN TEMP'
   'HTR 5B DRAIN TEMP             '
   'HTR 6A DRAIN TEMP             '
   'HTR 6B DRAIN TEMP             '
   'HTR 7A DRAIN TEMP             '
   'HTR 7B DRAIN TEMP             '
   'HTR DRN PMP A EFF              '
   'HTR DRN PMP B EFF              '
   'HTR DRN PMP A EFF DESIGN       '
   'HTR DRN PMP B EFF DESIGN       '
   'HTR DRN PMP A EFF DEV          '
   'HTR DRN PMP B EFF DEV          '
}';

Ginfo(5).variables_state = {
}';

Ginfo(6).variables_state = {
}';

Ginfo(7).variables_state = {
}';

Ginfo(8).variables_state = {
   'TRB HEAT RATE GROSS DESIGN     '
}';

Ginfo(9).variables_state = {
}';

temp = 0;
Group_info.variables_state = [];
Group_info.variables_s_group_number = [];
Group_info.variables_s_index = [];
```

```
Group_info.total_number_variables_s = [];

for i = 1:length(Ginfo)
   temp = temp + length(Ginfo(i).variables_state);
   [C,IA,IB] = intersect(Group(i).ws_description,Ginfo(i).variables_state);
      Group_info.variables_state = [Group_info.variables_state
Ginfo(i).variables_state];
      Group_info.variables_s_group_number =
[Group_info.variables_s_group_number i.*ones(1,length(IA))];
      Group_info.variables_s_index = [Group_info.variables_s_index IA];
end

for i = 1:length(Group_info)
   Group_info.total_number_variables_s=temp;
end

Group_info.state_y.data = [];
Group_info.state_y.group_number = [];
Group_info.state_y.index = [];
Group_info.state_y.description = [];

[C,IAY,IB] = intersect(Group(3).ws_description,'HEAT RATE (CORRECTED)');
Group_info.state_y.data = Group(3).ws_data(:,IAY);
Group_info.state_y.group_number = 3;
Group_info.state_y.index = IAY;
Group_info.state_y.description = Group(3).ws_description(IAY);
clear C IAY IB

Group_info.state_x.data = [];
Group_info.state_x.group_number = [];
Group_info.state_x.index = [];
Group_info.state_x.description = [];
Group_info.state_x.removed = [];
Group_info.state_x.removed_description = [];

for i = 1:length(Group)
   if i ~= 3
      [C,IA,IB] = intersect(Group(i).ws_description,Ginfo(i).variables_state);
      Group_info.state_x.data = [Group_info.state_x.data
Group(i).ws_data(:,IA)];
      Group_info.state_x.group_number = [Group_info.state_x.group_number
i.*ones(1,length(IA))];
      Group_info.state_x.index = [Group_info.state_x.index IA];
            Group_info.state_x.description = [Group_info.state_x.description
Group(i).ws_description(IA)];
   end
   if i == 3
      [C,IA,IB] = intersect(Group(i).ws_description,Ginfo(i).variables_state);
      Group_info.state_x.removed = [Group_info.state_x.removed
Group(i).ws_data(:,IA)];
            Group_info.state_x.removed_description =
[Group_info.state_x.removed_description Group(i).ws_description(IA)];
   end
end

clear i temp Ginfo C IA IB
```

File 3 Minresv3.m

```
function [pattern,m,c] = minres(R,m,t,cc,flag_cluster,flag_randn)
%
%function [A,h,Fvar,e] = minres(R,m,t,cc,flag_cluster)
%


%
%[m,temp] = size(R);
%
%m = 6;
%t = 3000; cc = .001; flag_cluster = 1;

epscc = 10^-8;

%% minres method for calculating factor patterns and structure
%% 21 steps
%%
%% C. Irvine, UTK, January, 1999

%% R   -- correlation matrix, observed (with ones in the diagonal)
%% n   -- number of variables
%% N   -- number of observations or individuals
%% m   -- number of factors
%% t   -- maximum number of major iterations
%% cc  -- convergence criterion for pattern matrix
%%     -- convergence for logical tests (epscc)
%% i   -- minor iteration counter
%% j   -- variable iteration counter
%% c   -- major iteration counter

%% flag_cluster -- plots clustering of matrices
%% flag_randn   -- randimizes starting matrix

%% ----- variables -----

%% h      -- communalities for each variable
%% Fvar   -- variance for each factor

%% ----- STEP 1 -----
%% Input, store and/or print the variables

n = size(R,1);

if (size(R,2) ~= n ) disp(['Bad matrix']), break, end

disp(['Number of variables:    ' num2str(n)])
disp(['Number of factors:      ' num2str(m)])
disp(['Number of iterations:   ' num2str(t)])
disp(['Convergence criterion:  ' num2str(cc)])
disp(['Rank of observed correlation matrix: ' num2str(rank(R))]);

%% ----- STEP 2 -----
%% Calculate eigenvalues and eigenvectors, columns represent each eigenvector

if flag_randn == 1

   V  = randn(n);

else

       [V,D] = eig(R);
       [V,D] = sorteig(V,D);
```

```
end

%% ----- STEP 3 -----
%% Initial factor matrix Ao (n * m) consisting of first m principal components
%% calculate communalities for the variables
%% calculate variance of each factor

Ao = V(:,1:m);

% communality's (h) for each variable
% note: h stands for h{j}^2

ho = communality(Ao,n,m);

% variance (V) for each factor

Fvaro = communality(Ao',m,n);

%% ----- STEP 4 -----

%% START OF MAJOR LOOP

for c = 1:t

   %% START OF MINOR LOOP

   for j = 1:n

      % pattern matrix always uses the previous estimate

      if (c==1)&(j==1)
         A{j} = Ao;
      %elseif (j==1)
         %A{j} = Asave{c-1,n};
      %elseif (j~=1)
         %A{j} = Asave{c,j-1};
      else
         A{j} = Aold;
      end

      % calculate incremental changes e{j} (row vector, where e{j} = (e1, e2,
e3, ..., em) )

      e{j} = mines(A{j},R,j,m,n);

      %% ----- STEP 6 -----
      %% Calculate new b's, update A{i},
      %% New iteration

      for p = 1:m
         b{j}(p) = A{j}(j,p) + e{j}(p);
      end

      A{j}(j,:) = b{j};

      %% ----- STEP 7 -----
      %% check communality h{j}^2

      h{j} = communality(A{j}(j,:),1,m);

      if (h{j} > 1 + epscc)
```

```
%% ----- STEP 8 -----
%% Compute matrix W, K, and vp

[W,K,vp] = Wmatrix(A{j},R,m,n,j);

%% ----- STEP 9 -----
%% Diagonalize matrix W

[Q,D] = eig(W);
Gam = Q'*W*Q;

%% ----- STEP 10 -----
%% Additional transformations

% eigenvalues (eigenvalues are squared)
ev = diag(D)'; % row vector

% indices that have eigenvalues > 0
%       ind = find(ev>10^-7);
ind = find(ev>10^-5);

% indices that have eigenvalues <= 0
indz = find(ev<=10^-7);

if any(ev<0)
   disp(['Negative eigenvalues -- error']);
   temp = find(ev<0);
   ev(temp) = 0;
   %break;
end

vphat = vp*Q;

% zero vectors
xp         = zeros(1,length(ev)); % row vector
up         = zeros(1,length(ev)); % row vector
bhat= zeros(1,length(ev)); % row vector
sigma = zeros(1,length(ev)); % row vector

up(ind)    = vphat(ind)./ev(ind);
sigma(ind) = sqrt(ev(ind)).*up(ind);

%% ----- STEP 10.5 -----
%% Check to see if point belongs to region

temp = sum( (sigma(ind).^2)./ev(ind) );
if temp <= 1
   xp(ind) = sigma(ind);
else

   %% ----- STEP 11 -----
   %% Determine initial x1
   xp = solve_x1(sigma,ev,xp);

end

%% ----- STEP 14 -----
%% Determine modified matrix A
%% corrects for hj2>1

bhat(ind) = xp(ind) ./ sqrt(ev(ind));
```

```
         b{j} = (Q*bhat')';
         A{j}(j,:) = b{j};

      else
         %break; % Proceed to STEP 15
      end

      %% ----- STEP 15 -----
      %% Compute residual matrix

      %res{c,j} = (R - eye(n)) - (A{j}*A{j}' - diag(diag(A{j}*A{j}')));
      %res_values(c) = norm(res{c,j});

      res = (R - eye(n)) - (A{j}*A{j}' - diag(diag(A{j}*A{j}')));
      res_values(c) = norm(res);

      %% ----- STEP 16 -----
      %% Store results

      %hsave{c,j} = communality(A{j}(j,:),1,m);
      hsave = communality(A{j}(j,:),1,m);
      %bsave{c,j} = b{j};
      bsave = b{j};
      %Asave{c,j} = A{j};
          Aold = A{j};

      disp(['iteration: ', int2str(c),' ', int2str(j)])

      %% ----- STEP 17 -----
      %% Skip

   end % minor loop

   %% ----- STEP 18 -----
   %% Test for maximum change
   flag_escape = 1;
   if c > 1
      for k = 1:n
         for p = 1:m
            %if abs( Asave{c,n}(k,p)-Asave{c-1,n}(k,p) ) > cc, flag_escape = 0;
end
            if abs( Aold(k,p)-Asave(k,p) ) > cc, flag_escape = 0; end
         end
      end
      if flag_escape == 1,
         disp(['major convergence met'])
         break; end
   end

   % Asave should be Asave{c-1,n}
   if j == n
             Asave = Aold;
   end

end % major loop

disp(['number of iterations (total and major): ', int2str(c), ' ',
int2str(c/n)])
indicies = [c,j];
%residual = res{c,j}
residual = res;
%pattern = Asave{c,j}
```

```
%communalities = sum((Asave{c,j}.^2)')
pattern = Asave;
communalities = sum((Asave.^2)');

%% ----- STEP 19 -----
%% Test for preset maximum number of iterations
if c == t
   disp(['possible none convergence'])
   disp(['maximum number of iterations reached'])
   temp = min(res_values);
   templ = find(res_values==temp);
 %  disp(['best pattern matrix is in bA'])
 %  bA = Asave(templ,m);
end

%% ----- PLOTTING -----
%% clustering plots

if flag_cluster == 1

   CUTOFF = 3;

   Yo=pdist(Ao,'euclid'); Zo=linkage(Yo,'single'); To=cluster(Zo,CUTOFF);
dendrogram(Zo);
   set(gca,'Title',text('String',['Dendrogram, PCA pattern matrix.  Number of
components: ' int2str(m)],'Color','r'))
   set(gca,'Xlabel',text('String',['variables, CUTOFF: ' int2str(CUTOFF)']))
   set(gca,'Ylabel',text('String','linkage distance'))
   set(gcf,'Tag','f1')

   Yp=pdist(pattern,'euclid'); Zp=linkage(Yp,'single'); Tp=cluster(Zp,CUTOFF);
dendrogram(Zp);
   set(gca,'Title',text('String',['Dendrogram, Factor pattern matrix.  Number
of factors: ' int2str(m)],'Color','r'))
   set(gca,'Xlabel',text('String',['variables, CUTOFF: ' int2str(CUTOFF)']))
   set(gca,'Ylabel',text('String','linkage distance'))
   set(gcf,'Tag','f2')

   Yr=pdist(R,'euclid'); Zr=linkage(Yr,'single'); Tr=cluster(Zr,CUTOFF);
dendrogram(Zr);
   set(gca,'Title',text('String','Dendrogram, Correlation matrix','Color','r'))
   set(gca,'Xlabel',text('String',['variables, CUTOFF: ' int2str(CUTOFF)']))
   set(gca,'Ylabel',text('String','linkage distance'))
   set(gcf,'Tag','f3')

   Ypp=pdist(pattern*pattern','euclid'); Zpp=linkage(Ypp,'single');
Tpp=cluster(Zpp,CUTOFF); dendrogram(Zpp);
   set(gca,'Title',text('String',['Dendrogram, Estimated correlation matrix
A*A''  Number of factors: ' int2str(m)],'Color','r'))
   set(gca,'Xlabel',text('String',['variables, CUTOFF: ' int2str(CUTOFF)']))
   set(gca,'Ylabel',text('String','linkage distance'))
   set(gcf,'Tag','f4')

end

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

File 4 W.m

```
function W = W(A,m,n,j)

% determine pattern loadings for W
for p = 1:m
   temp = 0;
   for k = 1:n
      if (k <> j), temp = temp + A(k,p); end
   end
   D(p) = temp
end

% fill W
for i = 1:m
   for ii = 1:m
      W(i,ii) = D(i)*D(ii);
   end
end
% ---------------------------------------------------------------
% ---------------------------------------------------------------

% ---------------------------------------------------------------
% ---------------------------------------------------------------
function h = communality(A,n,m)
% A matrix of patterns
% n maximum outer iteration
% n maximum inner iteration

for j = 1:n
   temp = 0.0;
   for p = 1:m
      temp = temp + A(j,p)^2;
   end
   h(j) = temp;
end
% ---------------------------------------------------------------
% ---------------------------------------------------------------
```

File 5 Wmatrix.m

```
function [W,K,vp] = Wmatrix(A,R,m,n,j)

% determine pattern loadings for W
```

```
A(j,:) = 0;

W = A(1:n,1:m)'*A(1:n,1:m);

r = R(j,:);
r(j) = 0;

K = sum(r.^2); % constant

vp = r*A(1:n,1:m); % row vector

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

File 6 wsfind.m

```
function
[number,index,data,description,units,dates,tag,two_sigma,average]=wsfind(variab
le_name,ws)


% finds relevant data in worksheet (ws) for a variable name


if ismember(variable_name,ws(1).description)

   j = 1;

elseif ismember(variable_name,ws(3).description)

   j = 3;

else

   return;

end

[C,IA,IB] = intersect(variable_name,ws(j).description);

number = j;
index = IB;
data = [ws(j).data(:,IB); ws(j+1).data(:,IB)];
description = C;
units = ws(j).units(IB);
dates = [ws(j).dates(:); ws(j+1).dates(:)];
tag = ws(j).tag(IB);
two_sigma = ws(j).two_sigma(IB);
average = ws(j).average(IB);
```

File 7 SetPointPLS.m

```
%% load data

clear SP
load Group
Group_clean
```

```
Group_state
Group_setpoint
Group_fix

% set seed generator

rand('state',0)
randn('state',0)

% pre-process inputs
%        1 2 3 4 5 6 7 8 91011121314

flags = [1 1 0 0 1 0 0 0 0 0 1 1 1 1 1];
sp = preprocessv2(Group_info.sp_x.data,flags,1,[]);
flags = [1 0 0 0 1 0 0 0 0 0 1 1 1 1 0];
st = preprocessv2(Group_info.state_x.data,flags,[],[]);
flags = [1 0 0 0 1 0 0 0 0 0 1 1 1 1 0];
hr   = preprocessv2(Group_info.state_y.data,flags,[],[]);

clear flags

%%%%%% STATE MAPPING (MAP2) %%%%%%

model2 = 'mlr';
%model2 = 'pls';
stages = 100;
num_of_scores = 5;
%num_of_scores = iii;
type_scores = 'pca';
[temp1, temp2] = inputs(st.XWSD,type_scores,num_of_scores);
%MAP2 = fmap(inputs(st.XWSD,'factors',5),hr.XWSD,'mlr');
% factors pca same highcorr prinvar
MAP2 = fmap(temp1,hr.XWSD,model2,stages);
MAP2.model = model2;
MAP2.stages = stages;
MAP2.type_scores = type_scores;
MAP2.TM = temp2;
MAP2.num_of_scores = num_of_scores;
clear model2 stages type_scores num_of_scores temp1 temp2

%post scale data
MAP2.Yps = prepost(MAP2.results.Y,hr.AVG,hr.STDDEV,1,1);   % post scales data
MAP2.Tps = prepost(MAP2.T,hr.AVG,hr.STDDEV,1,1);   % post scales data

% mean-squared error
MAP2.mse = (mean((MAP2.Yps-MAP2.Tps).^2));

% graphs
switch lower(MAP2.flag)
case 'mr'
   mr_graphs(Group,[1 [] []],MAP2.Tps,MAP2.Yps,MAP2.mse,MAP2.results)
end

%%%%%% END STATE MAPPING %%%%%%

%%%%%% SETPOINT MAPPING %%%%%%

% Setpoints as predictors and MAP2.P as target

model1 = 'mmr';
%model1 = 'pls';
stages = 55;
```

```
num_of_scores = 55;
type_scores = 'same';
%MAP1 = fmap(sp.XWSD,MAP2.P,model1,stages);
[temp1, temp2] = inputs(sp.XWSD,type_scores,num_of_scores);
MAP1 = fmap(temp1 ,MAP2.P,model1,stages);
MAP1.model = model1;
MAP1.stages = stages;
MAP1.type_scores = type_scores;
MAP1.TM = temp2;
MAP1.num_of_scores = num_of_scores;
clear model1 stages type_scores num_of_scores temp1 temp2

%post scale data

% mean-squared error
MAP1.rmse = sqrt(mean((MAP1.results.Y-MAP2.P).^2));
MAP1.rmspet = sqrt(mean((100*(MAP1.results.Y-MAP2.P)./MAP2.P).^2));
%graph('percentage error',MAP1.results.Y,MAP2.P,1)

%%%%%% END SETPOINT MAPPING %%%%%%

%%%%%% PERTURBATION %%%%%%

signal = Group_info.sp_x.description(sp.Igood2);

%[m,n] = size(sp.XWSD);
%temp = sp.XWSD;
[m,n] = size(MAP1.P);
temp = MAP1.P;

%%%%%% CHECK MODEL STATE WITH NO CHANGES %%%%%%

switch lower(MAP1.model)
case 'nn'
   temp1 = sim(MAP1.results.tnet,temp')';
case 'mmr'
   temp1 = temp*MAP1.results.beta;
case 'pls'
   temp1 =
pls_reverse(temp,MAP1.results.actual_stages,MAP1.results.T,MAP1.results.W,MAP1.
results.P,MAP1.results.Q);
end
switch lower(MAP2.model)
case 'nn'
   temp2 = sim(MAP2.results.tnet,temp1')';
case 'mlr'
   temp2 = [ones(m,1) temp1]*MAP2.results.beta;
case 'pls'
   temp2 =
pls_reverse(temp1,MAP2.results.actual_stages,MAP2.results.T,MAP2.results.W,MAP2
.results.P,MAP2.results.Q);
end

%post scale data
MS.hr = prepost(temp2,hr.AVG,hr.STDDEV,1,1);   % post scales data

%%%%%% CHECK MODEL STATE WITH CHANGES %%%%%%

pert = 'ramp';
begin = 100;
stop  = 110;
for i = 1:55
```

```
    value = sp.XW(begin,i);
    vinc(i) = .0005*value;
    %   vinc(i) = .01*value;
    temp = perturb(sp.XW(:,i),vinc(i),pert,begin,stop);
    pertsig{i} = temp;
    cvec{i} = prepost(temp,sp.AVG(i),sp.STDDEV(i),0,1);
    pmat = sp.XWSD;
    pmat(:,i) = cvec{i};
    switch lower(MAP1.type_scores)
    case 'pca'
       pmat = pmat*MAP1.TM;
       pmat = pmat(:,1:MAP1.num_of_scores);
    end
    %pmat(:,i) = perturb(sp.XWSD(:,i),.1,pert,begin,stop); % SD pert

    switch lower(MAP1.model)
    case 'nn'
       temp1 = sim(MAP1.results.tnet,pmat')';
    case 'mmr'
       temp1 = pmat*MAP1.results.beta;
    case 'pls'
       temp1 =
pls_reverse(pmat,MAP1.results.actual_stages,MAP1.results.T,MAP1.results.W,MAP1.
results.P,MAP1.results.Q);
    end
    switch lower(MAP2.model)
    case 'nn'
       temp2 = sim(MAP2.results.tnet,temp1')';
    case 'mlr'
       temp2 = [ones(m,1) temp1]*MAP2.results.beta;
    case 'pls'
       temp2 =
pls_reverse(temp1,MAP2.results.actual_stages,MAP2.results.T,MAP2.results.W,MAP2
.results.P,MAP2.results.Q);
    end

    %post scale data
    MS.phr(:,i) = prepost(temp2,hr.AVG,hr.STDDEV,1,1);    % post scales data
    sensitivity(i) = mean(100*[MS.phr(stop:m,i)-
MS.hr(stop:m)]./MS.hr(stop:m))/vinc(i);
    PET{i} = 100*[MS.phr(:,i)- MS.hr]./MS.hr;
    MPET(i) = mean(PET{i}(stop:m));
end

clear m n temp i pmat value temp1 temp2
%%%%%% END PERTURBATION %%%%%%

%%%%%% SAVE RESULTS %%%%%%
i = 10;
load PLS
SP(i).description = {'5% RAMP INCREASE','43 STATE SD SCORES','PLS-PLS'};
SP(i).MAP1 = MAP1;
SP(i).MAP2 = MAP2;
SP(i).MS = MS;
SP(i).sp = sp;
SP(i).hr = hr;
SP(i).st = st;
SP(i).signal = signal;
SP(i).Group_info = Group_info;
SP(i).pert = pert;
SP(i).vinc = vinc;
SP(i).cvec = cvec;
```

```
SP(i).pertsig = pertsig;
SP(i).sensitivity = sensitivity;
SP(i).PET = PET;
SP(i).MPET = MPET;
save PLS SP
clear i MAP1 MAP2 MS sp hr st signal Group_info Group pert vinc cvec
sensitivity stop begin PET MPET

%%%%%% END %%%%%%
```

File 8 trainNN.m

```
function [temp1,temp2,time,total_epochs,num_nodes] = trainNN(x,y)


temp = 1e10; temp1 = []; temp2 = [];

tic

total_epochs = 0;

[P,T,VV] = PTVV(x,y);
[m,n] = size(x);
num_nodes = round(n/4);
num_nodes = 4;

for i = 1:5;
   net = nnet(x,y,num_nodes,'tansig','purelin');
%   net = nnet(x,y,num_nodes,'tansig','purelin');
%   net = nnet(x,y,10,'purelin','purelin');

 %  [tnet, trd] = train(net,P,T,[],[],VV,[]);
     [tnet, trd] = train(net,P,T);
   if min(trd.perf) < temp
      temp1 = tnet; temp2 = trd;% save junk
      temp = min(trd.perf);
   end
   total_epochs = total_epochs + length(trd.epoch);

end

old_tot = total_epochs;
time = toc;
total_epochs = old_tot;
```

File 9 sorteig.m

```
function [vs,ds] = sorteig(v,d);

% v matrix of eigenvectors

% d matrix of eigenvalues

% vs matrix of sorted eigenvectors in descending order

% ds matrix of sorted eigenvalues in descending order
```

```
[m,n] = size(v);
vs = [];
ds = [];
temp = diag(d);

for i = 1:m
   [IB] = find(temp==max(temp));
   vs = [vs v(:,IB)];
   ds = [ds temp(IB)];
   temp(IB) = -1e20;
end
ds = diag(ds);
```

File 10 PTVV.m

```
function [P,T,VV] = PTVV(x,y)


[m,n]  = size(x);
[m1,n1] = size(y);


% x - input
% y - output
% P - input vectors
% T - target vectors
% VV - validation vectors


per      = 3;


ind1 = find(mod([1:m1],per));
ind2 = find(mod([1:m1],per)==0);
P    = x(ind1,:)';                    % input vectors
VV.P = x(ind2,:)';                    % validation vectors
T    = y(ind1,:)';                    % layer target vectors
VV.T = y(ind2,:)';                    % validation vector

flag_1 = 0;

%c = randn(400,n1);
%d = randn(400,n);
%IA = find(c >= .5);
%IB = find(c < .5);

%noise_y = [.001*c(IA,:) -.01*c(IB,:)];
%noise_y = .05*c;
%noise_x = .05*d;
```

```
%P    = x(1:400,:)';                    % input vectors
%VV.P = [x(1:400,:)+noise_x]';                  % validation vectors
%VV.P = x(1:400,:)';                    % validation vectors
%VV.T    = y(1:400,:)';                    % layer target vectors
%VV.T = [y(1:400,:)+noise_y]';          % validation vector
%T = y(1:400,:)';          % validation vector

if flag_1 == 1
   begin = 215;
   finish = 290;

   ind3a = find(ind2<=begin);
   ind3b = find(ind2>=finish);
   ind4a = find(ind1<=begin);
   ind4b = find(ind1>=finish);

   b = begin+1:finish-1;
   a = begin+1:4:finish-1;
   c = setdiff(b,a);

%   ind5 = [ind1(ind4a) a ind1(ind4b)];
   ind5 = [ind1(ind4a) b ind1(ind4b)];
%   ind5 = [ind1(ind4a) c ind1(ind4b)];
%   ind6 = [ind2(ind3a)  ind2(ind3b)];
%   ind6 = [ind2(ind3a) a ind2(ind3b)];
   ind6 = [ind2(ind3a) b ind2(ind3b)];
%   ind6 = [ind2(ind3a)  ind2(ind3b)];
   P = x(ind5,:)';
   T = y(ind5,:)';
   VV.P = x(ind6,:)';
   VV.T = y(ind6,:)';
end
```

File 11 preprocessv2.m

```
function pp = preprocessv2(x,flags,Svar,Lvar)
%% preprocess.m
%%
%% preprocess data, prior to training for neural network
%% variables in x (rows are observations, columns are variables)
%% modified October 99, 1999 -- set all variables into 1 structure called pp
%% 15 flags allowed:
%% flags equal to 1 are set on
%%
%% flag_filter  -- filter data using MATLAB medfilt1 with N = 5
%% flag_killvs  -- kill signals with small variance ( < Svar, default is 1e-12
sigma)
%% flag_killvl       -- kill signals with large variance ( > Lvar, default is
50 sigma)
%% flag_mscale  -- mean scale data only
%% flag_sscale  -- standardize data, (standard deviates)
%% flag_rscale  -- scale data to a range
%% flag_ymscale -- mean scales criterion data  (not used)
%% flag_ysscale -- standardize criterion data  (not used)
%% flag_yrscale -- range scales criterion data (not used)
%% flag_biased  -- biased estimate uses multiplication by 1/n, the unbiased
estimate uses 1/(n-1)
%% flag_ssdev   -- calculates both standard deviates and small standard
deviates
```

```
%% flag_corr     -- determince correlation matrix coefficients, note: flag_ssdev
must be set
%% flag_cov      -- determince covariance matrix
%% flag_kdisp    -- displays list of killed signals
%% flag_kcoll    -- kills collinear signals, picks only one
%% Svar          -- small variance cutoff (Svar less than var(XWD))
%% Lvar          -- large variance cutoff (Lvar greater than var(XWD))

%clear all

%% ----- load data and flags -----
%% ----- data must be column vectors!, not MATLAB row vectors -----

%          1 2 3 4 5 6 7 8 9101112131415
% flags = [0 1 0 0 1 0 0 0 0 1 1 1 1 1 1];

flag_filter         = flags(1);
flag_killvs     = flags(2);
flag_killvl      = flags(3);
flag_mscale         = flags(4);
flag_sscale     = flags(5);
flag_rscale     = flags(6);
flag_ymscale  = flags(7);
flag_ysscale  = flags(8);
flag_yrscale  = flags(9);
flag_biased     = flags(10);
flag_ssdev      = flags(11);
flag_corr           = flags(12);
flag_cov        = flags(13);
flag_kdisp      = flags(14);
flag_kcoll      = flags(15);

%% ----- variables -----

% x                         % input variables in x
% XW                        % duplicate copies of x, also working set in this
program

Nfil = 5;             % used in medfilt1 to determine order N
if isempty(Svar)
   Svar = 1e-12;      % used to determine killing of small variance data
(default)
end
if isempty(Lvar)
   Lvar = 100;              % used to determine killing of large variance data
(default)
end
Is = [];                    % indices of small variance data
Il = [];                    % indices of large variance data
Ikilled = []; % indices of killed variables
Igood = [];   % indices of variables kept
Igood2 = [];    % indices of variables kept
Ikcoll = [];    % indices of killed collinear variables
kcoll = [];     % matching of collinear variables
Svarkill = []; % small variance of killed variables
Lvarkill = []; % large variance of killed variables
Rmin = 0;       % scale data to minimum value
Rmax = 1;       % scale data to maximum value
Re = [];        % vector of maximum values for XW
Rf = [];        % vector of minimum values for XW
BE = [];        % biased estimator variable, 0 for unbiased (1/n-1), 1 for
biased (1/n)
```

```
XW = [];        % matrix of raw data, working set
XWD = [];       % matrix of deviates
XWSD = [];      % matrix of standard deviates
XWSSD = [];     % matrix of small standard deviates
XWCORR = [];    % matrix of correlation coefficients
XWCOV = [];     % matrix of covariance coefficients
XWS = [];       % matrix of scaled data
m = [];         % total number of indivduals or observations
n = [];         % number of variables
AVG = [];       % mean of variables
STDDEV = [];    % standard deviation of variables

%% ----- create working matrices  -----

XW = x;

%% ----- set biased estimator variable -----

if (flag_biased == 1), BE = 1; else, BE = 0; end

%% ----- filter data -----

if (flag_filter == 1)

   XW  = medfiltl(XW,Nfil);

end

%% ----- determine sizes, means, and standard deviations  -----

[m,n]    = size(XW);
AVG      = mean(XW);
STDDEV   = std(XW,BE);
XWD = prepost(XW,AVG,STDDEV,0,0);     % deviates

%% ----- find indicies (Is) of signals with small variance components -----

if (flag_killvs == 1)
   if flag_biased == 1
      temp = var(XWD,BE); Is = find(abs(temp)<Svar); Svarkill = temp(Is);
   elseif flag_biased ==0
      temp = var(XWD); Is = find(abs(temp)<Svar); Svarkill = temp(Is);
   end
end

%% ----- find indicies (Il) of signals with large variance -----

if (flag_killvl == 1)
   if flag_biased == 1
      temp = var(XWD,BE); Il = find(abs(temp)>Lvar); Lvarkill = temp(Il);
   elseif flag_biased ==0
      temp = var(XWD); Il = find(abs(temp)>Lvar); Lvarkill = temp(Il);
   end
end

%% ----- kill variables if necessary -----

[m,n]    = size(x);
Igood    = 1:n;

if (flag_killvs == 1) | (flag_killvl == 1)
```

```
   temp = [Is Il];
   Ikilled = sort(temp);
   [C,IA,IB] = intersect(Igood,Ikilled);
   Igood(IA) = [];
   XW(:,[Ikilled]) = [];

end

%% ----- kill collinear signals -----

Igood2 = Igood;
if (flag_kcoll == 1)

   temp = [];
   [m,n]   = size(XW);
   for i = 1:(n-1)
      for j = (i+1):n
         R = corrcoef([XW(:,i) XW(:,j)]);
         if abs(R(1,2)) > .95, temp = [temp;[i j]]; end
      end
   end
   kcoll = temp;
   [C,IA,IB] = intersect([1:n],kcoll(:,2));
   Ikcoll = Igood(IA);
   Igood2(IA) = [];
   XW(:,IA) = [];
end

%% ----- START PRE-SCALING OF DATA -----

%% ----- determine sizes, means, and standard deviations  -----

[m,n]   = size(XW);
AVG     = mean(XW);
STDDEV  = std(XW,BE);

%% ----- calculate deviates, standard deviates and small standard deviates ----
-

if (flag_ssdev == 1)

       XWD = prepost(XW,AVG,STDDEV,0,0);     % deviates
   XWSD = prepost(XW,AVG,STDDEV,0,1);   % standard deviates, var([XW]) should
equal one

   if (flag_biased == 1),

      XWSSD = XWSD/sqrt(m);

   else

      XWSSD = XWSD/sqrt(m-1);

   end

   % Small standard deviates give the correct correlation matrix!!!
XWSSD'*XWSSD

end

%% ----- determine correlation coefficents -----
```

```
if ((flag_ssdev == 1) & (flag_corr == 1))

   XWCORR = XWSSD'*XWSSD;

end

%% ----- determine covariance matrix -----

if (flag_cov == 1)

   if (flag_biased == 1),

      XWCOV = XWD'*XWD/m;

   else

      XWCOV = XWD'*XWD/(m-1);

   end

end

%% ----- mean, standardize, or range scale data -----

if (flag_mscale == 1)

   XWS = prepost(XW,AVG,STDDEV,0,0); % mean-scaled data only

elseif (flag_sscale == 1)

   XWS = prepost(XW,AVG,STDDEV,0,1); % standard deviates, var([XW]) should
equal one

elseif (flag_rscale == 1)

   Re = max(XW);
   Rf = min(XW);
   XWS = prepost(XW,Rmin,Rmax,0,2,Re,Rf); % range should be between Rmin and
Rmax

end

%% ----- END SCALING PROCESS -----

%% ----- STRUCTURE pp -----

pp.x = x;                                % variables in x
pp.Nfil = Nfil;                  % used in medfilt1 to determine order N
pp.Svar = Svar;                  % used to determine killing of small variance
data
pp.Lvar = Lvar;                  % used to determine killing of large variance
data
pp.Is = Is;                              % indices of small variance data
pp.Il = Il;                              % indices of large variance data
pp.Ikilled = Ikilled;      % indices of killed variables
pp.Igood = Igood;          % indices of variables kept
pp.kcoll = kcoll;        % pairs of collinear variables
pp.Ikcoll = Ikcoll;      % indices of killed collinear variables
pp.Igood2 = Igood2;      % indices of variables kept
pp.Svarkill = Svarkill; % small variance of killed variables
pp.Lvarkill = Lvarkill; % large variance of killed variables
pp.Rmin = Rmin;            % scale data to minimum value
```

```
pp.Rmax = Rmax;            % scale data to maximum value
pp.Re = Re;                % vector of maximum values for XW
pp.Rf = Rf;                % vector of minimum values for XW
pp.BE = BE;                % biased estimator variable, 0 for unbiased (1/n-1),
1 for biased (1/n)
pp.XW = XW;                % matrix of raw data, working set
pp.XWD = XWD;              % matrix of deviates
pp.XWSD = XWSD;                  % matrix of standard deviates
pp.XWSSD = XWSSD;          % matrix of small standard deviates
pp.XWCORR = XWCORR;        % matrix of correlation coefficients
pp.XWCOV = XWCOV;          % matrix of covariance coefficients
pp.XWS = XWS;              % matrix of scaled data
pp.m = m;                        % total number of indivduals or observations
pp.n = n;                        % number of variables
pp.AVG = AVG;              % mean of variables
pp.STDDEV = STDDEV;        % standard deviation of variables
pp.flags = flags;
pp.flag_filter             = flag_filter;
pp.flag_killvs     = flag_killvs;
pp.flag_killvl         = flag_killvl;
pp.flag_mscale             = flag_mscale;
pp.flag_sscale     = flag_sscale;
pp.flag_rscale     = flag_rscale;
pp.flag_ymscale      = flag_ymscale;
pp.flag_ysscale      = flag_ysscale;
pp.flag_yrscale      = flag_yrscale;
pp.flag_biased     = flag_biased;
pp.flag_ssdev      = flag_ssdev;
pp.flag_corr         = flag_corr;
pp.flag_cov        = flag_cov;
pp.flag_kdisp      = flag_kdisp;

save pp pp

%% ----- END STRUCTURE pp -----

%% ----- clean up -----

clear x y Nfil Svar Lvar Is Il Ikilled Svarkill Lvarkill Rmin Rmax Re Rf
clear Rey Rfy BE XW YW XWD YWD XWSD YWSD XWSSD YWSSD XWCORR YWCORR XWCOV YWCOV
clear XWS YWS m my n ny AVG AVGY STDDEV STDDEVY
clear flags flag_filter flag_killvs flag_killvl flag_mscale flag_sscale
flag_rscale flag_ymscale flag_ysscale
clear flag_yrscale flag_biased flag_ssdev flag_corr flag_cov flag_kdisp
clear C IA IB
```

File 12 pls_reverse.m

```
function yhat = pls_reverse(x,m,T,W,P,Q)
□
% takes the predictor and reverses the procedure
□
% x - predictors
□
% m - number of stages
□
% T,W,P,Q - previous mappings

y = 0;
[j, i] = size(x);
```

```
for h = 1:m

   T(:,h) = x * W(:,h);
   x = x - T(:,h) * P(:,h)';
   y  = y + T(:,h) * Q(:,h)';

end
yhat = y;
```

File 13 pls_onestep.m

```
function [t,p,q,x,y,w,D] = pls_onestep(x,y)

% ------------------------------------------------------------------
% ------------------------------------------------------------------

% t    is score vector
% w    is weight vector
% p    is loading vector
% q    is regression coefficient
% V    is eigenvector
% D    is largest eigenvalue

% ------------------------------------------------------------------
% ------------------------------------------------------------------

[m,n] = size(x'*y*y'*x);
B = eye(m);

% compute score and loading vector
OPTIONS.disp=0;
[V,D] = eig((x'*y*y'*x)/m^2);
[V,D] = sorteig(V,D);
D = D(1);
%[V,D] = eigs((x'*y*y'*x)/m^2,B,1,'LM',OPTIONS );
%[V,D] = eigs(x'*y*y'*x,B,1,'LM',OPTIONS );
w = V(:,1);
t = x*w;
p = x'*t/(t'*t);

% reduce dependent matrix
x = x - t*p';

% regression coefficient on t vector
q = y'*t/(t'*t);

% reduce independent matrix
y = y - t*q';

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

File 14 perturb.m

```
function x = perturb(vec,mag,flag,A,B)
```

```
%
% vec    - input signal vector want to change (observations * 1)
%
% mag    - magnitude of perturbation (0 - 1)
%
% flag   - type of change
%

%
[m,n] = size(vec);
%
if n ~= 1, disp('Vector position error'), break, end

switch lower(flag)

case 'ramp'

   if nargin <= 3,

      slope = mag/(m-1);
      beta  = (mag - slope*(1+m))/2;
      indices = 1:m;
      x = slope*indices + beta;

   elseif nargin == 4,

      slope = mag/(m-A);
      beta  = (mag - slope*(m+A))/2;
      indices = A:m;
      x(1:A-1) = 0;
      x(indices) = slope*indices + beta;

   elseif nargin == 5,

      slope = mag/(B-A);
      beta  = (mag - slope*(B+A))/2;
      indices1 = A:B;
      indices2 = B+1:m;
      x(1:A-1) = 0;
      x(indices1) = slope*indices1 + beta;
      x(indices2) = mag;

   end

   x = x';
   x = vec + x;

case 'step'

   if nargin <= 3,
      A = 1;
   end

   x = vec;
   x(A:m) = x(A:m) + mag;

end
```

File 15 mr.m

```
function [mr] = mr(P,T)
%
% do multiple regression
% P  -- predictors
% T  -- target
% mr -- multiple regression results (structure)

tic

mr.x=[];
mr.beta=[];
mr.bint=[];
mr.r=[];
mr.rint=[];
mr.stats=[];
mr.time=[];
mr.P = [];
mr.Y = [];
mr.T = [];

mr.P = P;
mr.T = T;

[mr.x,mr.beta,mr.bint,mr.r,mr.rint,mr.stats]=multiple_reg(P,T);

% answer equals mr.x*beta
% can also use regstats(y,x,'linear')

mr.Y = mr.x*mr.beta;
mr.time = toc;
```

File 16 inputs.m

```
function [x,TransMatrix] = inputs(predictor,flag,m,criterion)
%%
%% factors
%% pca
%% same inputs returned
TransMatrix = [];

```

```
switch lower(flag)

case 'factors'


   % factor scores

   eval(['load pattern5_21st'])
   %eval(['load pattern' int2str(m)])

   % create patterns
   if isempty(pattern)
      R = corrcoef(predictor);
      [pattern,m,c] = minresv4(R,m,3000,.001,0,0); save pattern pattern m c;
   end

   Z = predictor;
   F = inv(pattern'*pattern)*pattern'*Z';
   F = F';
   x = F;

case 'pca'

   % pca scores

   [pcs,newdata,variances,t2] = princomp(predictor);
   x = newdata(:,1:m);
   TransMatrix = pcs;
%x = newdata(:,[1 5 6 12 18]);

case 'same'

   x = predictor;

case 'highcorr' % highest correlated variables with heat rate

   R = corrcoef([predictor criterion]);
   [m1,n] = size(R);
   vec = R(m1,:);
   clear R
   vec(m1) = [];
   temp = [];
   for i = 1:m
      [C,IA,IB]=intersect((max(abs(vec))),abs(vec));
      temp = [temp IB];
      vec(IB) = 0;
   end
   disp(['Highest correlated variable indices: ' int2str(temp)])
   x = predictor(:,temp);

case 'prinvar' % principal variables

   w = [];
   [m1,n] = size(predictor);
   vec = diag(predictor'*criterion*criterion'*predictor);
   for i = 1:m
         [a,j] = max(vec);
         w = [w j];
      vec(j) = 0;
   end
   disp(['Principal variable indices: ' int2str(w)])
```

```
    x = predictor(:,w);

end
```

File 17 Group_Setpoint.m

```
% Group_SetPoint


%% setpoint variables


Ginfo(1).variables_sp = {

    'CNDR HW LVL 1C                '

    'CNDR HW LVL 2C                '

    'CNDR A CLEAN FACTOR 1         '
    'CNDR B CLEAN FACTOR 1         '
    'CNDR C CLEAN FACTOR 1         '
    'CNDR A CLEAN FACTOR 2         '
    'CNDR B CLEAN FACTOR 2         '
    'CNDR C CLEAN FACTOR 2         '
    'CNDR CW FLW 2                 '
    'CNDR CW FLW 1                 '
    'CNDR A CW TMP DIFF 2          '
    'CNDR B CW TMP DIFF 2          '
    'CNDR C CW TMP DIFF 2          '
    'CNDR A CW TMP DIFF 1          '
    'CNDR B CW TMP DIFF 1          '
    'CNDR C CW TMP DIFF 1          '
    'CNDR A CW TMP RISE 2          '
    'CNDR B CW TMP RISE 2          '
    'CNDR C CW TMP RISE 2          '
    'CNDR A CW TMP RISE 1          '
    'CNDR B CW TMP RISE 1          '
    'CNDR C CW TMP RISE 1          '
    'CNDR A TTD 2                  '
    'CNDR B TTD 2                  '
    'CNDR C TTD 2                  '
    'CNDR A TTD 1                  '
    'CNDR B TTD 1                  '
    'CNDR C TTD 1                  '
    'CONDENSATE PUMP(A) FLOW       '
    'CONDENSATE PUMP(B) FLOW       '
    'CONDENSATE PUMP(C) FLOW       '
    'CW A/B DIFF TEMP'
    'CW C/D DIFF TEMP'
    'CNDSR (A) A/B CW DIFF TMP'
    'CNDSR (A) C/D CW DIFF TMP'
    'CNDSR (B) A/B CW DIFF TMP'
    'CNDSR (B) C/D CW DIFF TMP'
    'CNDSR (C) A/B CW DIFF TMP'
    'CNDSR (C) C/D CW DIFF TMP'
    'CNDSR (A) A/B TTD'
    'CNDSR (A) C/D TTD'
    'CNDSR (B) A/B TTD'
    'CNDSR (B) C/D TTD'
```

```
   'CNDSR (C) A/B TTD'
   'CNDSR (C) C/D TTD'
}';

Ginfo(2).variables_sp = {
   'FW PMP A FW TMP RISE          '
   'FW PMP B FW TMP RISE          '
   'FW PMP A IN ENTH              '
   'FW PMP B IN ENTH              '
   'FW PMP A OUT ENTH             '
   'FW PMP B OUT ENTH             '
   'FW PMP A IN ENTR              '
   'FW PMP B IN ENTR              '
   'FW PMP A FW SP VOL            '
   'FW TO SG PRESS               '
   'FEEDWATER PRESS 1            '
   'FEEDWATER PRESS 2            '
   'FP TURB(A) STEAM FLOW        '
   'FP TURB(B) STEAM FLOW        '
   'FWPT A HP STEAM INLET        '
   'FWPT B HP STEAM INLET        '
   'FP TURB(A) EXH PRESS         '
   'FP TURB(B) EXH PRESS         '
   'FWPT A 1ST STG PRESS         '
   'FWPT B 1ST STG PRESS         '
   'FW PUMP(A) SPEED             '
   'FW PUMP(B) SPEED             '
   'FWPT A HP SV CHEST TEMP      '
   'FWPT B HP SV CHEST TEMP      '
   'FW PUMP(A) FLOW              '
   'FW PUMP(B) FLOW              '
   'FW PUMP(A) SUCTION PRESS     '
   'FW PUMP(B) SUCTION PRESS     '
   'FW PUMP(A) PRESSURE          '
   'FW PUMP(B) PRESSURE          '
   'FW PUMP(A) HEAD              '
   'FW PUMP(B) HEAD              '
   'FW PMP A SUCT FLOW DP        '
   'FW PMP B SUCT FLOW DP        '
   'FW PUMP(A) DISCHARGE TEM     '
   'FW PUMP(B) DISCHARGE TEM     '
}';

Ginfo(3).variables_sp = {
}';

Ginfo(4).variables_sp = {
   'HTR NO 1A TERM TEMP DIFF     '
   'HTR NO 1B TERM TEMP DIFF     '
   'HTR NO 1C TERM TEMP DIFF     '
   'HTR NO 2A TERM TEMP DIFF     '
   'HTR NO 2B TERM TEMP DIFF     '
   'HTR NO 2C TERM TEMP DIFF     '
   'HTR NO 3A TERM TEMP DIFF     '
   'HTR NO 3B TERM TEMP DIFF     '
   'HTR NO 3C TERM TEMP DIFF     '
   'HTR NO 4A TERM TEMP DIFF     '
   'HTR NO 4B TERM TEMP DIFF     '
   'HTR NO 4C TERM TEMP DIFF     '
   'HTR DRAIN PUMP(A) FLOW       '
   'HTR DRAIN PUMP(B) FLOW       '
   'HTR DRN PUMP(A) PRESS        '
```

```
    'HTR DRN PUMP(B) PRESS          '
    'HTR DRN PUMP(A) TEMP           '
    'HTR DRN PUMP(B) TEMP           '
    'HTR NO 5A TERM TEMP DIFF       '
    'HTR NO 5B TERM TEMP DIFF       '
    'HTR NO 6A TERM TEMP DIFF       '
    'HTR NO 6B TERM TEMP DIFF       '
    'HTR NO 7A TERM TEMP DIFF       '
    'HTR NO 7B TERM TEMP DIFF       '
    'TOTAL DRAIN & COND FLOW'
    'DRN PUMP(A) ABS DIS PRES       '
    'DRN PUMP(B) ABS DIS PRES       '
    'HTR DRN PUMP(A) DEL PRES       '
    'HTR DRN PUMP(B) DEL PRES       '
}';

Ginfo(5).variables_sp = {
}';

Ginfo(6).variables_sp = {
    'RH A 1ST STG HT REMOVED        '
    'RH B 1ST STG HT REMOVED        '
    'RH C 1ST STG HT REMOVED        '
    'RH D 1ST STG HT REMOVED        '
    'RH A 2ND STG HT REMOVED        '
    'RH B 2ND STG HT REMOVED        '
    'RH C 2ND STG HT REMOVED        '
    'RH D 2ND STG HT REMOVED        '
    'RH A 1ST STG HT STM HEAT FLW   '
    'RH B 1ST STG HT STM HEAT FLW   '
    'RH C 1ST STG HT STM HEAT FLW   '
    'RH D 1ST STG HT STM HEAT FLW   '
    'RH A 1ST STG DRN HEAT FLW      '
    'RH B 1ST STG DRN HEAT FLW      '
    'RH C 1ST STG DRN HEAT FLW      '
    'RH D 1ST STG DRN HEAT FLW      '
    'RH A 2ND STG DRN HEAT FLW      '
    'RH B 2ND STG DRN HEAT FLW      '
    'RH D 2ND STG DRN HEAT FLW      '
    'RH STM OUT FLW                 '
    'RH STM OUT FLW PER MSR         '
    'RH STM OUT FLW TO LP TURB      '
    'RH A 2ND STG STM OUT HEAT FLW  '
    'RH B 2ND STG STM OUT HEAT FLW  '
    'RH C 2ND STG STM OUT HT FLW    '
    'RH D 2ND STG STM OUT HT FLW    '
    'RH A 2ND STG STM OUT FLW       '
    'MSR(A) 2ND STG STM FLOW        '
    'MSR(A) 1ST STG STM FLOW        '
    'MSR(B) 2ND STG STM FLOW        '
    'MSR(B) 1ST STG STM FLOW        '
    'MSR(C) 2ND STG STM FLOW        '
    'MSR(C) 1ST STG STM FLOW        '
    'MSR(D) 2ND STG STM FLOW        '
    'MSR(D) 1ST STG STM FLOW        '
    'MSR-A 2ND STG STM HT FLO       '
    'MSR-B 2ND STG STM HT FLO       '
    'MSR-C 2ND STG STM HT FLO       '
    'MSR-D 2ND STG STM HT FLO       '
    'HP TURB 1ST STG PRESS          '
    'TURB STM CHEST PRESS           '
    'MSR A 2ND STG RHTG STM PRESS   '
```

```
    'MSR A 1ST STG RHTG STM PRESS '
    'MSR B 2ND STG RHTG STM PRESS '
    'MSR B 1ST STG RHTG STM PRESS '
    'MSR C 2ND STG RHTG STM PRESS '
    'MSR C 1ST STG RHTG STM PRESS '
    'MSR D 2ND STG RHTG STM PRESS '
    'MSR D 1ST STG RHTG STM PRESS '
    'MSR(A) 1ST STG STM TEMP      '
    'MSR(B) 1ST STG STM TEMP      '
    'MSR A 2ND STG RHTG STM TEMP  '
    'MSR B 2ND STG RHTG STM TEMP  '
    'MSR A BETWEEN STG TEMP       '
    'MSR B BETWEEN STG TEMP       '
    'MSR C 2ND STG RHTG STM TEMP  '
    'MSR(C) 1ST STG STM TEMP      '
    'MSR D 2ND STG RHTG STM TEMP  '
    'MSR(D) 1ST STG STM TEMP      '
    'MSR C BETWEEN STG TEMP       '
    'MSR D BETWEEN STG TEMP       '
    'MSR (A) DRAIN TEMP           '
    'MSR (B) DRAIN TEMP           '
    'MSR (C) DRAIN TEMP           '
    'MSR (D) DRAIN TEMP           '
    'FP TURB HRH STM TEMP         '
    'MSR-A SEPARATOR DRN ENTH     '
    'MSR-B SEPARATOR DRN ENTH     '
    'MSR-C SEPARATOR DRN ENTH     '
    'MSR-D SEPARATOR DRN ENTH     '
    'MSR-A SEPARATOR STM ENTH     '
    'MSR-B SEPARATOR STM ENTH     '
    'MSR-C SEPARATOR STM ENTH     '
    'MSR-D SEPARATOR STM ENTH     '
    'MSR-A 1ST STAGE DRN ENTH     '
    'MSR-B 1ST STAGE DRN ENTH     '
    'MSR-C 1ST STAGE DRN ENTH     '
    'MSR-D 1ST STAGE DRN ENTH     '
    'MSR-A 1ST STAGE STM ENTH     '
    'MSR-B 1ST STAGE STM ENTH     '
    'MSR-C 1ST STAGE STM ENTH     '
    'MSR-D 1ST STAGE STM ENTH     '
    'MSR-A 2ND STG DRAIN ENTH     '
    'MSR-B 2ND STG DRAIN ENTH     '
    'MSR-C 2ND STG DRAIN ENTH     '
    'MSR-D 2ND STG DRAIN ENTH     '
    'MSR A 2ND STG RHT TTD'
    'MSR B 2ND STG RHT TTD'
    'MSR C 2ND STG RHT TTD'
    'MSR D 2ND STG RHT TTD'
}';

Ginfo(7).variables_sp = {
    'S/G 1 FW MASS FLOW RATE      '
    'S/G 2 FW MASS FLOW RATE      '
    'S/G 1 BLWDWN MASS FLOW RATE  '
    'S/G 2 BLWDWN MASS FLOW RATE  '
    'S/G 1 ENERGY REMOVAL RATE    '
    'S/G 2 ENERGY REMOVAL RATE    '
    'S/G PRESSURE 1               '
    'S/G PRESSURE 2               '
    'AVG MAIN STEAM PRESSURE      '
    'MAIN STEAM PRESSURE (A)      '
    'MAIN STEAM PRESSURE (B)      '
```

```
    'SG 1 FEEDWATER TEMP          '
    'SG 2 FEEDWATER TEMP          '
    'SG 1 LEVEL                   '
    'SG2 LEVEL                    '
    'MAIN STEAM TEMP (A)          '
    'MAIN STEAM TEMP (B)          '
    'MAIN STEAM TEMP (C)          '
    'MAIN STEAM TEMP (D)          '
    'AVG MAIN STEAM TEMP          '
}';

Ginfo(8).variables_sp = {
    'HP TURBINE PRESSURE RATIO'
    'TURB ECCENTRICITY             '
    'TURB ROTOR EXPANSION          '
    'TURB SHELL EXPANSION          '
    'TURB DIFF EXPANSION           '
    'TRB HEAT RATE GROSS UNCORR    '
    'TRB HEAT RATE NET UNCORR      '
    'TRB LP EXH PRS CORR FACTOR    '
    'TRB HEAT RATE NET CORR        '
    'TRB / GEN EFF                 '
    'TURB LO COR INLT TEMP         '
    'TURB LO CLR OUTLT TEMP        '
}';

Ginfo(9).variables_sp = {
    'AVG CONDENSER PRESSURE       '
    'TOTAL FEEDWATER FLOW         '
    'TOTAL MAIN STEAM FLOW        '
    'CONDENSATE PUMP(A) FLOW      '
    'CONDENSATE PUMP(B) FLOW      '
    'CONDENSATE PUMP(C) FLOW      '
    'FW PUMP(A) FLOW              '
    'FW PUMP(B) FLOW              '
    'TOTAL STEAM FLOW'
    'TOTAL FEED FLOW'
    'TOTAL DRAIN & COND FLOW'
    'AVERAGE STEAM GEN PRESS'
}';

% initialize cells

Group_info.variables_setpoint = [];
Group_info.variables_sp_group_number = [];
Group_info.variables_sp_index = [];
Group_info.total_number_variables_sp = [];
Group_info.sp_y.data = [];
Group_info.sp_y.group_number = [];
Group_info.sp_y.index = [];
Group_info.sp_y.description = [];
Group_info.sp_x.data = [];
Group_info.sp_x.group_number = [];
Group_info.sp_x.index = [];
Group_info.sp_x.description = [];

temp = 0;
for i = 1:length(Ginfo)
   temp = temp + length(Ginfo(i).variables_sp);
   [C,IA,IB] = intersect(Group(i).ws_description,Ginfo(i).variables_sp);
   Group_info.variables_setpoint = [Group_info.variables_setpoint
Ginfo(i).variables_sp];
```

```
   Group_info.variables_sp_group_number = [Group_info.variables_sp_group_number
i.*ones(1,length(IA))];
   Group_info.variables_sp_index = [Group_info.variables_sp_index IA];
end

for i = 1:length(Group_info)
   Group_info.total_number_variables_sp=temp;
end

Group_info.sp_y.data = [];
Group_info.sp_y.group_number = [];
Group_info.sp_y.index = [];
Group_info.sp_y.description = [];

for i = 1:length(Group)
   [C,IA,IB] = intersect(Group(i).ws_description,Ginfo(i).variables_sp);
   Group_info.sp_x.data = [Group_info.sp_x.data Group(i).ws_data(:,IA)];
   Group_info.sp_x.group_number = [Group_info.sp_x.group_number
i.*ones(1,length(IA))];
   Group_info.sp_x.index = [Group_info.sp_x.index IA];
   Group_info.sp_x.description = [Group_info.sp_x.description
Group(i).ws_description(IA)];
end

clear i temp Ginfo C IA IB
```

File 18 Group_fix.m

```
% Group_fix
%
% fixes data with bad indices using linear interpolation between good points

temp = 0;
for i = 1:length(Group_info.variables_bi_group_number)

   Variable = Group_info.variables_bad_indices(i);

   [C,IA,IB] = intersect(Variable,Group_info.variables_state);
   [C2,IA2,IB2] = intersect(Variable,Group_info.variables_setpoint);

   if ~isempty(IB),  flag = 1; temp = temp + 1; end
   if ~isempty(IB2), flag = 2; temp = temp + 1; end
   if ~isempty(IB) & ~isempty(IB2), disp('Flag error'), break, end

       if flag == 1
      [C3,IA3,IB3] = intersect(Variable,Group_info.state_x.description);
          x = Group_info.state_x.data(:,IB3);
      Group_info.state_x.data(:,IB3) = addpoints(x);
   elseif flag == 2
      [C3,IA3,IB3] = intersect(Variable,Group_info.sp_x.description);
          x = Group_info.sp_x.data(:,IB3);
              Group_info.sp_x.data(:,IB3) = addpoints(x);
   end

end

if temp ~= length(Group_info.variables_bi_group_number), disp('Error'), end


%   Group_number = Group_info.variables_bi_group_number(i);
```

```
%   Bad_index     = Group_info.variables_bi_index(i);

%   variable =
clear C IA IB C2 IA2 IB2 Variable ans flag i temp C3 IA3 IB3 x
```

File 19 fmap.m

```
function map = fmap(predictor,criterion,flag,num_stages)
%

%
% use P as predictors and T as target
%

%
map.P = predictor;
map.T = criterion;
map.flag = flag;

switch lower(flag)

case 'mlr'
   % do multivariate linear regression
   map.results = mr(map.P,map.T);

case 'mmr'
   % do multivariate multiple regression
   map.results.P = map.P;
   map.results.T = map.T;
   tic
   map.results.beta = inv(map.P'*map.P)*map.P'*map.T;
   map.results.Y = map.P*map.results.beta;
   map.results.time = toc;

case 'nn'

   [tnet,trd,time,total_epochs,nodes] = trainNN(predictor,criterion);
   map.results.Y = sim(tnet,map.P')';
   map.results.tnet = tnet;
   map.results.trd = trd;
   map.results.time = time;
   map.results.total_epochs = total_epochs;
   map.results.nodes = nodes;

case 'pls'

   [m,n] = size(predictor);
   x = predictor;
   y = criterion;
   T =[]; P =[]; Q = []; W = []; X = x; Y = y; D = [];
   count = 0;
   eigenv = [];
   map.results.actual_stages = num_stages;

   tic
   %   Coeff = 1;
   Coeff = 1e-6;
   Dold = 0;

   for i = 1:num_stages,
```

```
      [t{i},p{i},q{i}, x, y, w{i},D] = pls_onestep(x,y);
      if num_stages == 100
         if abs(D-Dold) >=Coeff,
            count = count + 1;
         elseif abs(D-Dold) < Coeff
            map.results.actual_stages = count;
            break;
         end
         Dold = D;
      end
      T = [T t{i}];
      P = [P p{i}];
      Q = [Q q{i}];
      W = [W w{i}];
      X = [X x];
      Y = [Y y];
      eigenv = [ eigenv D];
   end

   map.results.error = y;
   %T*Q' + err
   map.results.num_stages = num_stages;
   map.results.eigenvalues = eigenv;
   map.results.T = T;
   map.results.P = P;
   map.results.Q = Q;
   map.results.W = W;
   map.results.Y = T*Q'; % yhat
   map.results.time = toc;
   %mse = mean((yo-yhat).^2)
   %plot([yo yhat]);

end
```

File 20 addpoints.m

```
function [y] = addpoints(y)


xz  = find(y==0);

x   = find(y);

y1  = y;

len = length(x);


% fix any bad points inbetween

for i = 1:(len-1)

   temp = find( (x(i) < xz) & (xz < x(i+1)) );
   if ~isempty(temp)
      delta_x = x(i) - x(i+1);
      delta_y = y(x(i))-y(x(i+1));
      slope = delta_y/delta_x;
      beta  = y(x(i)) - slope*x(i);
```

```
      for j = 1:length(temp)
         y1(xz(temp(j))) = (xz(temp(j)))*slope + beta;
      end
   end
end

% fix end points, if necessary
% correct left endpoint
temp = find(xz < x(1));
if ~isempty(temp)
   delta_x = x(1) - (x(1)+1);
   delta_y = y1(x(1))-y1(x(1)+1);
   slope = delta_y/delta_x
   beta  = y1(x(1)) - slope*x(1)
   for j = 1:length(temp)
      y1(xz(temp(j))) = (xz(temp(j)))*slope + beta;
   end
end

% correct right endpoint
temp = find(xz > x(len));
if ~isempty(temp)
   delta_x = (x(len)-1) - x(len);
   delta_y = y1(x(len)-1)-y1(x(len));
   slope = delta_y/delta_x
   beta  = y1(x(len)) - slope*x(len)
   for j = 1:length(temp)
      y1(xz(temp(j))) = (xz(temp(j)))*slope + beta;
   end
end

y = y1;
```

Vita

Claude Irvine was born in Pittsburgh, Pennsylvania on October 31st, 1956. In 1986 he received the Bachelor of Science in Nuclear Engineering with a minor in mathematics from The University of Tennessee at Knoxville. While serving in the Air Force as a nuclear research officer, he entered the Master's program in nuclear weapon effects at the Air Force Institute of Technology at Wright-Patterson AFB, Ohio. In January of 1996, he returned to The University of Tennessee to pursue the Doctorate of Philosophy in Nuclear Engineering, specializing in nuclear data analysis and applied Artificial Intelligence.