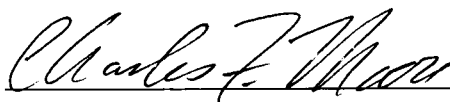


To the Graduate Council:

I am submitting herewith a thesis written by John Wesley Cox entitled "Use of Controller Performance Monitoring and Data Analysis Techniques in the Detection and Diagnosis of PID Control Loop Problems." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Chemical Engineering.



Dr. Charles F. Moore, Major Professor

We have read this thesis
and recommend its acceptance:




Accepted for the Council:



Associate Vice Chancellor
and Dean of the Graduate School

USE OF CONTROLLER PERFORMANCE MONITORING AND DATA ANALYSIS
TECHNIQUES IN THE DETECTION AND DIAGNOSIS OF PID CONTROL LOOP
PROBLEMS

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

John Wesley Cox
December 1998

DEDICATION

This work is dedicated to my wife Holli and son Wes, two precious gifts with which God has blessed me.

ACKNOWLEDGEMENTS

I would like to express my gratitude to a number of individuals for their support during my graduate education and thesis preparation.

First, I would like to thank my committee chairman Charlie Moore for his involvement throughout the course of this work. Despite a hectic schedule, he has taken significant time to meet with me, provide me with excellent technical input, and give me support and encouragement in completing this work. His advice, good nature, and technical suggestions greatly improved this work and the experience in preparing it. I would also like to thank Dr. Duane Bruns and Dr. Fred Weber for serving on my committee and providing helpful input to various portions of this work.

Thanks also to all members of the Eastman Chemical Advanced Controls Technology group in Kingsport for their encouragement and unending support during my thesis research and preparation. Thanks in particular to Dr. Jerry Cole, Dr. Jim Downs, Dr. Vera McGill, David Miller, Dr. Mike Paulonis, Dr. Ernie Vogel, and Joey Watson for their technical ideas, practical advice, guidance in the details of thesis preparation, many words of encouragement, and prayers.

My graduate work has been facilitated through the use of several resources within Eastman Chemical, for which I am very grateful.

Finally, thanks to my wife Holli for her patience, prayers,
understanding, and faithful support throughout my graduate work.
Thanks to my son Wes for his encouraging presence, smile, innocence,
and help in providing me books.

ABSTRACT

Optimum performance of PID control loops is critical to the successful operation of a continuous chemical process. Poorly-performing PID control loops significantly increase process variability and can even diminish the performance of advanced control strategies. Due to the tremendous number of PID control loops in large scale chemical processes, human monitoring of all the PID controllers in a given process is infeasible, and many PID controller problems go undetected. An automated way of detecting poorly-performing PID control loops (as well as diagnosis techniques to help determine the associated controller problem) would be extremely beneficial to the operation of continuous chemical processes.

In this work, algorithms and techniques will be sought which can automatically identify poorly-performing PID control loops and aid in diagnosis of 4 specific problems (valve stiction, valve hysteresis, disturbances, and tuning) which often cause poor performance. This work is limited to PID flow/pressure control loops with: 1) negligible dead time, 2) fast, first order responses, and 3) a performance objective of fast setpoint tracking with no steady-state offset.

To begin the performance monitoring and problem diagnosis studies, multiple simulated and real process data sets exhibiting poor control performance and the 4 problems of interest are obtained. A general Performance Monitoring Algorithm (PMA) is then developed which alarms

poor controller performance based on the controller percent of time oscillating (in real time). The usefulness of cross correlation, Principal Components Analysis (PCA), and descriptive statistics for diagnosing controller problems is then studied through application to the process data sets.

Based on the monitoring studies, the developed PMA is successful in detecting poor controller performance. The PMA alarming ability is independent of the controller oscillation frequency and the type of problem inducing the poor performance. The PMA is a robust detection algorithm and is easily tuned to prevent false performance alarms.

Based on the problem diagnosis studies, cross correlation and PCA provide positive diagnosis for hysteresis problems. Descriptive statistics related to controller error and oscillation frequency provide positive diagnosis for stiction problems and for some disturbance problems. When cross correlation, PCA, and descriptive statistics are used in concert, the most probable controller problem can be diagnosed with a limited degree of certainty for all cases except high frequency controller oscillations. While more work is needed, the studied diagnosis techniques can be used to identify specific problems and provide valuable controller troubleshooting assistance for many cases.

TABLE OF CONTENTS

CHAPTER		PAGE
1	INTRODUCTION	1
2	LITERATURE REVIEW	6
	2.1 The Need for Controller Performance Monitoring/Problem Diagnosis	6
	2.2 Minimum Integral Error Criteria	7
	2.3 Minimum Variance Criteria	8
	2.4 Statistical Variance Criteria	11
	2.5 Controller Oscillation Detection	12
	2.6 Other Performance Monitoring Methods	14
	2.7 Correlation Analysis	15
	2.8 Spectral Analysis	18
	2.9 Principal Components Analysis (PCA)	19
3	DESCRIPTION OF COMMON CONTROLLER PROBLEMS AND PROCESS DATA SET DEVELOPMENT	22
	3.1 Introduction to Flow/Pressure Controller Performance	22
	3.2 Description of Controller Configuration and Conventions	22
	3.3 Description of Common Controller Problems	24
	3.4 Development of Simulated Process Data Sets	28
	3.5 Selection of Real Process Data Sets	35
4	PERFORMANCE MONITORING ALGORITHM TESTING AND DEVELOPMENT	43
	4.1 Desired Characteristics for the Monitoring Algorithm	43
	4.2 Initial Testing of the Hagglund CLPM	44
	4.3 Limitations of Hagglund CLPM in Relation to Desired Algorithm	49
	4.4 Performance Monitoring Algorithm (PMA) Development and Testing	50
	4.5 Advantages of the Developed Performance Monitoring Algorithm	56
	4.6 Complementary Analyses for the Performance Monitoring Algorithm	57
	4.7 Extending the PMA to Other Types of Control Loops	58
	4.8 Other Potential Monitoring Algorithm Needs	60

5	APPLICATION OF CROSS CORRELATION ANALYSIS IN DIAGNOSIS OF COMMON CONTROLLER PROBLEMS	62
5.1	Introduction to Cross Correlation Analysis	62
5.2	Data Preconditioning with a High Pass Filter (Prior to Analysis)	63
5.3	Application of PV/OP Cross Correlation Analysis	64
5.4	Relation of PV/OP Correlation Patterns to Control Loop Problems	70
5.5	Other Valuable Cross Correlation Analyses	72
6	APPLICATION OF PRINCIPAL COMPONENTS ANALYSIS IN DIAGNOSIS OF COMMON CONTROLLER PROBLEMS	77
6.1	Introduction to Principal Components Analysis (PCA)	77
6.2	Application of Selected PCA Methods	78
6.3	Usefulness of Selected PCA Methods for Problem Diagnosis	80
7	APPLICATION OF SELECTED STATISTICS IN DIAGNOSIS OF COMMON CONTROLLER PROBLEMS	84
7.1	Motivation for Application of Statistics	84
7.2	Error and Oscillation Frequency Statistics	84
7.3	PV/OP and OP/PV Spread Statistics	88
7.4	Usefulness of the Selected Statistics for Problem Diagnosis	89
7.5	Problem Diagnosis Through Combination of Cross Correlation, PCA, and Descriptive Statistics	93
8	CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE STUDY	95
8.1	Conclusions	95
8.2	Recommendations for Future Study	97
	LIST OF REFERENCES	101
	APPENDICES	104
	A. SIMULATED AND REAL PROCESS DATA SETS	105
	B. PV/OP CROSS CORRELATIONS FOR ALL SIMULATED AND REAL PROCESS DATA SETS	127
	C. MATLAB M-FILES USED FOR MONITORING ALGORITHM APPLICATION AND DATA ANALYSIS APPLICATION	139
	D. PCA COMPARISONS FOR SIMULATED AND REAL PROBLEM DATA SETS (RELATIVE TO GOOD PERFORMANCE)	159

VITA	170
------------	-----

LIST OF TABLES

TABLE	PAGE
3.1. Simulated process data sets to be used in development of controller problem diagnosis techniques	32
3.2. Real process data sets to be used in development of controller problem diagnosis techniques	37
3.3. Real process data sets to be used in development of performance monitoring algorithm	41
5.1. Descriptive parameters from cross correlation analysis ...	71
6.1. Percent variance captured by principal components for PCA models applied to PV/OP time-shifted matrices	81
7.1. Average error and oscillation frequency statistics	87
7.2. PV/OP and OP/PV spread statistics	90

LIST OF FIGURES

FIGURE		PAGE
3.1.	Example configuration for flow control loop	23
3.2.	Hysteresis illustration	26
3.3.	Simulated data for valve stiction problem (STIC1S): (a) SP, PV, and OP data, (b) PV/OP characteristic plot ...	33
3.4.	Simulated data for poor tuning problem (TUNE1S): (a) SP, PV, and OP data, (b) PV/OP characteristic plot ...	34
3.5.	Real data for valve hysteresis problem (HYST1R): (a) SP, PV, and OP data, (b) PV/OP characteristic plot ...	38
3.6.	Real data for process disturbance problem (DIST1R): (a) SP, PV, and OP data, (b) PV/OP characteristic plot ...	39.
3.7.	Real data for controller with good performance (GOOD1R): (a) SP, PV, and OP data, (b) PV/OP characteristic plot ...	40
4.1.	Hagglund CLPM applied to real control loop data (STIC3R): (a) SP and PV data, (b) calculated CLPM rate of oscillation - no alarm given	45
4.2.	Hagglund CLPM applied to real control loop data (TUNE1R): (a) subset of SP and PV data (total data is 3 hours), (b) calculated CLPM rate of oscillation - no alarm given	47
4.3.	PMA applied to real control loop data (STIC3R): (a) SP and PV data, (b) calculated PMA percent time oscillating - alarm given just after 2 hours	54
4.4.	PMA applied to real control loop data (HYST4R): (a) subset of SP and PV data (total data is 8 hours), (b) calculated PMA percent time oscillating - alarm given just before 3 hours	55
4.5.	Frequency and peak deviation analyses applied to real control loop data (STIC3R): (a) half period oscillation frequencies, (b) peak deviation from setpoint for each oscillation	59
5.1.	Effect of high pass filter with cross correlation analysis, for controller with good performance (GOOD1R): (a) raw PV/OP data, (b) PV/OP data filtered with 20 min. constant, (c) cross correlation for raw data, (d) cross correlation for filtered data	65

5.2.	Comparison of PV/OP cross correlations (simulated data): (a) PV/OP correlation for STIC1S, (b) PV/OP correlation for HYST1S, (c) PV/OP correlation for TUNE1S, (d) PV/OP correlation for DIST1S	67
5.3.	Comparison of PV/OP cross correlations (real data): (a) PV/OP correlation for STIC3R, (b) PV/OP correlation for HYST3R, (c) PV/OP correlation for TUNE1R, (d) PV/OP correlation for DIST1R	68
5.4.	Comparison of PV/OP cross correlations (simulated/real data): (a) PV/OP correlation for GOOD1S, (b) PV/OP correlation for GOOD2S, (c) PV/OP correlation for GOOD1R, (d) PV/OP correlation for GOOD2R	69
5.5.	Comparison of PV/SP cross correlations for poor (HYST3R) and good (GOOD2R) performance: (a) raw PV/SP data for HYST3R, (b) raw PV/SP data for GOOD2R, (c) PV/SP cross correlation for HYST3R, (d) PV/SP cross correlation for GOOD2R	74
5.6.	Comparison of error/delta OP cross correlations for poor (TUNE1R) and good (GOOD2R) performance: (a) subset of raw PV/OP data for TUNE1R, (b) raw PV/OP data for GOOD2R, (c) error/dOP cross correlation for TUNE1R, (d) error/dOP cross correlation for GOOD2R	75
7.1.	Summary flowchart for problem diagnosis logic	94
A.1.	Simulated data for valve stiction problem (STIC1S and STIC2S): (a) SP/PV/OP data for STIC1S, (b) SP/PV/OP data for STIC2S, (c) PV/OP characteristic plot for STIC1S, (d) PV/OP characteristic plot for STIC2S	107
A.2.	Simulated data for valve stiction problem (STIC3S and STIC4S): (a) SP/PV/OP data for STIC3S, (b) SP/PV/OP data for STIC4S, (c) PV/OP characteristic plot for STIC3S, (d) PV/OP characteristic plot for STIC4S	108
A.3.	Simulated data for valve stiction problem (STIC5S and STIC6S): (a) SP/PV/OP data for STIC5S, (b) SP/PV/OP data for STIC6S, (c) PV/OP characteristic plot for STIC5S, (d) PV/OP characteristic plot for STIC6S	109
A.4.	Simulated data for valve hysteresis problem (HYST1S and HYST2S): (a) SP/PV/OP data for HYST1S, (b) SP/PV/OP data for HYST2S, (c) PV/OP characteristic plot for HYST1S, (d) PV/OP characteristic plot for HYST2S	110
A.5.	Simulated data for valve hysteresis problem (HYST3S and HYST4S): (a) SP/PV/OP data for HYST3S, (b) SP/PV/OP data for HYST4S, (c) PV/OP characteristic plot for HYST3S, (d) PV/OP characteristic plot for HYST4S	111

A.6.	Simulated data for valve hysteresis problem (HYST5S): (a) SP/PV/OP data for HYST5S, (b) PV/OP characteristic plot for HYST5S	112
A.7.	Simulated data for tuning problem (TUNE1S and TUNE2S): (a) SP/PV/OP data for TUNE1S, (b) SP/PV/OP data for TUNE2S, (c) PV/OP characteristic plot for TUNE1S, (d) PV/OP characteristic plot for TUNE2S	113
A.8.	Simulated data for tuning problem (TUNE3S and TUNE4S): (a) SP/PV/OP data for TUNE3S, (b) SP/PV/OP data for TUNE4S, (c) PV/OP characteristic plot for TUNE3S, (d) PV/OP characteristic plot for TUNE4S	114
A.9.	Simulated data for tuning problem (TUNE5S): (a) SP/PV/OP data for TUNE5S, (b) PV/OP characteristic plot for TUNE5S	115
A.10.	Simulated data for disturbance problem (DIST1S and DIST2S): (a) SP/PV/OP data for DIST1S, (b) SP/PV/OP data for DIST2S, (c) PV/OP characteristic plot for DIST1S, (d) PV/OP characteristic plot for DIST2S	116
A.11.	Simulated data for disturbance problem (DIST3S and DIST4S): (a) SP/PV/OP data for DIST3S, (b) SP/PV/OP data for DIST4S, (c) PV/OP characteristic plot for DIST3S, (d) PV/OP characteristic plot for DIST4S	117
A.12.	Simulated data for disturbance problem (DIST5S and DIST6S): (a) SP/PV/OP data for DIST5S, (b) SP/PV/OP data for DIST6S, (c) PV/OP characteristic plot for DIST5S, (d) PV/OP characteristic plot for DIST6S	118
A.13.	Simulated data for good performance (GOOD1S and GOOD2S): (a) SP/PV/OP data for GOOD1S, (b) SP/PV/OP data for GOOD2S, (c) PV/OP characteristic plot for GOOD1S, (d) PV/OP characteristic plot for GOOD2S	119
A.14.	Real data for valve stiction problem (STIC3R and STIC1R): (a) SP/PV/OP data for STIC3R, (b) SP/PV/OP data for STIC1R, (c) PV/OP characteristic plot for STIC3R, (d) PV/OP characteristic plot for STIC1R	120
A.15.	Real data for valve stiction problem (STIC2R and STIC4R): (a) SP/PV/OP data for STIC2R, (b) SP/PV/OP data for STIC4R, (c) PV/OP characteristic plot for STIC2R, (d) PV/OP characteristic plot for STIC4R	121
A.16.	Real data for valve hysteresis problem (HYST1R and HYST3R): (a) SP/PV/OP data for HYST1R, (b) SP/PV/OP data for HYST3R, (c) PV/OP characteristic plot for HYST1R, (d) PV/OP characteristic plot for HYST3R	122

A.17.	Real data for valve hysteresis problem (HYST2R): (a) SP/PV/OP data for HYST2R, (b) PV/OP characteristic plot for HYST2R	123
A.18.	Real data for tuning problem (TUNE1R): (a) SP/PV/OP data for TUNE1R, (b) PV/OP characteristic plot for TUNE1R	124
A.19.	Real data for disturbance problem (DIST1R and DIST2R): (a) SP/PV/OP data for DIST1R, (b) SP/PV/OP data for DIST2R, (c) PV/OP characteristic plot for DIST1R, (d) PV/OP characteristic plot for DIST2R	125
A.20.	Real data for good performance (GOOD1R and GOOD2R): (a) SP/PV/OP data for GOOD1R, (b) SP/PV/OP data for GOOD2R, (c) PV/OP characteristic plot for GOOD1R, (d) PV/OP characteristic plot for GOOD2R	126
B.1.	PV/OP cross correlations for valve stiction problem: (a) PV/OP correlation for STIC1S, (b) PV/OP correlation for STIC2S, (c) PV/OP correlation for STIC3S, (d) PV/OP correlation for STIC4S	129
B.2.	PV/OP cross correlations for valve stiction problem: (a) PV/OP correlation for STIC5S, (b) PV/OP correlation for STIC6S, (c) PV/OP correlation for STIC1R, (d) PV/OP correlation for STIC2R	130
B.3.	PV/OP cross correlations for valve stiction problem: (a) PV/OP correlation for STIC3R, (b) PV/OP correlation for STIC4R	131
B.4.	PV/OP cross correlations for valve hysteresis problem: (a) PV/OP correlation for HYST1S, (b) PV/OP correlation for HYST2S, (c) PV/OP correlation for HYST3S, (d) PV/OP correlation for HYST4S	132
B.5.	PV/OP cross correlations for valve hysteresis problem: (a) PV/OP correlation for HYST5S, (b) PV/OP correlation for HYST1R, (c) PV/OP correlation for HYST2R, (d) PV/OP correlation for HYST3R	133
B.6.	PV/OP cross correlations for tuning problem: (a) PV/OP correlation for TUNE1S, (b) PV/OP correlation for TUNE2S, (c) PV/OP correlation for TUNE3S, (d) PV/OP correlation for TUNE4S	134
B.7.	PV/OP cross correlations for tuning problem: (a) PV/OP correlation for TUNE5S, (b) PV/OP correlation for TUNE1R	135

B.8.	PV/OP cross correlations for disturbance problem: (a) PV/OP correlation for DIST1S, (b) PV/OP correlation for DIST2S, (c) PV/OP correlation for DIST3S, (d) PV/OP correlation for DIST4S	136
B.9.	PV/OP cross correlations for disturbance problem: (a) PV/OP correlation for DIST5S, (b) PV/OP correlation for DIST6S, (c) PV/OP correlation for DIST1R, (d) PV/OP correlation for DIST2R	137
B.10.	PV/OP cross correlations for good performance: (a) PV/OP correlation for GOOD1S, (b) PV/OP correlation for GOOD2S, (c) PV/OP correlation for GOOD1R, (d) PV/OP correlation for GOOD2R	138
D.1.	Scores plots for stiction data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for STIC1S, (b) scores for STIC2S, (c) scores for STIC3S, (d) scores for STIC4S	161
D.2.	Scores plots for stiction data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for STIC5S, (b) scores for STIC6S, (c) scores for STIC1R, (d) scores for STIC2R	162
D.3.	Scores plots for stiction data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for STIC3R, (b) scores for STIC4R	163
D.4.	Scores plots for hysteresis data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for HYST1S, (b) scores for HYST2S, (c) scores for HYST3S, (d) scores for HYST4S	164
D.5.	Scores plots for hysteresis data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for HYST5S, (b) scores for HYST1R, (c) scores for HYST2R, (d) scores for HYST3R	165
D.6.	Scores plots for tuning data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for TUNE1S, (b) scores for TUNE2S, (c) scores for TUNE3S, (d) scores for TUNE4S	166
D.7.	Scores plots for tuning data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for TUNE5S, (b) scores for TUNE1R	167
D.8.	Scores plots for disturbance data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for DIST1S, (b) scores for DIST2S, (c) scores for DIST3S, (d) scores for DIST4S	168

- D.9. Scores plots for disturbance data projected onto PCA
models for good performance, with 95% confidence limits:
(a) scores for DIST5S, (b) scores for DIST6S,
(c) scores for DIST1R, (d) scores for DIST2R 169

CHAPTER 1

INTRODUCTION

Continuous industrial chemical processes are, for the most part, automatically monitored and controlled by the PID (proportional-integral-derivative) control algorithm. Although advanced control algorithms such as Model Predictive Control may be implemented on many processes, these algorithms are most often implemented at a "supervisory" level, with many PID control algorithms at a lower level maintaining basic process variables such as flow and pressure. A large continuous chemical process may contain several hundreds to thousands of PID control algorithms (each PID algorithm being part of an overall PID control loop). Each PID control loop may consist of several components: field sensor, control algorithm, control valve transducer, control valve positioner, and/or the control valve (or final control element). Any of these components can malfunction at any time due to a number of causes. If a control loop is performing poorly, the cause of the poor performance may be: a problem with one of the several control loop components (e.g., valve hysteresis, poor controller tuning, etc.), a problem in an associated control loop, a cycling process disturbance, malfunctioning process equipment, poor control strategy design, etc. The large number of PID control loops in a typical continuous process and the numerous potential causes of poor performance combine to form a very large monitoring and troubleshooting problem. The size of the problem makes it virtually impossible for operators, engineers, and managers to manually monitor

and troubleshoot PID control loop performance in a comprehensive manner.

The performance of PID control loops is critical to the overall operation of a continuous chemical process. Poorly-performing PID control loops significantly increase process variability and can even diminish the performance of advanced control strategies. The resulting increased variability often leads to excessive time spent by operations in troubleshooting and in manually controlling portions of the process. More importantly, significantly increased process variability prohibits the chemical process from running near equipment and/or product limits, which adversely affects the process economic performance and net profit (via an increase in the amount of reject product, reduced throughput, increased energy costs, etc.). Shinskey (1990) states that significant control loop oscillations result in increased energy consumption, waste of raw materials, and a potentially less uniform end product.

Process control experience by the author (at Eastman Chemical in Kingsport, Tennessee, USA) has shown that even simple PID flow/pressure control loops often perform poorly and inflate the process variability. Many of the most common performance problems result in oscillatory control loop behavior. Experience has also shown that a large percentage of PID control loop problems can often be fixed quickly and inexpensively, once the underlying problem is identified. Additionally, four of the most common flow/pressure control loop problems have been found to be one of the following:

control valve stiction, control valve hysteresis, poor controller tuning, or associated oscillating disturbances.

Because human monitoring of all the PID control loops in a large chemical process is infeasible, there are many PID control loop problems which go undetected. As a result, an automated way of detecting poorly-performing PID control loops (as well as data analysis methods to help determine the associated control loop problem) would be extremely beneficial to the operation of continuous chemical processes.

Because of current data acquisition, data analysis, and computer processing capabilities, it is viable to monitor the performance of many or all the control loops in a continuous chemical process. All that is needed is development of the proper real time monitoring algorithms for installation on plant control system computers.

A Control Loop Performance Monitor (CLPM) developed by Hagglund (1995) may be useful for identifying poorly-performing control loops via oscillation detection. Many common data analysis techniques (e.g., cross correlation function) and statistics (e.g., exponentially weighted moving variance) may be useful for diagnosing the most common problems leading to poor performance. In addition, Principal Components Analysis (PCA), a tool used in the science of chemometrics for data compression and information extraction, may be useful for problem diagnosis.

The objective of this work is twofold: 1) develop a monitoring algorithm to automatically detect PID flow/pressure control loops exhibiting poor performance, and 2) determine specific data analysis techniques/statistics which can aid in diagnosing four of the common problems causing poor performance (i.e., determine if the control loop problem is valve stiction or hysteresis, poor controller tuning, or an oscillating disturbance). To begin this work, multiple flow/pressure controller process data sets exhibiting the common performance problems will be obtained for use. A dynamic flowsheet simulator will be used to develop simulated process data sets, while real process data sets will be captured from various Eastman processes. With the process data sets available for testing, the monitoring algorithm will be developed using the CLPM as a starting point or basis. The monitoring algorithm will seek to identify poor performance via detection of sustained control loop oscillations. The developed monitoring algorithm should possess the following characteristics: the algorithm should be amenable to real time implementation on a computer control system, algorithm parameter selection should require minimal user input, and false performance alarms should be minimized. Additional analyses complementary to the monitoring algorithm may also be investigated for use. Following monitoring algorithm development, selected data analysis techniques (correlation analysis and PCA) and selected statistics will be applied to the simulated and real process data sets in hopes of finding useful techniques for diagnosing the common problems which cause poor flow/pressure control loop performance. Effort will be made at this stage to determine analyses and statistics which can aid in diagnosis without requiring visual

inspection of data, such that diagnosis application can be made in a quantitative manner. Effort will also be made to determine the need for data preprocessing prior to application of diagnosis techniques. The scope of this work will be limited to PID flow/pressure control loops with: 1) negligible dead time, 2) fast, first order responses, and 3) a performance objective of fast setpoint tracking with no steady-state offset.

CHAPTER 2

LITERATURE REVIEW

2.1 The Need for Controller Performance Monitoring/Problem Diagnosis

Kozub (1996) points out that maintenance of control strategies is one of the most frequently carried out activities by plant location engineers. Based on industrial experience at Eastman, troubleshooting of poorly-performing control loops is frequently carried out by process control engineers, process improvement engineers, control system technicians, and operators. The cause of the poor performance is often easy to remedy, but excessive time is spent before the cause is pinpointed. The poorly-performing loops often go unnoticed for months or years and may force non-optimal modes of operation. According to Ender (1993), an extensive testing of thousands of individual control loops in hundreds of operating plants revealed that more than 30% of the installed process control loops operate in manual. Also, approximately 30% of the loops have instrumentation problems (e.g., incorrectly sized valves; excessive hysteresis, resolution, or stiction in the valves; and measurement problems).

Much research has been carried out on the subject of controller performance monitoring. On a relative basis, less research has been concerned with use of data analysis methods for diagnosis of common problems leading to poor control loop performance. Recent research and traditional controller performance monitoring techniques consist

of a variety of performance criteria, performance indices, and data analysis methods for evaluating controller performance and, to a lesser extent, diagnosing potential controller problems. Many of the published controller performance monitoring techniques are related to measures of variance. The most widespread performance monitoring and problem diagnosis techniques are briefly discussed below.

2.2 Minimum Integral Error Criteria

Process control textbooks discuss the traditional performance indices of *IAE*, *ISE*, and *ITAE*, the integrated absolute error, integrated squared error, and integrated time-weighted absolute error, respectively:

$$IAE = \int_0^{\infty} |e(t)| dt$$

$$ISE = \int_0^{\infty} [e(t)]^2 dt$$

$$ITAE = \int_0^{\infty} t|e(t)| dt$$

where t is time, and e is the controller error (setpoint minus measurement). Control textbooks also focus on characteristics of closed loop performance such as settling time, overshoot, decay ratio, etc. The well-known Cohen-Coon controller design relations seek to produce a closed loop performance with a $\frac{1}{4}$ decay ratio, which is often too oscillatory, according to Seborg (1989), and considers only the first two peaks of the closed loop response. Minimization of the *IAE*, *ISE*, or *ITAE* indices generally promotes less oscillatory closed loop

performance (relative to the Cohen-Coon relations), but often results in tuning that is too aggressive for industrial control objectives. Shinskey (1990) suggests two related indices for measuring controller performance: the absolute performance index (API), which is the ratio of the best possible *IAE* attainable with a feedback controller to the actual *IAE* (in response to a step load change), and the robustness index (RI), which measures the proximity of the control loop to instability.

2.3 Minimum Variance Criteria

The theoretical "best achievable" control, as measured by the mean square error, is called minimum variance control. Harris (1989) states that if a univariate time series model is fit to closed loop process data, and if the number of whole periods of delay is known, the minimum variance performance can be calculated and used as a benchmark to judge the actual control loop performance. Stanfelj, Marlin, and MacGregor (1993) state that the minimum variance controller cancels the prediction of the disturbance after the dead time f , leaving only the uncorrelated prediction error to influence the controlled variable. To compute the minimum variance, the controller error (Y_t) is first fit with a moving average time series model of order f (Harris 1989):

$$Y_t = (1 + \psi_1 z^{-1} + \dots + \psi_f z^{-f}) a_t$$

where z^{-1} is the backward shift operator, a_t is the white noise driving force, f is the controller output/controlled variable sampling intervals of dead time, and $(\psi_1, \dots, \psi_f)$ are polynomial fit weights.

The minimum variance (σ_{mv}^2) is then given by:

$$\sigma_{mv}^2 = (1 + \psi_1^2 + \psi_2^2 + \dots + \psi_f^2) \sigma_a^2$$

where σ_a^2 is the variance of a_t .

Several published controller performance measures stem from the minimum variance method. Desborough and Harris (1992) define a normalized performance index ($\eta(b)$) as a means of assessing a controller's deviation from minimum variance control:

$$\eta(b) = 1 - \frac{\sigma_{mv}^2}{\sigma_y^2 + \mu_y^2}$$

where σ_{mv}^2 is the minimum variance, σ_y^2 is the actual closed loop variance, and μ_y^2 is the mean deviation from setpoint. When the variance inflation is zero, $\eta(b)$ is 0, indicating minimum variance control is being achieved. As $\eta(b)$ increases toward its maximum value of 1, control performance increasingly deteriorates. Desborough and Harris (1993) note that $\eta(b)$ can also be used as an on-line controller tuning tool.

Kozub (1993) finds useful a related measure noted as the Closed-Loop Potential (CLP):

$$CLP = \frac{\sigma_{mv}^2}{\sigma_e^2}$$

where σ_{mv}^2 is the minimum variance and σ_e^2 is the existing closed-loop variance. When CLP is close to 1, the best achievable feedback control is being applied.

Kozub (1996) describes variations on the CLP measure for the more common industrial cases where minimum variance control may not be practical or desirable. For these cases, a desired closed loop response specification (e.g., first order response, specific settling time, etc.) can be made to determine the "desired" variance. The CLP can then be computed using the actual closed loop variance and the "desired" variance (in place of the minimum variance). Kozub (1996) has found this CLP measure useful as a first-pass problem indicator.

Perrier and Roche (1992) document the use of a minimum variance related index (similar to the CLP) to study the regulatory performance of pressure, temperature, level, flow, and consistency control loops. Their performance index is used to predict the potential performance improvement through controller redesign or retuning.

The minimum variance performance criterion possesses several desirable characteristics related to industrial implementation. No process

perturbations are required; routine closed loop operating data can be used. The minimum variance computation directly provides a quantitative performance measure. If close to minimum variance control is being achieved, and performance is still unacceptable, plant personnel can be sure that the performance problem is not in the controller.

Although the minimum variance performance criterion does give a quantitative performance measure, the desired performance of most industrial controllers is sluggish compared to minimum variance. The majority of well-performing industrial loops have an actual variance far greater than the minimum variance. Another drawback of the minimum variance approach is the user time required to determine the dead time parameter f for a large number of loops. Overall, Kozub (1996) states that correlation methods are far more useful for studying controller performance because they give more insight into controlled variable response characteristics.

2.4 Statistical Variance Criteria

Statistics such as the Exponentially Weighted Moving Average (EWMA) and the Exponentially Weighted Moving Variance (EWMV) have been proposed for monitoring continuous process variation. They can be adapted for monitoring continuous chemical processes which contain highly autocorrelated data with changing means and variances.

MacGregor and Harris (1993) list the EWMA (Z_n) and EWMV (s_n^2) as:

$$Z_n = (1 - \lambda)Z_{n-1} + \lambda Y_n$$

$$s_n^2 = (1 - r)s_{n-1}^2 + r[Y_n - \eta_n]^2$$

where λ is the EWMA weight (between 0 and 1), Y_n is the actual process measurement, η_n is an estimate of the process mean at time n , and r is the EWMV weight (between 0 and 1), corresponding to the rate of exponential discounting of past data.

For the typical chemical process cases where the process mean is time varying, MacGregor and Harris (1993) recommend the EWMV for monitoring (with the EWMA used for calculation of η_n). The use of more than one EWMV monitor for each control loop is also recommended: an EWMV with a low value of r can be used to reach far back into the data and measure the longer term process variation, while a second EWMV with a higher value of r can monitor the local random component of the variance. Hunter, Harris, and MacGregor (1993) suggest potential variations of the EWMV by setting η_n equal to the previous process measurement (Y_{n-1}) or the setpoint value.

2.5 Controller Oscillation Detection

The control loop performance monitor (CLPM) introduced by Hagglund (1995) uses oscillation detection as a means of flagging poor control loop performance. The CLPM detects performance deterioration by sensing sustained loop oscillations. The CLPM monitors for

oscillation frequencies in the neighborhood of the ultimate frequency. The CLPM uses the familiar sine wave function as a calculation basis to approximate the oscillation of control loops.

In the CLPM algorithm, the control loop integrated absolute error (IAE) is monitored between consecutive zero crossings (t_{i-1} and t_i) of the controller error. The IAE is reset to zero upon each control error sign change:

$$IAE = \int_{t_{i-1}}^{t_i} e(t) dt$$

If the IAE value exceeds a specified IAE_{lim} value (between t_{i-1} and t_i), a load disturbance is detected. A maximum of 1 disturbance is detected for each control error sign change. The IAE_{lim} value is determined by:

$$IAE_{lim} = \int_0^{\frac{\pi}{\omega}} a \sin(\omega t) dt = \frac{2a}{\omega}$$

where $2a$ is the acceptable percent peak-to-peak oscillation (in percent of the measurement range) and ω is the loop ultimate frequency (known or estimated). If the number of load disturbances during a time window T_{sup} exceeds a specified limit n_{lim} , it is concluded that a sustained oscillation is present and that performance is unacceptable. The number of load disturbances is calculated at

each sample time via an exponential weighting based on T_{sup} and the CLPM sample period.

Hagglund (1995) gives several suggestions for utilizing the CLPM to monitor all the PID control loops in a given process, such that little user input is required. A reasonable choice of a is 1 %. Other suggestions are $n_{lim} = 10$, and $T_{sup} = 10 * n_{lim} * (T_u/2)$, where T_u is the ultimate period. T_u can be set equal to the integral time of the controller if the ultimate period is unknown.

The CLPM may be very useful for detecting a poorly-performing control loop, because many of the most common performance problems result in oscillatory loop behavior. The CLPM should be easy to apply due to its simplicity and minimal user input requirements. Study of the CLPM input parameter selection may be needed to ensure performance detection robustness and minimize false performance alarms.

2.6 Other Performance Monitoring Methods

Tyler and Morari (1996) promote the use of likelihood ratio tests to monitor control performance. Using closed loop data, the impulse response coefficients of the closed loop transfer function are identified. Desired performance criteria are translated into impulse response coefficient constraints. Likelihood ratio tests then reveal if the desired performance is being achieved.

Belanger and Luyben (1996) describe a LATV (load autotune variation) test for evaluating the regulatory performance of controlled processes. The LATV test involves the insertion of a relay from the controlled variable to a given load disturbance under automatic feedback control. This test causes the plant to exhibit a sustained oscillation at the worst case frequency, allowing the plant regulatory performance to be evaluated.

2.7 Correlation Analysis

The autocorrelation and cross correlation functions estimate the linear correlation of a given signal or the linear correlation between different signals, respectively, at different sample lags for discrete data sequences. At a given lag k , a correlation value close to 1 (in absolute value) indicates high correlation between signals; a value close to 0 indicates negligible correlation. The estimate of the autocorrelations ($r_{yy}(k)$) for signal Y_t (at lag k) is listed by MacGregor and Taylor (1994) as:

$$r_{yy}(k) = \frac{c_{yy}(k)}{c_{yy}(0)}$$

with

$$c_{yy}(k) = \frac{1}{N} \sum_{t=1}^{N-k} (Y_t - \bar{Y})(Y_{t+k} - \bar{Y}) \quad k = 0, 1, 2, \dots$$

Similarly, the estimate of the cross correlations between signals u_t and Y_t ($r_{uy}(k)$) is:

$$r_{uy}(k) = \frac{c_{uy}(k)}{\sqrt{c_{uu}(0) \cdot c_{yy}(0)}}$$

with

$$c_{uy}(k) = \frac{1}{N} \sum_{t=1}^{N-k} (u_t - \bar{u}) (Y_{t+k} - \bar{Y}) \quad k = 0, 1, 2, \dots$$

For both functions, there are N observations of the time series $(Y_1, Y_2, \dots, Y_N)$ and/or $(u_1, u_2, \dots, u_N)$ of interest, with $\bar{Y}$ and $\bar{u}$ representing the sample means of the respective time series.

These correlation functions are useful for determining cause/effect relationships and for illustrating the nature of control loop performance. Cycling correlations normally indicate oscillations in controlled variables or associated signals. A cycling autocorrelation could indicate an overtuned controller, while a slowly decaying autocorrelation might indicate a sluggish controller.

Hunter, Harris, and MacGregor (1993) view the autocorrelation function as a valuable tool for understanding the cause of process variation and for assessing control loop performance. Large autocorrelations beyond the controller output/controlled variable dead time lag indicate significant deviation from minimum variance performance. Kozub (1996) states that autocorrelation analysis can simplify the analysis of complicated controlled variable trends and can be useful for separating controller and process problems.

Cross correlation analysis can be used as a test to determine if measurable loop disturbances and setpoint changes have a significant association with the controlled variable error, as well as determine which of several measurable disturbances is having the largest influence on the control loop performance. For most well-performing flow/pressure control loops, the cross correlations between the controlled variable and the controller output should be high at small lag values. For purposes of diagnosing the problems causing poor loop performance, cross correlation analysis may be useful, as different problems may be manifested to some extent as different correlation characteristics.

Hackney (1989) describes a Time-Shifted Distribution (TSD) measure which appears to be related to the cross correlation function. The TSD measure calculates the percentages of controller error versus controller delta output data points which lie in the various four quadrants (with the axes origin at controller error and controller output equal to 0), at different sample time lags. It appears that poor performance due to a controller problem may result in a large percentage of TSD data points lying in quadrants opposite that of data resulting from a process disturbance problem. Thus, the TSD measure may be particularly useful in diagnosing problems causing poor control loop performance.

Simple "y versus x" type plots, such as that of controlled variable versus the controller output, may also be useful for identifying causes of poor control performance. Common problems such as valve

hysteresis, poor tuning, and oscillating disturbances may result in distinct "y versus x" type plots.

Care should be taken when utilizing correlation analysis.

Preconditioning the loop data (data differencing, subtracting the mean of the data, etc.) may be needed to prevent misleading correlations due to shifting mean values and other time variant process characteristics.

2.8 Spectral Analysis

Krauss, Shure, and Little (1994) define spectral analysis as an analysis which describes the frequency content of a signal based on a finite set of data. Spectral analysis provides an analysis of variance with respect to frequency, with the area under the spectrum curve between any two frequencies representing the respective variance contribution or signal power. The sample spectrum $S_{yy}(f)$ for a signal Y_t , obtained with application of a discrete time Fourier transform, is defined as (MacGregor and Taylor 1994):

$$S_{yy}(f) = \frac{1}{N} \sum_{K=-(n-1)}^{n-1} c_{yy}(k) e^{-i2\pi f k}$$

with

$$c_{yy}(k) = \frac{1}{N} \sum_{t=1}^{N-k} (Y_t - \bar{Y}) (Y_{t+k} - \bar{Y}) \quad k = 0, 1, 2, \dots$$

Similarly, the sample cross spectrum between two signals Y_t and u_t is defined as (MacGregor and Taylor 1994):

$$S_{uy}(f) = \sum_{k=-(n-1)}^{n-1} c_{uy}(k) e^{-i2\pi f k}$$

with

$$c_{uy}(k) = \frac{1}{N} \sum_{t=1}^{N-k} (u_t - \bar{u}) (Y_{t+k} - \bar{Y}) \quad k = 0, 1, 2, \dots$$

For both the sample spectrum ($S_{yy}(f)$) and sample cross spectrum ($S_{uy}(f)$) functions, there are N observations of the time series ($Y_1, Y_2, \dots, Y_N$) and/or ($u_1, u_2, \dots, u_N$) of interest, with $\bar{Y}$ and $\bar{u}$ representing the sample means of the respective time series. The frequency f ($-\frac{1}{2} \leq f \leq \frac{1}{2}$) is $\frac{1}{2}$ the Nyquist frequency.

According to Desborough and Harris (1992), spectral analysis can be useful for determining specific problems underlying poor controller performance. A large spectral peak at low frequencies may indicate an undertuned controller allowing excessive process drift, while a large peak at high frequencies may indicate an overtuned controller causing excessive cycling.

2.9 Principal Components Analysis (PCA)

PCA is a chemometrics tool which has wide-ranging application in process monitoring and fault detection. Simply put, PCA is used for analyzing the variability in a set of data. PCA describes data variation by determining the dominant directions in which the data varies, i.e., by finding the principal components or latent vectors

which describe the greatest amount of information about the data set in the smallest dimensional space. According to Miller (1994), for a data matrix $\mathbf{X}$ composed of k samples (the rows of the matrix) and n variables (the columns of the matrix), the PCA decomposition into the sum of n pairs of vectors is:

$$\mathbf{X} = \sum_{i=1}^n \mathbf{t}_i \mathbf{p}_i^T$$

Each of the n vector pairs is made up of a vector $\mathbf{p}_i$ of length n , called the *loadings*, and a vector $\mathbf{t}_i$ of length k called the *scores*. The loadings contain information on how the variables relate to each other, while the scores contain information on how the samples relate to each other.

Associated with PCA are Q and T^2 statistics. Wise and Gallagher (1996) describe the Q statistic as indicating how well each sample conforms to the PCA model, as Q measures the difference, or residual between a sample and its projection into the principal components retained in a given PCA model (a large Q indicates unusual variation outside the model). They describe the T^2 statistic as a measure of the variation in each sample within the PCA model (a large T^2 indicates unusual variation within the model). Confidence limits can be calculated for the Q and T^2 statistics and violations of the confidence limits can help in identifying significant process changes or disturbances; this is illustrated by Miller (1994) for a reactive distillation column.

PCA may be very useful in diagnosing the different problems which cause poor control loop performance. For instance, the loading values or percent variance descriptions associated with the PCA latent vectors may change significantly depending on the specific control loop problem. Wise and Gallagher (1996) describe how PCA can be applied to lagged input/output variable matrices to detect changes in the input/output relationships of dynamic systems. Kresta, Marlin, and MacGregor (1991) demonstrate techniques related to PCA which use scores plots to detect significant process changes.

CHAPTER 3

DESCRIPTION OF COMMON CONTROLLER PROBLEMS AND PROCESS DATA SET DEVELOPMENT

3.1 Introduction to Flow/Pressure Controller Performance

Flow and pressure control loops with negligible dead time (as well as fast, first order responses) and a performance objective of fast setpoint tracking with no steady-state offset will be considered in the scope of the work to follow. Basic flow/pressure control loops often comprise a large percentage of all the PID loops in a given chemical process. Although this type of control loop is relatively simple, satisfactory flow/pressure control loop performance is critical for meeting the overall economic and product quality requirements for a given chemical process. The overall success of supervisory control strategies will often depend in part on the performance of multiple flow/pressure control loops.

3.2 Description of Controller Configuration and Conventions

The typical control loop configuration and conventions which will be considered in the work to follow is shown for an example flow controller in Figure 3.1. The configuration consists of several control loop components: 1) a measurement element (FE) for sensing the process flow, 2) a transmitter (FT) for transmitting the

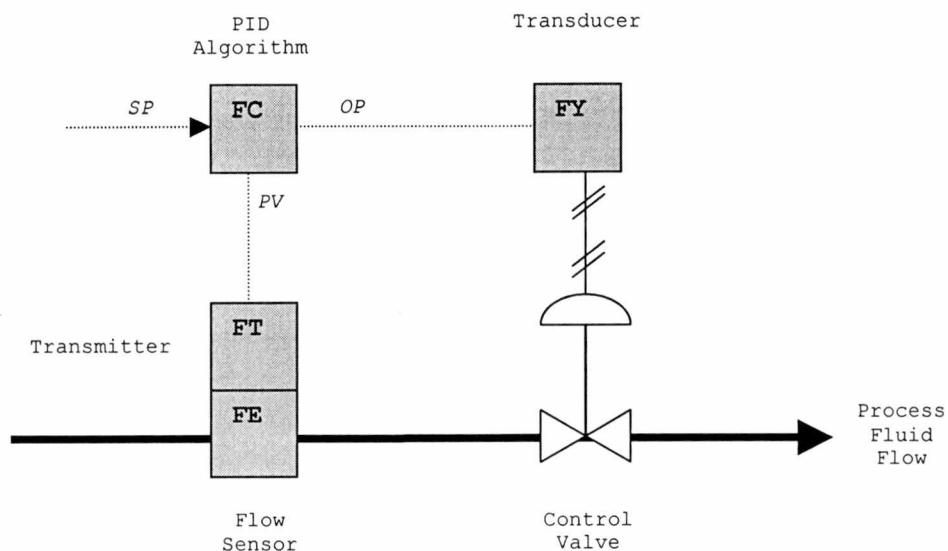


Figure 3.1. Example configuration for flow control loop.

measurement value, 3) the PID control algorithm (FC) which typically resides in a Distributed Control System (DCS) and computes a controller output (OP) based on the incoming measurement or process value (PV), controller setpoint (SP), and PID tuning settings, 4) a transducer (FY) which translates the controller output signal to the control valve positioner or actuator, 5) a control valve positioner (may not be present for all control loops), and 6) a control valve. This control loop configuration is typical for the majority of basic flow/pressure control loops, as control valve movement affects the process flow/pressure directly and instantly. Filtering of the process value (PV) can be applied with the transmitter or as part of the PID algorithm. All of the process data for a PID flow control loop is contained in 3 variables: the controller setpoint (SP), controller process value (PV), and controller output (OP). In the work to follow, the collection of these 3 variables will be used as

the process data set for a specific control loop and will be used extensively in the subsequent monitoring algorithm and diagnosis technique development.

3.3 Description of Common Controller Problems

Four problems which often cause poor flow/pressure controller performance are control valve stiction, control valve hysteresis, poor controller tuning, or associated oscillating disturbances. While any of the several control loop components mentioned previously can malfunction at times and result in diminished controller performance, the four problems listed are some of the most common found in industrial experience at Eastman. Although each problem has distinct origins and characteristics, the problems are not necessarily easily identified in the midst of measurement noise, controller setpoint changes, and varying process disturbances. Each of the four problems is discussed below.

Control valve stiction is a common problem which occurs when the control valve stem "sticks" in place as the controller output changes. As the controller output continues to increase in one direction, the valve remains stationary. As the output continues to increase, a point is reached where the resistance to valve movement is overcome and the valve moves to the new position signaled for by the controller output. Often, control valve stiction results in a valve position alternating between two discrete points: an increased valve opening (e.g., controller PV significantly above setpoint) and a decreased

valve opening (e.g., controller PV significantly below setpoint). Due to the stiction problem the valve often can not be positioned at the opening needed to attain the setpoint, resulting in a square wave type oscillation around the controller setpoint.

According to Liptak (1985), most valve actuators display this valve stem dead band or stiction due to friction present in the valve packing. The valve packing may often be overtightened to prevent process fluid leakage. Properly working control valve positioners can often help overcome the problems of valve stiction or lessen the degree of stiction, but the use of a positioner requires the control loop to be slightly detuned due to the extra delay introduced by the positioner element.

Control valve hysteresis is a common problem which is defined as the difference in valve stroke between an increasing and decreasing valve signal expressed in percent of valve stroke (McMillan 1983).

Hysteresis is an indicator of the change in valve signal required to change the direction of the stroke or to start the stroke from a stationary position. Liptak (1998) describes hysteresis as dead play or dead band in the valve mechanical linkages; when the control signal reverses, the valve stem does not start moving in the opposite direction until the dead band is exceeded. As such, as the degree of hysteresis increases, there will be decreased control performance, with increased instances of the controller output changing but the valve position remaining fixed. This dead area of valve operation where the valve does not move with controller output changes leads to

significant oscillations and excursions of the controller PV away from setpoint. Hysteresis is illustrated in Figure 3.2. Control valve hysteresis is somewhat different from valve stiction. Once the hysteresis dead band is overcome, the valve moves incrementally from its current position as directed by the controller output (until the controller output reverses direction), resulting in a smoothly responding controller PV. With valve stiction, the valve jumps from its current position to the significantly different valve position called for by the controller output, resulting in a sudden step change in controller PV. Liptak (1998) asserts that hysteresis is often found in rotary control valves, where a linear valve actuator motion is converted to rotation. The linear to rotational conversion requires mechanical linkages which often introduce friction and dead

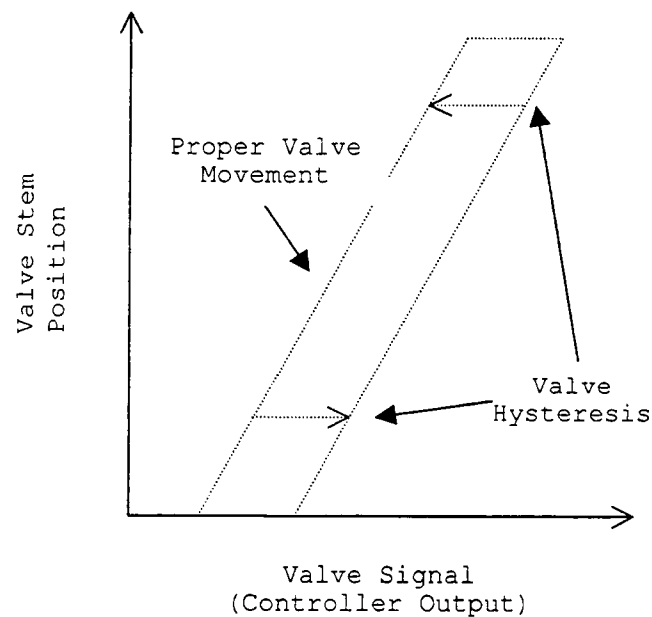


Figure 3.2. Hysteresis illustration.

play. Properly working control valve positioners can often help overcome the problems of hysteresis or at least lessen the degree of hysteresis.

Poor controller tuning is a problem where the PID tuning parameters (proportional gain, integral, and derivative) are set incorrectly, leading to significant oscillations around the setpoint. For the typical flow/pressure control loop, correct tuning parameter settings are as follows: a low gain is used to reduce the effect of noise in the measurement (controller PV signal), a low value of integral is used to get fast setpoint tracking, and derivative is not used (Luyben 1990). Examples of tuning parameter settings which can cause poor control loop performance are: 1) the gain value is too high causing excessive amplification of measurement noise, and 2) the integral value is set too low (fast reset), causing the loop to become oscillatory or unstable. Poor tuning problems can also occur where the controller PV is filtered too excessively in an attempt to reduce noise, as low integral values can attempt to make the loop respond more quickly than the filter setting allows.

Oscillating or cyclic process disturbances are problems outside the components of the control loop which cause poor control loop performance. For these cases, the performance issue begins as the disturbance causes the controller PV to move away from setpoint. As the controller responds and moves the control valve to compensate, the disturbance quickly disappears or reverses its effect on the controller PV, resulting in the PV quickly moving in the opposite

direction and the controller error changing sign. Although many disturbances are typically present in a continuous chemical process, many are transient in nature or of low frequency or of insufficient magnitude to significantly affect flow/pressure control loop performance. Other oscillating disturbances, however, can cause a significant control loop oscillation even for loops which are well tuned and have accurate controller output to valve movement relationships. A few examples of process disturbances which could deteriorate the performance of a given control loop are: cycling process pressures upstream or downstream of the loop of interest, other oscillating control loops which have a direct relation to the loop of interest, malfunctioning process equipment, etc.

3.4 Development of Simulated Process Data Sets

For developing simulated process data sets exhibiting the four common control loop problems, Eastman's Dynamic Plant (DP) dynamic flowsheet simulation package was utilized. Within DP, a simple flowsheet was developed consisting of a tank with one liquid inlet and one liquid outlet. A PID flow control loop was configured for both the inlet and outlet streams. To run the dynamic simulation, a level controller was configured to automatically adjust the flow setpoint for the inlet or outlet PID flow controller, with the setpoint for the other flow controller set manually. The control loop problem of interest was seeded in one of the PID flow control loops using custom algorithms within the DP simulator. The DP simulator was then run for a nominal length of time (typically between 0.5 and 1.5 hours) to gather

sufficient process data for the problem of interest (i.e., the simulation was performed long enough to capture multiple oscillations around the setpoint). The DP simulations were executed at a frequency of 0.25 seconds, with the process data sampled every 1.0 seconds. To help the simulated data mimic real behavior as closely as possible, first order time lags were added to the process measurement (1.0 seconds) and valve movement (4.0 seconds). Measurement noise was also added. Many simulations for each of the four common problems were performed with both constant and varying controller setpoints. From the multiple data sets, several data sets (felt to be the most accurate representations of each of the common problems) were selected for use in the scope of this work.

To generate simulated process data sets for a valve stiction problem, a DP simulator algorithm was used which set the actual flow controller output to its previous value until the change in the calculated controller output exceeded a user specified limit value (this limit corresponded to the amount of valve stiction or dead band present in the valve). When the limit was exceeded, the actual flow controller output was changed to the calculated controller output. This allowed the capture of process data showing the characteristic behavior of a valve stiction problem.

To generate simulated process data sets for a valve hysteresis problem, a DP simulator algorithm was used which allows user specified Reverse Polish Notation (RPN) type statements to be executed each sample time. Using the RPN statements, the characteristic hysteresis

relations between controller output and valve position were programmed for one of the PID flow control loops in the DP simulator. A variable was configured in the simulator for specifying the percent hysteresis to allow different magnitudes of hysteresis to be studied. Use of the DP simulator in this way allowed the capture of process data showing the characteristic behavior of a valve hysteresis problem.

To generate simulated process data sets for a tuning problem, the gain and integral tuning parameters for the flow control loop were varied until sustained oscillations or significant variations around the setpoint were exhibited. In addition, simulations were made with overaggressive gain and integral tuning parameters for cases where significant filtering of the controller PV or significant loop delays were present. Use of the DP simulator in this way allowed the capture of process data showing the characteristic behavior of a tuning problem.

To generate simulated process data sets for a process disturbance problem, a centrifugal pump unit was configured for the tank liquid outlet stream in the DP simulator. This allowed the pump discharge pressure (upstream of the flow controller of interest) to be cycled in a sine wave or square wave fashion at a user specified frequency and magnitude, simulating a cycling inlet pressure disturbance for the flow controller. Simulations were made with the discharge pressure cycling at different frequencies and magnitudes. Use of the DP simulator in this way allowed the capture of process data

demonstrating the characteristics of a process disturbance affecting control loop performance.

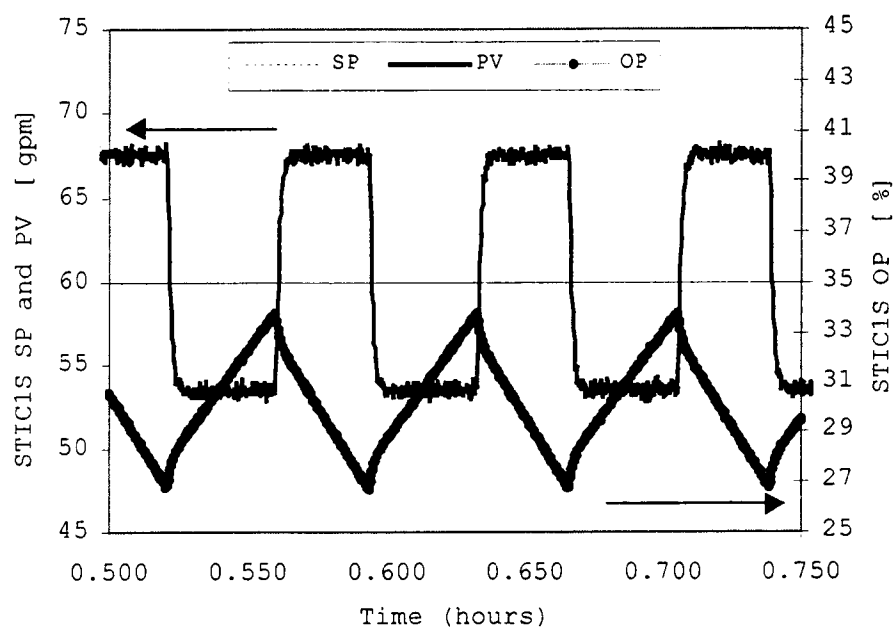
In addition to the simulations described above for the control loop problems, DP simulations were also utilized to obtain process data exhibiting good flow control performance. Tuning parameter values and flowsheet configurations used were similar to those used in the simulation of the 4 common controller problems.

From the multiple data sets generated via the simulations, five to six data sets for each controller problem and two data sets for good performance were selected for use in developing problem diagnosis techniques (Table 3.1). Note in Table 3.1 that each data set has a unique ID as follows: four letters (problem), a sequential number, and an S (for "simulated"). As an example, SP, PV, and OP data for a control valve stiction problem, along with the corresponding PV/OP characteristic plot, are shown (Figure 3.3). Another example is shown for a controller tuning problem (Figure 3.4). Analogous plots for all of the data sets listed in Table 3.1 are shown in Figures A.1-A.13. Note: subsets of the entire process data sets are shown for clarity and to visually demonstrate the different controller PV response characteristics and PV/OP relations for the individual problems.

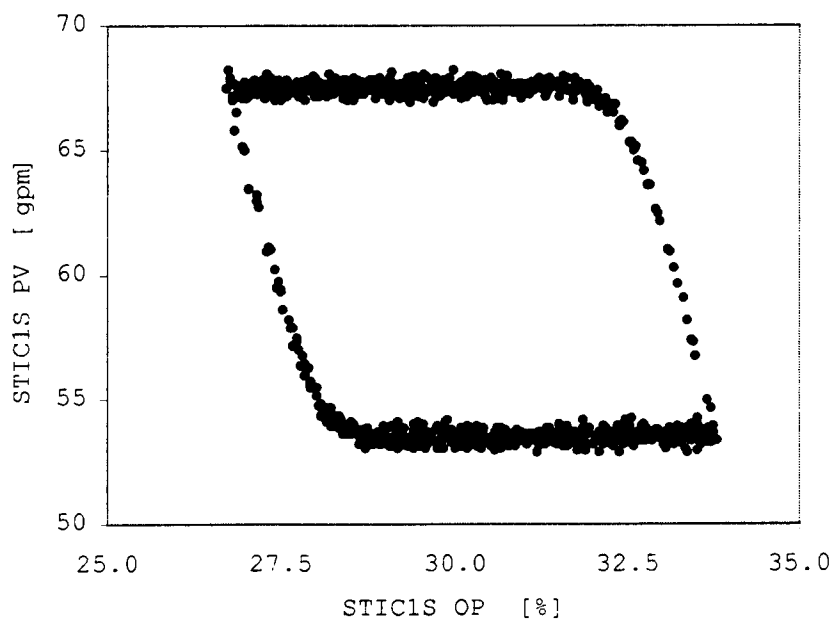
The simulated process data sets described previously were obtained for use in developing controller problem diagnosis techniques. Because of the abundance of real process data exhibiting significant

Table 3.1. Simulated process data sets to be used in development of controller problem diagnosis techniques.

Sim. Loop ID	Problem	Control Loop Details
STIC1S	Stiction	Stiction = 7%, total simulation time = 1.0 hr, controller gain = 0.25
STIC2S	Stiction	Stiction = 7%, total simulation time = 1.0 hr, controller gain = 0.50
STIC3S	Stiction	Stiction = 12%, total simulation time = 1.0 hr, controller gain = 0.25
STIC4S	Stiction	Stiction = 10% (dec. output)/5% (inc. output), total simulation time = 1.0 hr, controller gain = 0.25
STIC5S	Stiction	Stiction = 5%, total simulation time = 0.6 hr, controller gain = 0.25, wide range of SPs
STIC6S	Stiction	Stiction = 0.5%, total simulation time = 1.0 hr, controller gain = 0.25
HYST1S	Hysteresis	Hysteresis = 8%, total simulation time = 1.0 hr, controller gain = 0.25, varying SP
HYST2S	Hysteresis	Hysteresis = 8%, total simulation time = 1.0 hr, controller gain = 0.50, varying SP
HYST3S	Hysteresis	Hysteresis = 8%, total simulation time = 1.5 hr, controller gain = 0.25, varying SP
HYST4S	Hysteresis	Hysteresis = 4%, total simulation time = 1.0 hr, controller gain = 0.25, varying SP
HYST5S	Hysteresis	Hysteresis = 1%, total simulation time = 1.0 hr, controller gain = 0.25, varying SP
TUNE1S	Tuning	High gain/low integral, total simulation time = 0.5 hr, controller gain = 4.25
TUNE2S	Tuning	Low integral, total simulation time = 0.5 hr, controller gain = 0.40
TUNE3S	Tuning	Excessive measurement noise/high gain, total simulation time = 0.5 hr, gain = 9.50
TUNE4S	Tuning	Excessive PV filter, total simulation time = 0.5 hr, controller gain = 0.50
TUNE5S	Tuning	Dead time on valve movement, total simulation time = 0.5 hr, controller gain = 0.75
DIST1S	Disturbance	Low frequency sine wave, total simulation time = 1.0 hr, controller gain = 0.25
DIST2S	Disturbance	High frequency sine wave, total simulation time = 0.5 hr, controller gain = 0.25
DIST3S	Disturbance	High frequency sine wave, total simulation time = 1.0 hr, controller gain = 0.50
DIST4S	Disturbance	Mid. frequency sine wave, total simulation time = 0.5 hr, controller gain = 0.25
DIST5S	Disturbance	Mid. frequency square wave, total simulation time = 1.0 hr, controller gain = 0.25
DIST6S	Disturbance	High frequency square wave, total simulation time = 0.8 hr, controller gain = 0.25
GOOD1S	None (good)	Total simulation time = 1.0 hr, gain = 0.25
GOOD2S	None (good)	Total simulation time = 1.0 hr, gain = 0.50

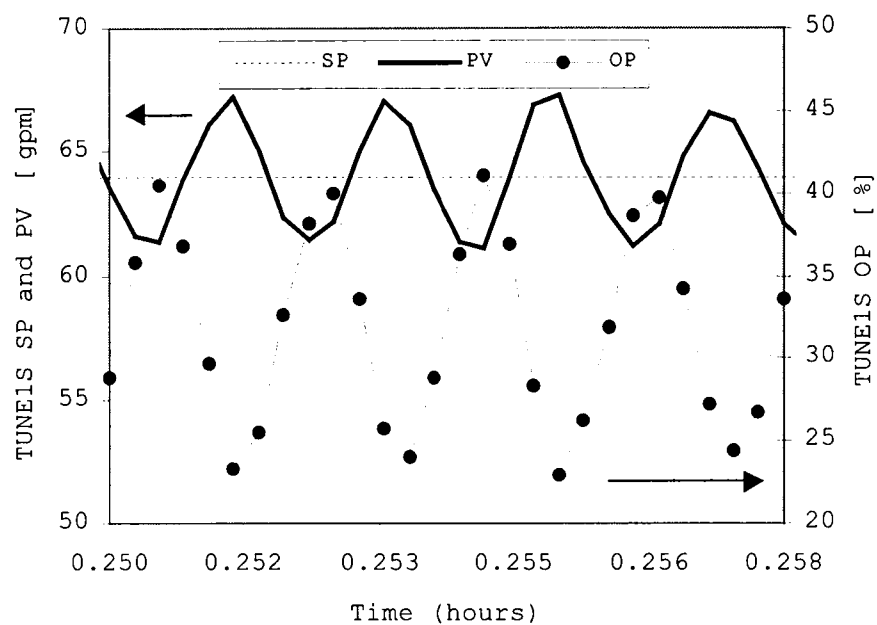


(a)

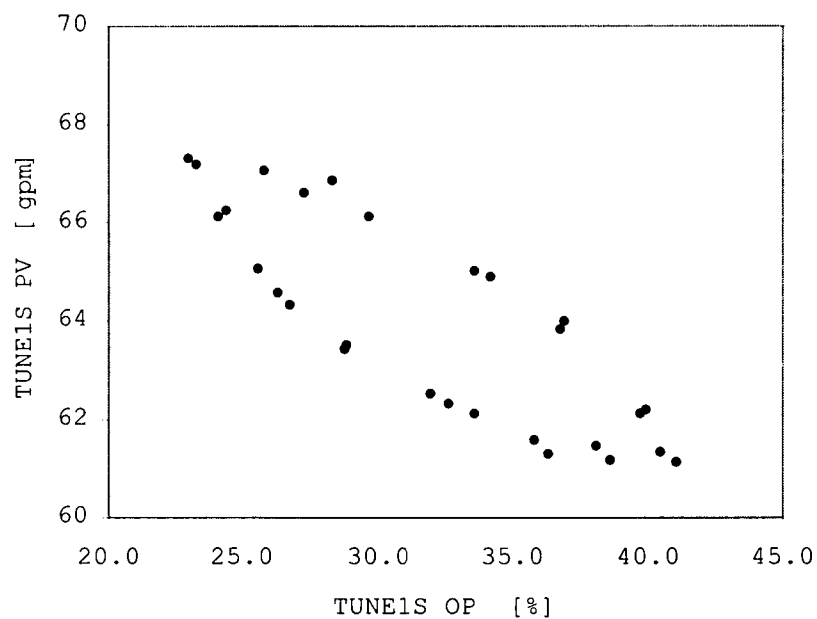


(b)

Figure 3.3. Simulated data for valve stiction problem (STIC1S):
 (a) SP, PV, and OP data, (b) PV/OP characteristic plot.



(a)



(b)

Figure 3.4. Simulated data for poor tuning problem (TUNE1S):
 (a) SP, PV, and OP data, (b) PV/OP characteristic plot.

oscillations, no simulated data were needed for use in developing the monitoring algorithm (for detecting loops with poor performance).

3.5 Selection of Real Process Data Sets

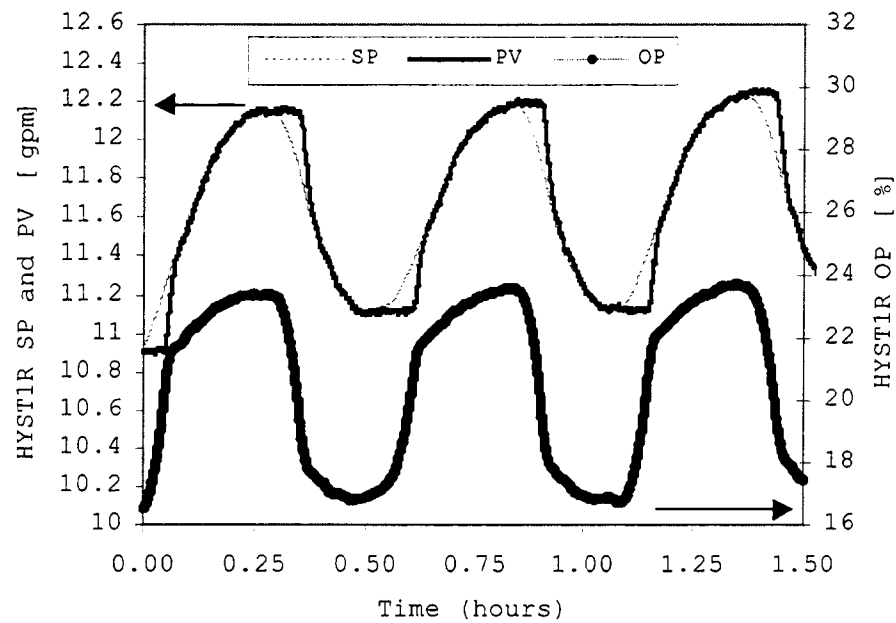
Eastman's Apex process monitoring package (a Fortran application which runs on DEC Vax/Alpha computers attached to plant Distributed Control Systems) was used to view flow/pressure control loop process data from various Eastman processes for the purposes of finding loops with the 4 common performance problems. Controller SP, PV, and OP data were trended (viewed) using Apex for a large number of flow/pressure control loops from various processes. Visual inspection of the process data was used to identify specific control loops which were performing poorly due to one of the four common problems of interest, as well as loops which were performing well (for use as a reference). In some cases, manual plant testing of specific loops was performed to determine if the controller problem was poor tuning or a process disturbance outside the control loop. Once a specific control loop was identified, Apex was used to collect and write (on a 5.0 second sample frequency) controller SP, PV, and OP data to a user file suitable for export to other software packages for analysis. In addition, a custom Fortran program was written and used to collect data at a higher sample frequency (1.5 to 2.0 seconds) for specific loops. For each control loop of interest, sufficient data was collected to capture multiple oscillations around the controller SP (e.g., most data sets collected were between 1.0 hours and 3.0 hours in length).

From the multiple real process data sets generated by the manual loop monitoring and visual inspection, one to three data sets for each controller problem and two data sets for good performance were selected for use in developing problem diagnosis techniques (Table 3.2). Note in Table 3.2 that each data set has a unique ID as follows: four letters (problem), a sequential number, and an R (for "real"). As an example, SP, PV, and OP data for a control valve hysteresis problem, along with the corresponding PV/OP characteristic plot, are shown (Figure 3.5). Another example is shown for a process disturbance problem (Figure 3.6). Analogous plots for all of the data sets listed in Table 3.2 are shown in Figures A.14-A.20. Note: subsets of the entire process data sets are shown for clarity and to visually demonstrate the nature of the different controller PV response characteristics and PV / OP relations for the individual problems. For reference, a subset displaying good flow control performance is also shown (Figure 3.7).

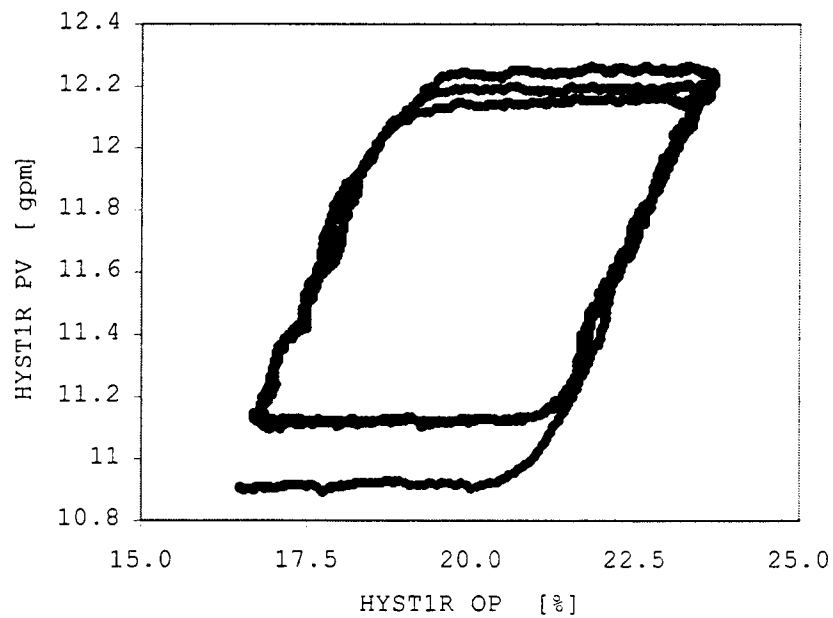
The Apex monitoring software was used in an analogous fashion to identify real control loops which would be good candidates for use in the testing and development of the performance monitoring algorithm. Effort was made to find loops which would yield adequate data for purposes of testing and development. The selected control loop data needed to be over a wide range of operating and performance conditions to provide information concerning the sensitivity and robustness of the monitoring algorithm. Thus, the specific control loops selected (Table 3.3) contain a variety of characteristics: good performance,

Table 3.2. Real process data sets to be used in development of controller problem diagnosis techniques.

Plant		
Loop ID	Problem	Control Loop Details
STIC1R	Stiction	Liquid flow, sample time = 1.46 seconds, total time for data set = 3.0 hr
STIC2R	Stiction	Acetic acid gas flow, sample time = 2.12 seconds, total time for data set = 1.0 hr
STIC3R	Stiction	Liquid flow, sample time = 5.00 seconds, total time for data set = 5.0 hr
STIC4R	Stiction	Acetic acid gas flow, sample time = 2.57 seconds, total time for data set = 1.0 hr
HYST1R	Hysteresis	Liquid flow, sample time = 1.57 seconds, total time for data set = 3.5 hr
HYST2R	Hysteresis	Liquid flow, sample time = 1.47 seconds, total time for data set = 3.0 hr
HYST3R	Hysteresis	Liquid flow, sample time = 5.00 seconds, total time for data set = 4.5 hr
TUNE1R	Tuning	Natural gas pressure, sample time = 1.92 seconds, total time for data set = 1.0 hr
DIST1R	Disturbance	Natural gas pressure, sample time = 1.90 seconds, total time for data set = 3.0 hr
DIST2R	Disturbance	Acetic acid gas flow, sample time = 2.59 seconds, total time for data set = 1.0 hr
GOOD1R	None (good)	MeOH liquid flow, sample time = 1.41 seconds, total time for data set = 3.0 hr
GOOD2R	None (good)	Natural gas flow, sample time = 2.06 seconds, total time for data set = 4.0 hr

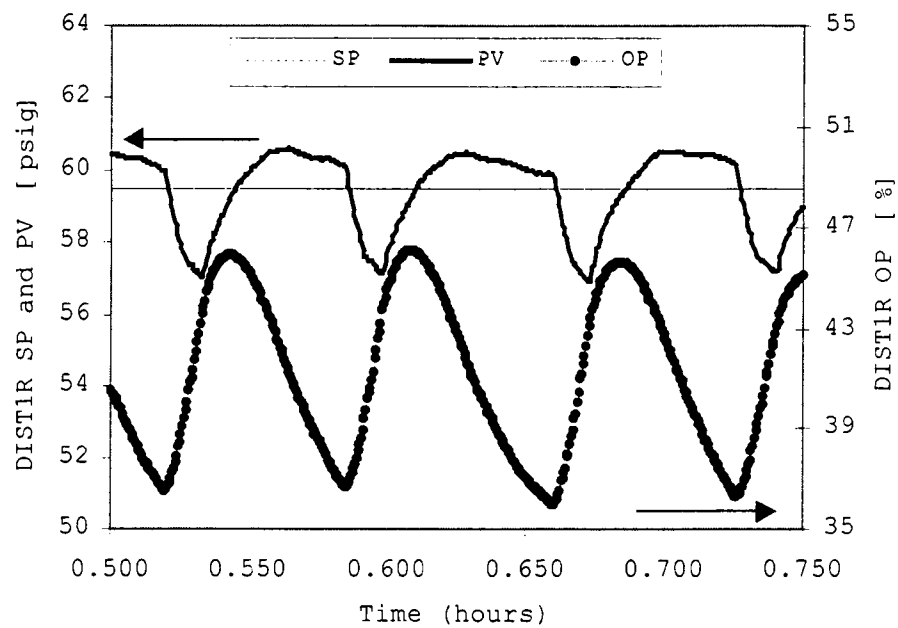


(a)

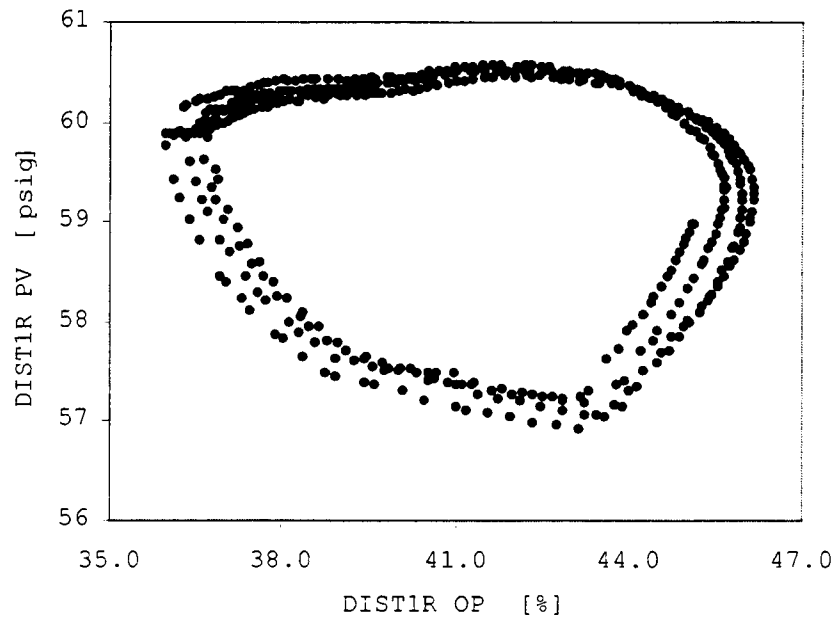


(b)

Figure 3.5. Real data for valve hysteresis problem (HYST1R):
(a) SP, PV, and OP data, (b) PV/OP characteristic plot.

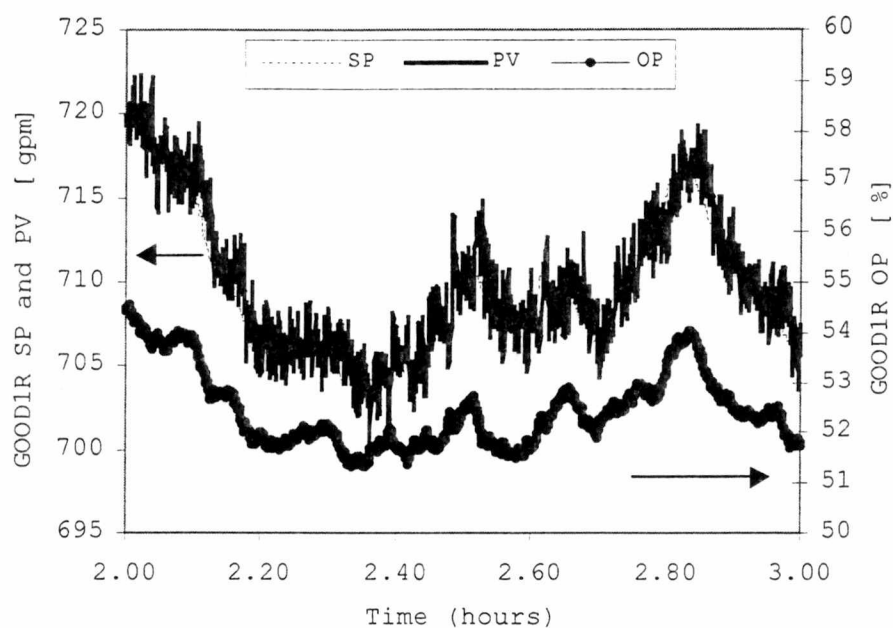


(a)

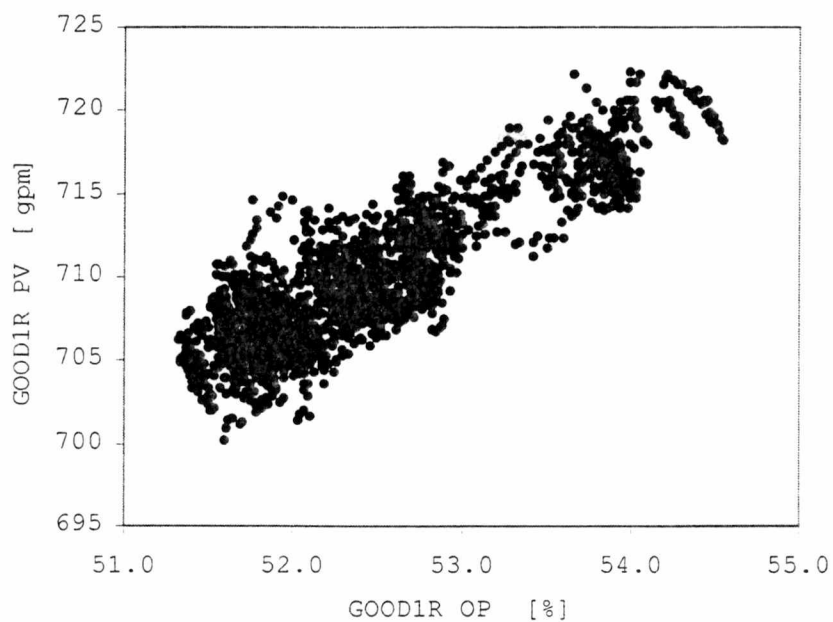


(b)

Figure 3.6. Real data for process disturbance problem (DIST1R): (a) SP, PV, and OP data, (b) PV/OP characteristic plot.



(a)



(b)

Figure 3.7. Real data for controller with good performance (GOOD1R): (a) SP, PV, and OP data, (b) PV/OP characteristic plot.

Table 3.3. Real process data sets to be used in development of performance monitoring algorithm.

Plant	
Loop ID	Performance Details
STIC3R	Poor performance due to valve stiction, low frequency oscillation, sample time = 5.0 seconds
TUNE1R	Poor performance due to poor tuning, high frequency oscillation, sample time = 1.92 seconds
DIST1R	Poor performance due to disturbance, sample time = 1.90 seconds
HYST4R	Poor performance due to valve hysteresis, sample time = 5.0 seconds
GOOD2R	Good performance (slight oscillations due to disturbance), sample time = 2.06 seconds

poor performance, marginal performance, low frequency oscillations, high frequency oscillations, etc. Development and testing of the performance monitoring algorithm (using the selected data) will be addressed in the next chapter.

CHAPTER 4

PERFORMANCE MONITORING ALGORITHM TESTING AND DEVELOPMENT

4.1 Desired Characteristics for the Monitoring Algorithm

Before actual testing and development of the proposed monitoring algorithm can begin, the desired characteristics of the monitoring algorithm need to be listed. The goal is to develop a monitoring algorithm which will reliably and robustly detect poorly-performing flow/pressure control loops present in continuous chemical processes, given a set of user specified criteria. Thus, the successful monitoring algorithm will possess the following characteristics: 1) detection of problems in any oscillation frequency range, 2) the required user input is minimal and applicable to most or all of the loops of interest (i.e., unique user input not required for each individual control loop), 3) capable of real time implementation for continuous chemical processes, and 4) false performance alarms are minimized while maintaining alarming sensitivity. In addition, the alarming ability of the monitoring algorithm should be independent of the problem causing the poor control performance; if the user specified performance limits are being violated, the monitoring algorithm should alarm regardless of the problem source.

4.2 Initial Testing of the Hagglund CLPM

Because the Hagglund CLPM is the basis or starting point for developing the performance monitoring algorithm in this work, initial testing of the Hagglund CLPM is needed to determine its strengths and limitations. While the Hagglund CLPM is specifically designed for detecting oscillations in the proximity of the ultimate frequency of the controller (relatively high frequencies for the basic flow/pressure control loops considered in this work), the intention in developing the performance monitoring algorithm is to develop an algorithm which will detect poor performance at any frequency.

The CLPM will first be tested on real process data for an oscillating flow control loop (STIC3R). The STIC3R loop has a half period oscillation frequency of 5-6 minutes and the average peak error for each oscillation is roughly 1.5 % of the STIC3R PV range (0-315 lb/min). The suggested default user input values (Hagglund 1995) will be utilized in the CLPM algorithm. Using the default values gives: $a = 1.0$ % (the limit in percent of PV range for counting as an oscillation), $n_{lim} = 10$ (the alarm limit for the number of oscillations), $T_u = 30$ seconds (the actual integral time of the STIC3R controller), and $T_{sup} = 10 * n_{lim} * (T_u / 2)$ (the moving time window period). Given the Hagglund CLPM algorithm and the specified inputs, the CLPM was tested by programming an m-file (CLPM_HAGG.M, Appendix C) in Matlab to perform the CLPM calculations on the STIC3R process data. For this test (Figure 4.1), the CLPM failed to alarm despite the peak oscillations for STIC3R significantly exceeding the user specified limit (1.5 % versus limit of 1.0 %). The CLPM oscillation rate lines

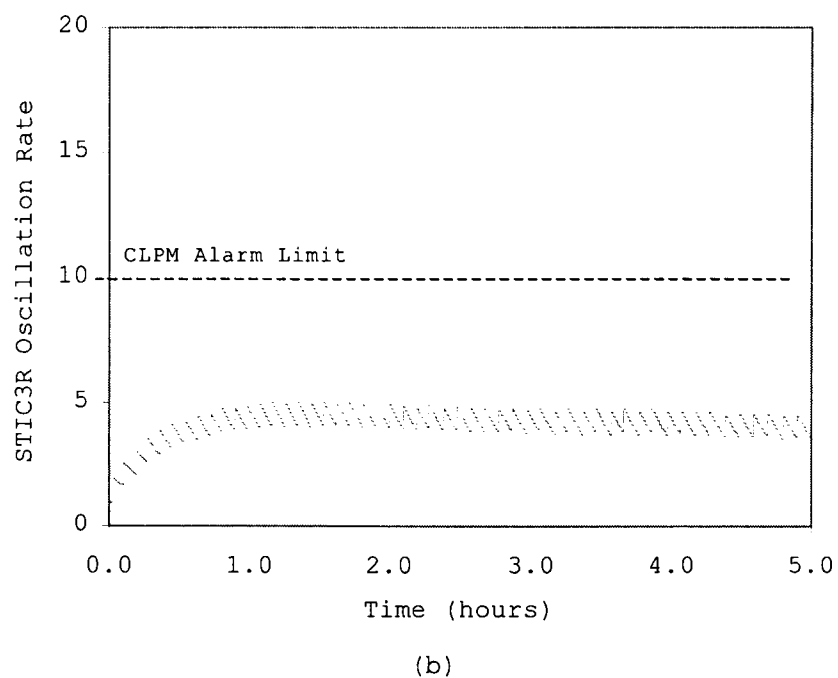
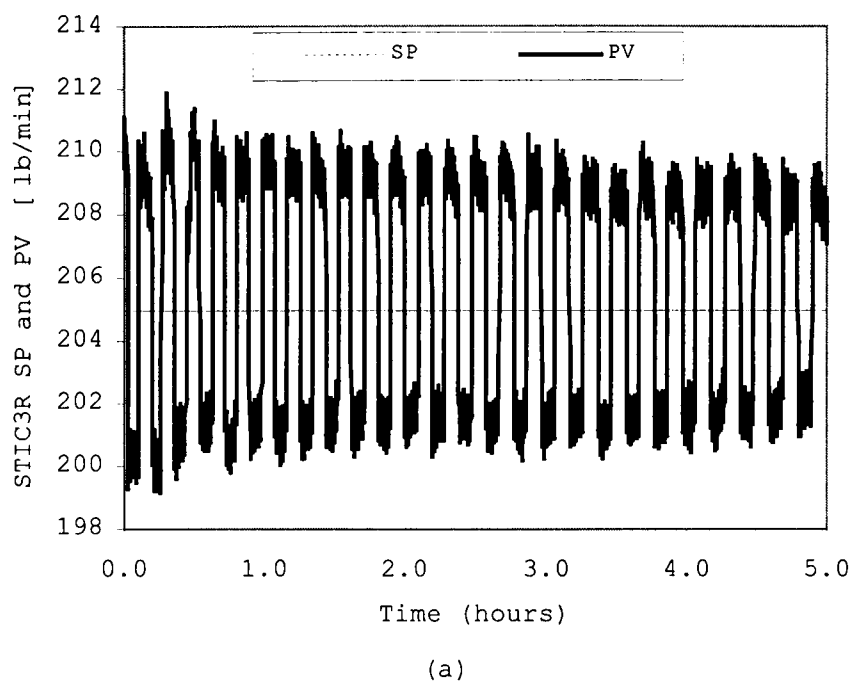
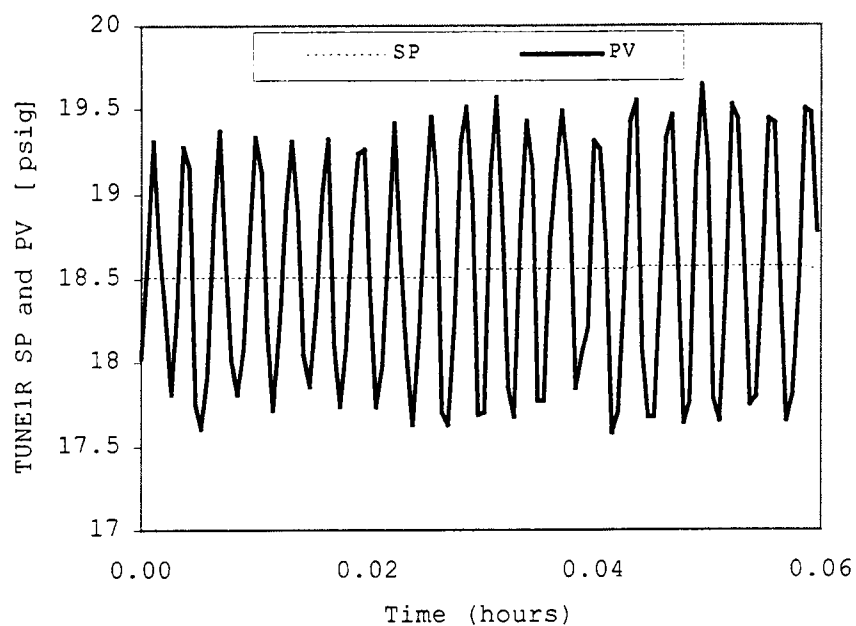


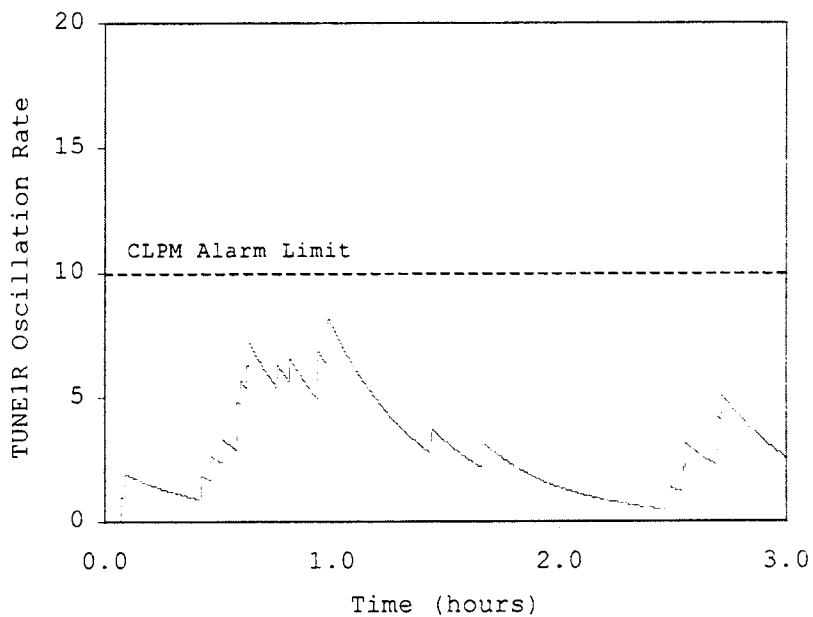
Figure 4.1. Hagglund CLPM applied to real control loop data (STIC3R): (a) SP and PV data, (b) calculated CLPM rate of oscillation - no alarm given.

out at around 4.0, while it must exceed a value of 10.0 for a performance alarm to be initiated. The failure of the CLPM is due chiefly to the poor estimate for T_u which is made by using the controller integral time (the suggested default). Given the actual STIC3R average half period oscillation frequency of approximately 6 minutes (360 seconds), the correct value of T_u for this application would be approximately 720 seconds. Because the T_u used for the CLPM calculations is much too low, the moving time window (T_{sup}), used for discounting the rate of oscillation, is much too small to alarm the low frequency oscillations for STIC3R. A close look at Figure 4.1 confirms this; each half period oscillation is correctly being counted in the oscillation rate, but the oscillation rate is being discounted too quickly, preventing the rate from exceeding the limit of 10. (An additional test for STIC3R using a better T_u value of 720 seconds did produce a performance alarm, with the oscillation rate lining out above 40.)

The CLPM will be further tested on real process data for an oscillating pressure control loop (TUNE1R). The TUNE1R loop has a half period oscillation frequency of 5-6 seconds and the average peak error for each oscillation is roughly 2.3 % of the TUNE1R PV range (0-40 psig). The following user input values will be used for this test: $a = 1.35 \%$, $n_{lim} = 10$ (default), $T_u = 30$ seconds (the actual integral time of the TUNE1R controller), and $T_{sup} = 10 * n_{lim} * (T_u / 2)$ (the moving time window period). For this test (Figure 4.2), the CLPM failed to alarm despite the peak oscillations for TUNE1R significantly exceeding the user specified limit (2.3 % versus the limit of 1.35 %). The CLPM



(a)



(b)

Figure 4.2. Hagglund CLPM applied to real control loop data (TUNE1R): (a) subset of SP and PV data (total data is 3 hours), (b) calculated CLPM rate of oscillation - no alarm given.

oscillation rate never exceeds a value of 8, while it must exceed a value of 10.0 for a performance alarm to be initiated. The failure of the CLPM is again due to the poor estimate for T_u which is made by using the controller integral time (the correct value of T_u for this application would be approximately 12 seconds). Whereas the T_u used for the STIC3R test was too low, the T_u used for this TUNE1R test was too high. Because the T_u value was too high for this application, the IAE_{lim} value was much higher than it should have been (because it is based on a sine wave shape which is affected by the estimated frequency T_u) for the prescribed a value of 1.35 %. The a value of 1.35 % was selected (as opposed to 1.0 %) to show how a poor T_u estimate (too high or too low) can cause the CLPM to fail to alarm even when the user specified error is clearly being exceeded.

Both the STIC3R and TUNE1R tests point out limitations of the CLPM for use as a general flow/pressure control loop performance monitor. While the two examples demonstrated show failure of the CLPM to alarm poor performance, other examples were found where the CLPM initiates a false performance alarm on real control loop data. For cases where the T_u estimate is lower than the actual loop oscillation frequency, the calculated IAE_{lim} value (based on a and T_u) will be lower than it should. This could allow the IAE_{lim} to be violated even when the peak controller error is not violating the user specified a limit. While the lower T_u value will also make the moving time window smaller than it should be, there is a good potential for the time window to be long enough for the alarm limit to be reached, as the default T_{sup} is conservatively defaulted to 10 times longer than theoretically needed.

Thus, a performance alarm could be initiated when the control loop oscillations are not truly violating the user specified a limit value.

4.3 Limitations of Hagglund CLPM in Relation to Desired Algorithm

The CLPM has several desirable features related to its simplicity and ability to function in real time with the continuous chemical process. But, as shown in the previous section, the CLPM has several limitations and potential problems when used for the purposes of a general flow/pressure controller performance monitoring algorithm. To be useful on a widespread basis, the use of default user input values is critical. As shown, using the suggested default values for the Hagglund CLPM can lead to failures to alarm as well as false performance alarms. The sine wave approximation for defining the IAE_{lim} provides problems because the user specified oscillation limit value a is only completely accurate if the user estimate of T_u is completely accurate. Because the CLPM was designed for detection of oscillations in a given frequency window, using the CLPM for a general monitoring algorithm to detect problems at any frequency is not feasible. The testing of the CLPM is very instructive in helping define process data characteristics which need to be addressed by a general performance monitoring algorithm.

Based on the findings in testing the CLPM on process data, development of the specific proposed Performance Monitoring Algorithm (henceforth noted as PMA) should address the following: 1) the oscillation limit should be defined concretely such that the algorithm will only count

oscillations that truly exceed the user specified limit value, 2) setting an alarm limit based on exceeding a specified number of oscillations may not be optimal due to the uncertainty in the actual oscillation frequency for a given control loop (i.e., the desired alarm limit for the number of oscillations may increase as the oscillation frequency increases), and 3) provision must be made to ensure alarming of poorly-performing loops exhibiting oscillations in any reasonable frequency range. These issues will be considered in development of the PMA.

4.4 Performance Monitoring Algorithm (PMA) Development and Testing

Based on the findings of the CLPM testing and the structure of the CLPM, visual inspection of process data characteristics, the necessity of developing an algorithm with very generic inputs, and the monitoring of real flow control loops displaying both poor and desirable performance, a basic structure for a proposed Performance Monitoring Algorithm (PMA) was developed. This structure is as follows. User inputs to the algorithm are: 1) percent of controller PV range limit PV_{lim} (for counting oscillations), 2) desired time window duration for monitoring T_{mon} , and 3) a percent of time oscillating limit P_{alm} (for initiating a performance alarm). Much like the CLPM, the PMA will monitor the controller error and look at the user specified percent of PV range limit to determine if a countable half period oscillation has occurred. Unlike the CLPM, the PMA will count an oscillation if the controller error exceeds the user specified limit (in percent of PV range) at any point between

successive zero crossings of the controller error (i.e., an oscillation will be counted if the peak deviation from setpoint between error crossings exceeds the specified limit PV_{lim}). Thus, the PMA will not use an IAE type limit, but will directly compare the controller error (in % of PV range) with the user specified limit (PV_{lim}). When a limit violation occurs, an oscillation will be counted and the total time of the oscillation (between controller error sign changes) will be added to a total time oscillating counter. This counter will be filtered with a first order filter (using a time constant equal to the user specified time window T_{mon}) to give a moving value for the total time oscillating for the loop. This value will be converted to a percent of time oscillating value (the percent of time during the moving time window that the loop is in a countable oscillation). The percent of time oscillating will be compared with the user specified limit (P_{alm}) and if the limit is exceeded, a performance alarm will be initiated. Suggested default values for implementation for basic flow/pressure controllers on continuous chemical processes are $PV_{lim} = 1.0 \%$, $T_{mon} = 8$ hours, and $P_{alm} = 50.0 \%$. The suggested default value for PV_{lim} (1.0 %) is analogous to the CLPM default a value of 1.0 % and is chosen to be well above the measurement noise and allow for some slight deviation from the controller setpoint. Although the specifics of each control loop will vary widely, a deviation from the setpoint of greater than 1.0 % is typically significant and should be counted as an oscillation. The suggested default value for T_{mon} (8 hours) is chosen to provide a long monitoring window and help minimize false performance alarms due to short-lived disturbances. The suggested default value for P_{alm} (50.0

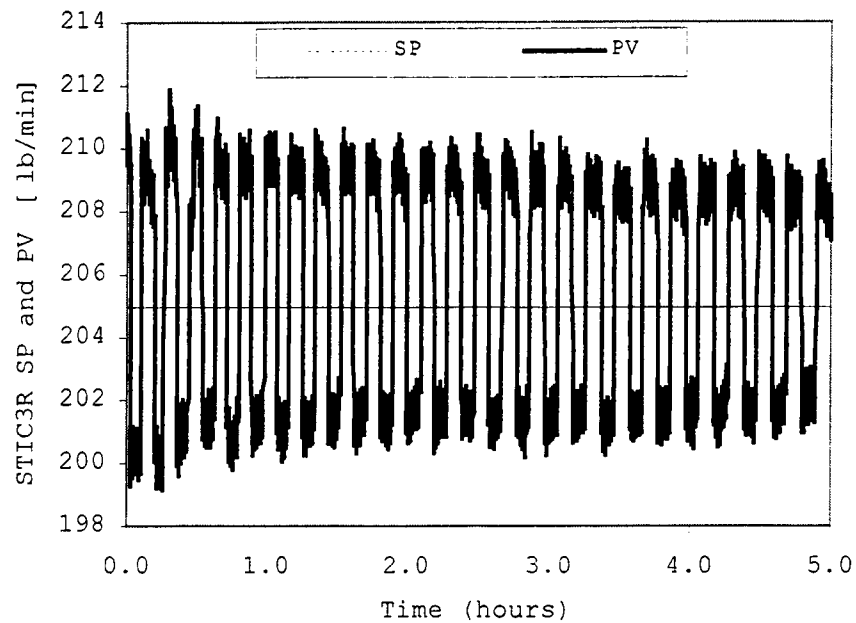
%) is chosen to help ensure that oscillations are sustained for significant periods of time (e.g., approximately 4 hours for a T_{mon} of 8 hours) and to help ensure that multiple oscillations are counted before an alarm is given. Using values for P_{alm} significantly less than 50.0 % (e.g., 20.0-30.0 %) could lead to alarms for problems which are only temporary. Using values for P_{alm} greater than 70.0 or 80.0 % could result in loops with poor performance not being alarmed (e.g., loops with hysteresis problems which are effectively oscillating continuously may only have a maximum percent time oscillating of 70.0-80.0 %). Thus, the suggested default value for P_{alm} of 50.0 % is thought to provide a good balance for minimizing false alarms and for positively detecting poor performance.

While the percent time oscillating will be the value monitored for alarming purposes, the monitoring of other parameters may also be of significant value. These parameters include the average peak deviation from setpoint (for each countable oscillation) and the average half period oscillation frequency (for each countable oscillation), both calculated for the moving time window T_{mon} . For example, if plant personnel notice a control loop which is only oscillating 30 or 40 % of the time but has an unusually high average peak deviation from setpoint (e.g., 10.0 %), they may wish to take action on that particular loop.

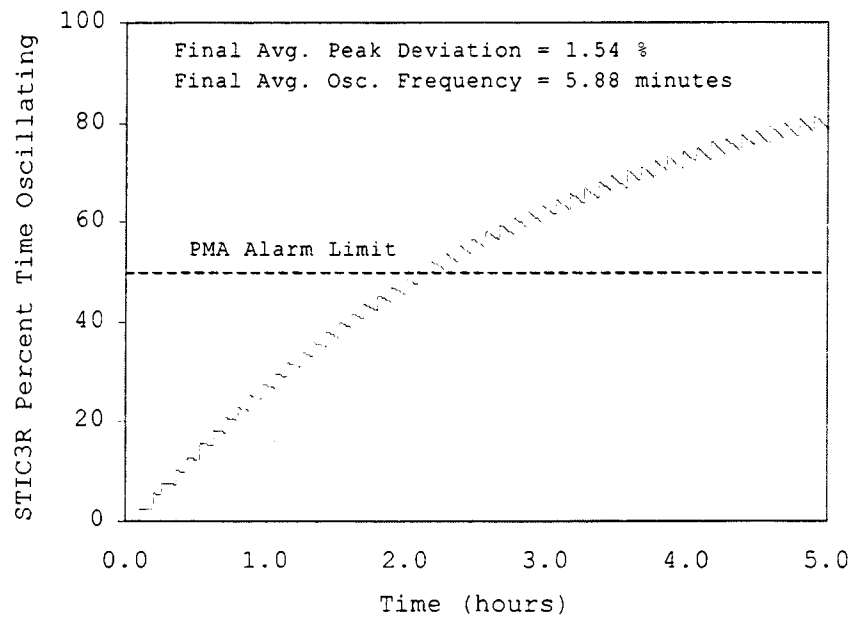
For an initial test, the PMA will be applied to the same STIC3R loop data (Figure 4.1) used for testing the CLPM. The selected user input values are: $PV_{lim} = 1$ %, $T_{mon} = 3$ hours, and $P_{alm} = 50.0$ %. These values

are the suggested default values except for T_{mon} (3 hours is used for T_{mon} because there are only 5 hours total for the STIC3R loop data). Given the structure of the PMA and the specified user inputs, the PMA was tested by programming an m-file (PMA_ALG.M, Appendix C) in Matlab to perform the PMA calculations on the STIC3R process data. For this test (Figure 4.3), the PMA alarmed just after 2 hours. This was a successful alarm due to the peak deviation from setpoint consistently exceeding the specified limit (1.5 % versus the limit of 1.0 %). Additional useful information which is listed (Figure 4.3) but not plotted: the final average peak deviation of 1.54 % and the final average half period oscillation frequency of 5.88 minutes. The PMA successfully alarms for this application in which the CLPM did not. It can also be shown that the PMA successfully alarms for the same TUNE1R loop data used for testing the CLPM (Figure 4.2), and that the PMA successfully does not alarm for approximately 8 hours worth of process data for a flow loop exhibiting good performance (GOOD1R).

For further testing, the PMA will be applied to the HYST4R loop data. The HYST4R loop is oscillating due to valve hysteresis at an average peak deviation of approximately 2.5 %, thus the PMA should alarm. The selected user input values are: $PV_{lim} = 1 \%$, $T_{mon} = 3$ hours, and $P_{aldm} = 50.0 \%$. These values are the suggested default values except for T_{mon} (3 hours is used for T_{mon} because there are only 8 hours total for the HYST4R loop data). For this test (Figure 4.4), the PMA successfully alarms at approximately 3 hours. Note that although each significant change in error for the HYST4R loop is being counted as an oscillation, the percent time oscillating appears to be lining out at

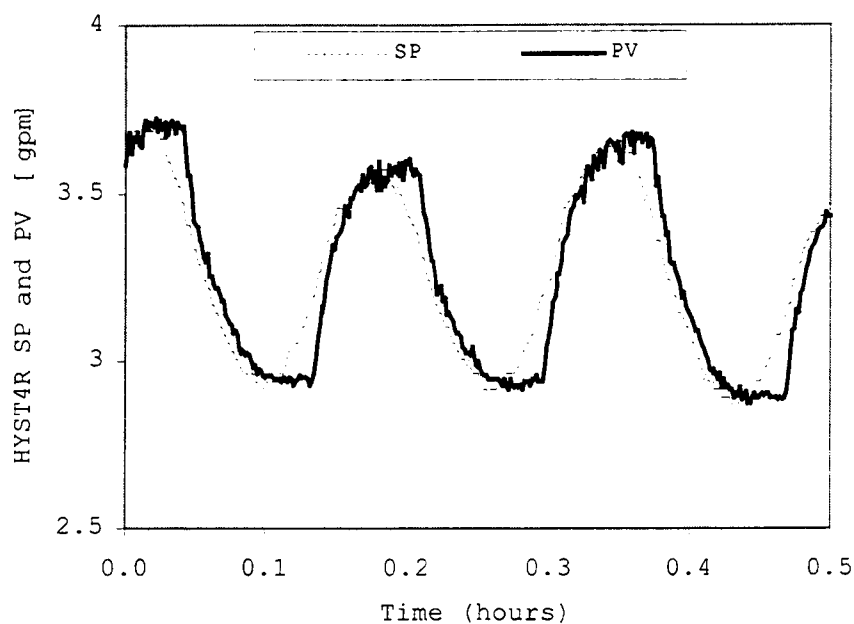


(a)

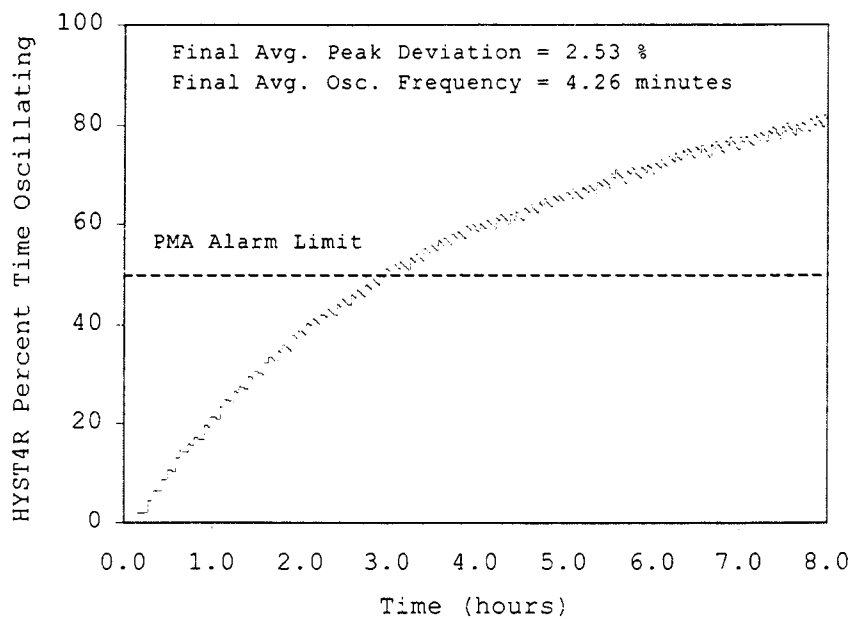


(b)

Figure 4.3. PMA applied to real control loop data (STIC3R):
 (a) SP and PV data, (b) calculated PMA percent time oscillating -
 alarm given just after 2 hours.



(a)



(b)

Figure 4.4. PMA applied to real control loop data (HYST4R):
 (a) subset of SP and PV data (total data is 8 hours), (b) calculated PMA percent time oscillating - alarm given just before 3 hours.

approximately 80 % (not 100 %). This is due to the nature of the hysteresis problem, as once the valve begins to move as requested, the PV does move relatively smoothly and for short periods of time is at or near the setpoint. Additional useful information which is listed (Figure 4.4) but not plotted with time: the final average peak deviation of 2.53 % and the final average half period oscillation frequency of 4.26 minutes.

4.5 Advantages of the Developed Performance Monitoring Algorithm

As shown, the PMA is successful at alarming poor performance for instances where the CLPM was unsuccessful. Due to the structure of the PMA and its development as a general controller monitoring algorithm, the PMA is robust and reliable in detecting poor control performance. Advantages of the PMA over the CLPM in relation to detecting poor performance include: 1) the PMA uses a direct limit (PV_{lim}) for counting oscillations which avoids the CLPM sine wave approximation errors (related to IAE_{lim}) due to oscillation frequency uncertainty, 2) the default PMA input parameters can be more easily selected and applied to large numbers of flow/pressure control loops, 3) the PMA moving time window input T_{mon} can be set for a long monitoring window to help minimize false alarms and ensure that the violating oscillations are sustained and not due to a short-lived disturbance, 4) the PMA ability to alarm is independent of oscillation frequency - this is very useful for detecting loops with common controller problems such as valve stiction or valve hysteresis which tend to exhibit low frequency (relative to the ultimate period of the

controller) oscillations, and 5) the PMA alarm limit P_{alm} is based on a percent of time that poor performance is occurring, as opposed to the CLPM which uses an arbitrary alarm limit (for the number of oscillations) which can be easily set too high or low depending on the actual loop oscillation frequency.

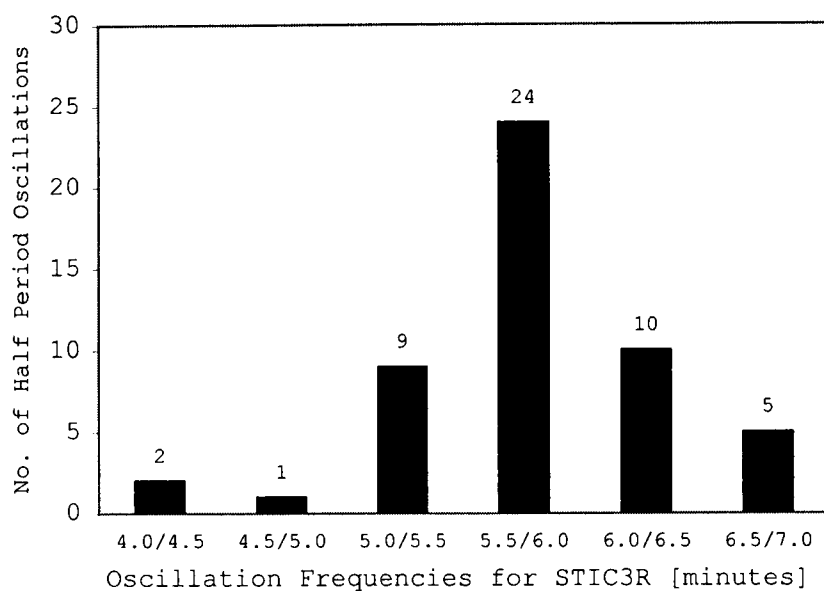
4.6 Complementary Analyses for the Performance Monitoring Algorithm

While the percent time oscillating parameter for the PMA could be the sole value monitored for performance alarming purposes, there are other analyses which are complementary to the PMA in providing valuable insight into control loop performance. These analyses are: 1) a histogram of the peak deviation (peak % error) values for each half period oscillation, and 2) a histogram of the half period oscillation frequencies. The deviation and oscillation histograms demonstrate the characteristics of the control loop oscillations by displaying the variability of the deviation values and oscillation frequencies, as well as the actual values for a given time window of operation. These analyses may be useful for controller troubleshooting and for helping diagnose specific controller problems. A clustering of most of the oscillation frequencies at a high frequency may signify disturbance or tuning problems, while clusters at low frequencies may help to rule out the presence of tuning problems. Excessive variability in peak deviation values could signify that combinations of problems are occurring or that non-cyclic disturbances are present. Visual comparison of these analyses for different control loops (in the same process) could help identify

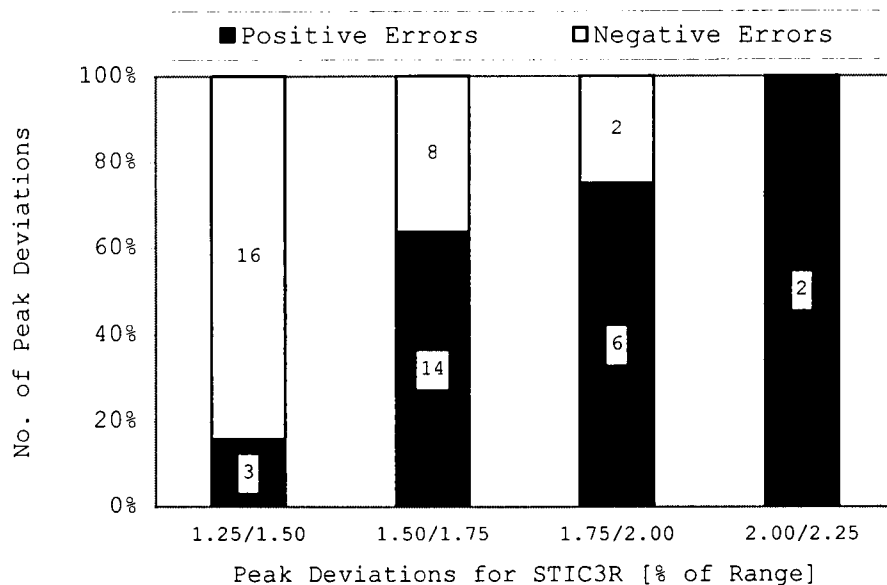
loops which are interacting with one another. As another example, for a valve stiction problem, significant magnitude differences for positive and negative peak deviations would visually indicate different degrees of valve stiction for increasing and decreasing controller outputs. Plots of these analyses (Figure 4.5) are shown for the STIC3R loop data (previously shown in Figure 4.1). These plots visually show that there is some spread in the oscillation frequencies for STIC3R and also show that the positive error excursions typically result in larger deviations from setpoint (relative to the negative error excursions). Thus, there is a difference in the positive and negative degrees of stiction for STIC3R. If the problem was unknown, this information could also be used to infer that the problem is probably not tuning, due to the oscillations being of low frequency and also due to the variations in the oscillation frequencies.

4.7 Extending the PMA to Other Types of Control Loops

Though not a goal of this work, some discussion is merited concerning the extension of the PMA for monitoring the other major types of PID control loops in continuous chemical processes: temperature, level, analysis, etc. The major change needed for applying the PMA to the other major types of control loops should be modification of the default values for the user inputs PV_{lim} and T_{mon} . Because the other types of control loops are typically slower responding and oscillate at lower frequencies, the default value for the time window T_{mon} may



(a)



(b)

Figure 4.5. Frequency and peak deviation analyses applied to real control loop data (STIC3R): (a) half period oscillation frequencies, (b) peak deviation from setpoint for each oscillation.

need to be increased to 16 or 24 hours. In addition, the PV_{lim} value may need to be increased above 1.0 %, as the desired performance for many of the other types of loops may be to allow more controller PV variation for purposes of absorbing some of the process disturbances. For some of these loops, lessening the controller OP variation at the expense of increasing the controller PV variation may be desirable (this is especially true for level controllers for in-process surge tanks where level variation is much preferred over inlet or outlet flow variation). Thus, the default PV_{lim} and T_{mon} values for a given loop could be set based on where the loop fits in groupings or classes of loops (e.g., loops needing tight control, loops where significant variation is acceptable within certain bounds, etc.). Much study and testing with real process data would be needed to effectively develop these default PV_{lim} and T_{mon} values and loop groupings/classifications.

4.8 Other Potential Monitoring Algorithm Needs

An issue which should also be considered with regard to the PMA is the need for monitoring the controller output. Just as the controller PV is monitored for sustained oscillations, should the controller OP be monitored for sustained oscillations? For most loops, if the controller PV is not oscillating significantly, then neither is the controller OP. For some cases, however, where the controller tuning is very aggressive or where certain process disturbances or problems are present, the controller output will be oscillating significantly even when the controller PV is not deviating significantly from setpoint. For example, a flow loop which is holding the PV close to

SP with a slight oscillation may contain a significant controller OP oscillation due to a valve problem or an oscillating process disturbance. For another example, a level control loop which is overtuned may be within 1 % of SP but may be swinging its output wildly, disturbing other parts of the process. For both cases, there would be value in detecting the controller OP oscillation. If the controller OP is to be monitored, another issue arises. When the PV is monitored, the reference for calculating percent error is easily found with the controller SP. For the controller OP, there is no easy reference value for calculating percent deviations from target. As a remedy, perhaps the controller OP data could be filtered with a high-pass filter such that the mean changes are removed, leaving a valid reference of 0.0 for calculating percent deviation from target.

CHAPTER 5

APPLICATION OF CROSS CORRELATION ANALYSIS IN DIAGNOSIS OF COMMON CONTROLLER PROBLEMS

5.1 Introduction to Cross Correlation Analysis

Cross correlation measures the linear dependence between signals (or process data variables) at different sample lags. Because it analyzes at different sample lags, cross correlation analysis gives valuable dynamic information. Because it measures linear dependence, cross correlation analysis captures the most dominant cause/effect relationships between two different signals.

The four common problems (valve stiction, valve hysteresis, poor tuning, and process disturbances) for which diagnosis techniques are to be developed often display unique relationships between the controller PV and OP variables. For example, valve stiction and hysteresis often produce large dead times between controller OP changes and the resulting PV changes. Tuning and process disturbance problems often result in PV and OP relations being completely out of phase. As a result, patterns or quantitative inferences from PV/OP cross correlation analyses may be useful as problem diagnosis techniques. In the following sections, PV/OP cross correlation analyses will be applied to controller data (containing the four common problems) in hopes of developing valuable problem diagnosis techniques.

5.2 Data Preconditioning with a High Pass Filter (Prior to Analysis)

Because signal (process variable) means are used in calculation of cross correlation coefficients, the direct application of cross correlation analysis to control loop data with mean changes can manifest misleading results. If the mean changes are sustained for significant periods of time, the correlation coefficients at all sample lags will show significant positive correlation (i.e., the calculated process variable means (signal means) will be somewhere in between the mean changes, and the majority of both the sampled controller PV and OP values will be above or below their respective mean values for long periods of time). False correlations will be seen at long sample lags (well beyond any dynamic relations) when there is actually no real correlation present.

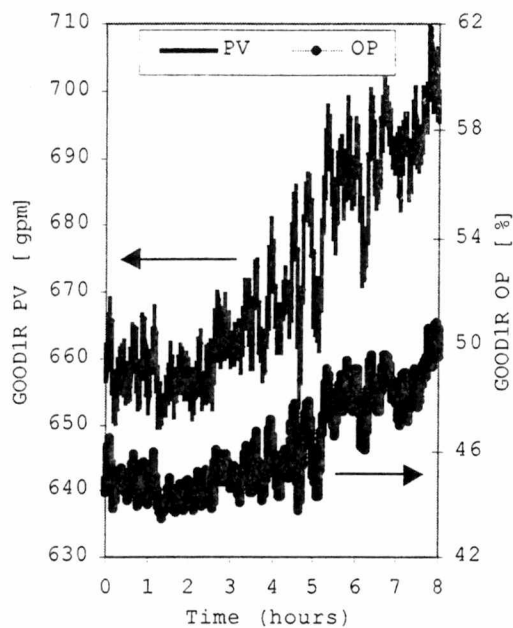
While there are several options for addressing this problem (data differencing, subtracting a moving mean from the data, etc.), the most effective means has been to apply a simple first order high pass filter to the control loop data prior to cross correlation analysis. The high pass filter effectively removes the unimportant mean changes from the data, allowing the important dynamic relations between the PV and OP to be retained and accurately analyzed. The high pass filter in effect allows longer durations of data to be analyzed (and in turn it allows loops with constantly changing setpoints to be analyzed). Use of a high pass filter also helps to ensure that correlation analyses for the different control loop problems can be compared on the right basis. The effectiveness of using the high pass filter is

illustrated for loop data for GOOD1R, a loop displaying good performance (Figure 5.1). Application of cross correlation directly to the raw GOOD1R data gives false results (significant positive correlations for all lags up to 30 minutes), while application of cross correlation to the high pass filtered data allows the true highly positive correlations (at a low lag) between the OP and PV to be captured. The positive correlations at all frequencies present for the raw data are no longer present for the filtered data.

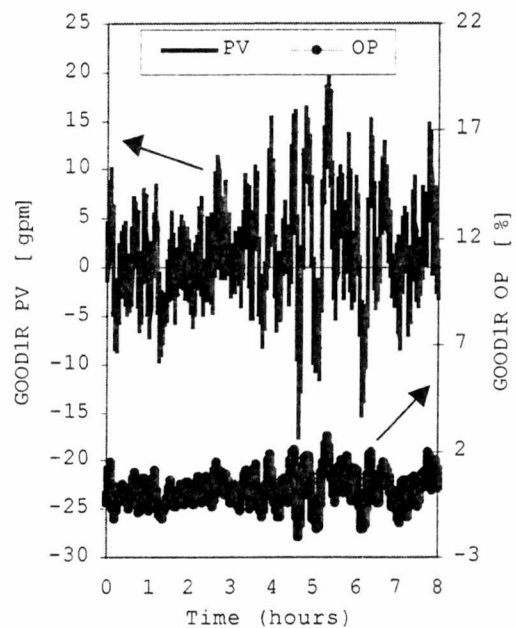
Caution must be used when applying the high pass filter to precondition the data. Filtering too much (e.g., filtering out information at frequencies higher than the oscillation frequency) can cause valuable PV/OP information to be lost. Filtering should be done just enough to remove the lower frequency mean changes from the data, leaving the rest of the data for cross correlation analysis. One suggested method for determining the high pass filter time constant is to base it on the predetermined half period oscillation frequency of the data. In general, a filter time constant value significantly greater than the half period oscillation frequency should be used. For many basic flow/pressure control loops, a high pass filter time constant of 10-15 minutes is a good starting point.

5.3 Application of PV/OP Cross Correlation Analysis

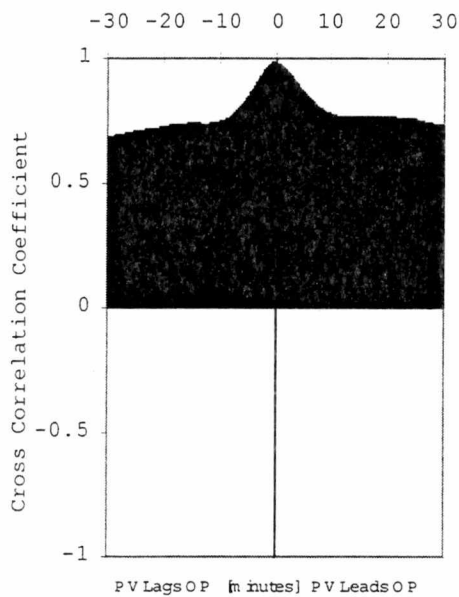
While there is certain utility in other types of cross correlation analyses, this work will focus on PV/OP cross correlations. It is



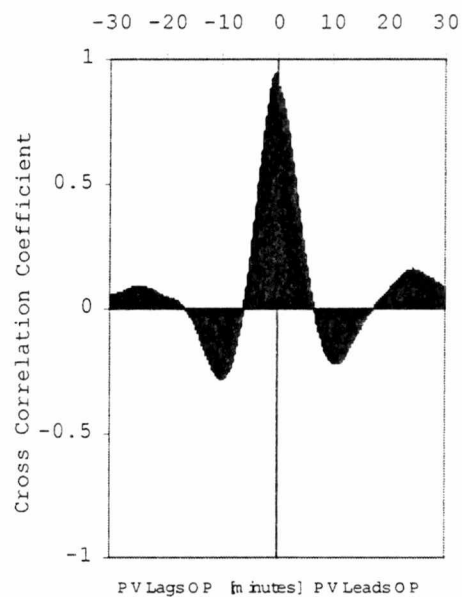
(a)



(b)



(c)

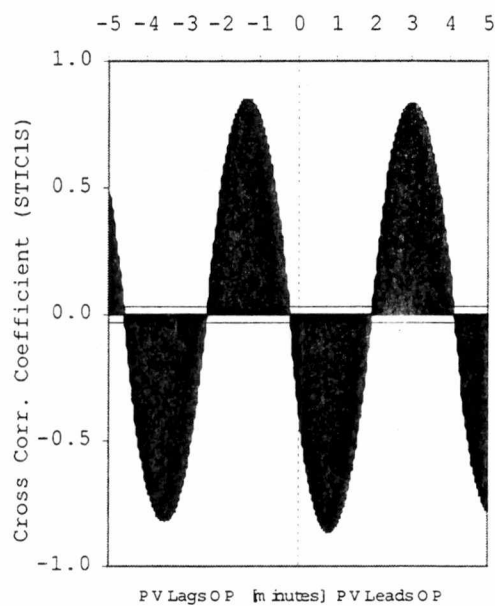


(d)

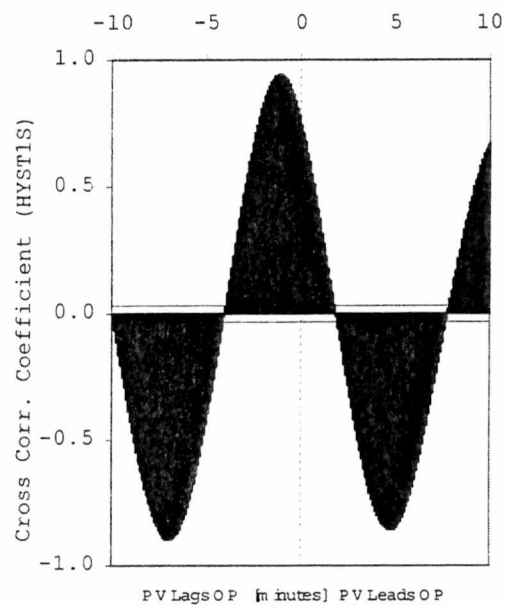
Figure 5.1. Effect of high pass filter with cross correlation analysis, for controller with good performance (GOOD1R): (a) raw PV/OP data, (b) PV/OP data, filtered with 20 min. constant, (c) cross correlation for raw data, (d) cross correlation for filtered data.

felt that these analyses provide the greatest value in developing problem diagnosis techniques for the four common problems of interest. In hopes of finding useful diagnosis techniques, PV/OP cross correlations will be applied to the simulated and real data sets containing the four common control problems, as well as to the data sets containing good performance. Visual inspection and comparison will then be made of the resulting correlations to identify patterns or quantitative inferences which may be useful as problem diagnosis techniques.

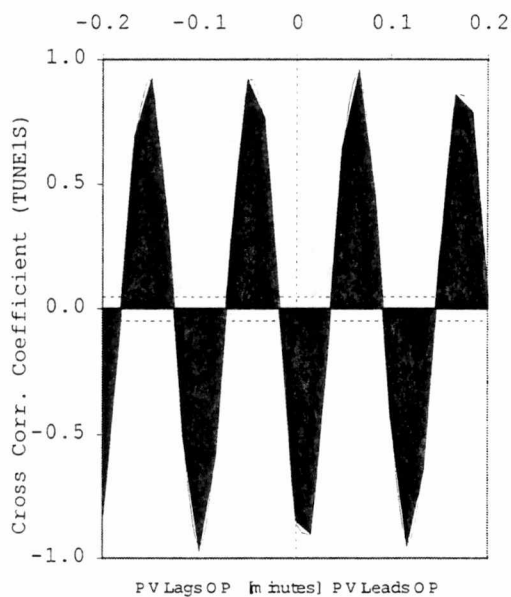
The procedure for applying the PV/OP cross correlation analyses is as follows. The previously obtained data sets are loaded into Matlab, and a user written Matlab m-file containing Signal Processing Toolbox correlation routines (CCORR_PLOT.M, Appendix C) is then used to calculate the PV/OP cross correlations. The m-file applies a user specified high pass filter to the PV and OP data prior to performing the correlation calculations. The correlations are calculated out to sufficient time lags to capture all important dynamic information. The resulting PV/OP cross correlations show the OP to PV lag relationships as correlations at negative time values, while the PV to OP lead relationships are shown at positive time values (e.g., a positive correlation at -2.0 minutes signifies that a change in the OP at time zero affects the PV strongly in the same direction at a time two minutes later). Selected PV/OP cross correlations for each of the four common problems are shown in Figure 5.2 (simulated controller problems) and Figure 5.3 (real controller problems). For comparison, correlations for good performance are shown in Figure 5.4 (simulated



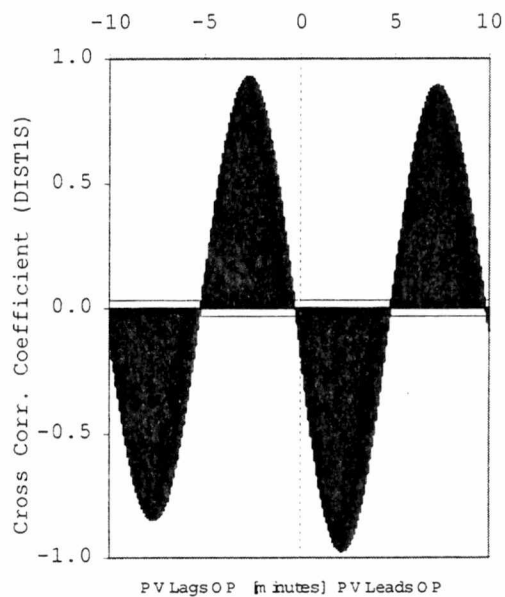
(a)



(b)

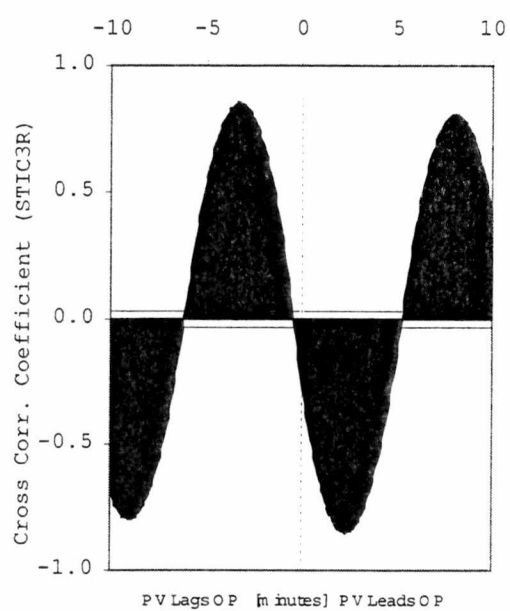


(c)

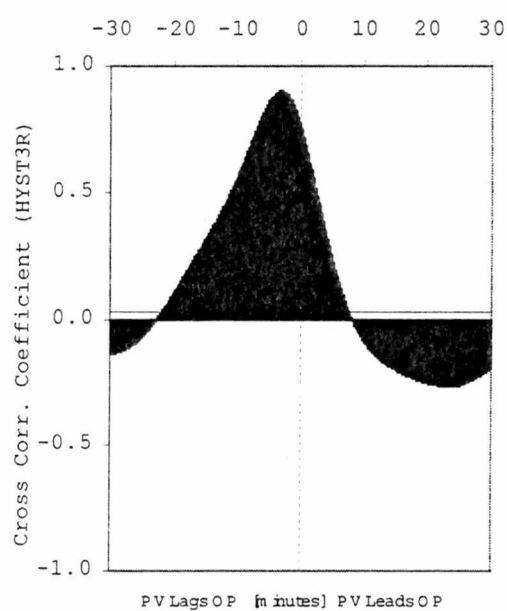


(d)

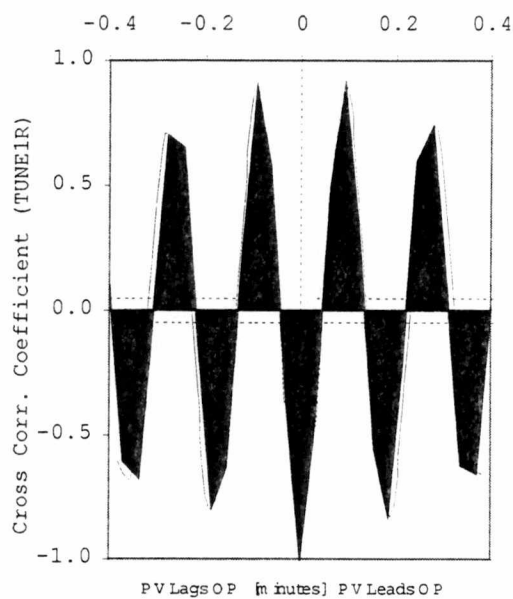
Figure 5.2. Comparison of PV/OP cross correlations (simulated data):
 (a) PV/OP correlation for STIC1S, (b) PV/OP correlation for HYST1S,
 (c) PV/OP correlation for TUNE1S, (d) PV/OP correlation for DIST1S.



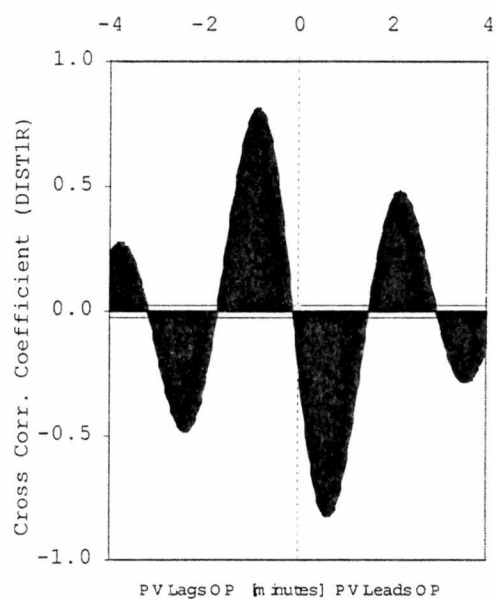
(a)



(b)

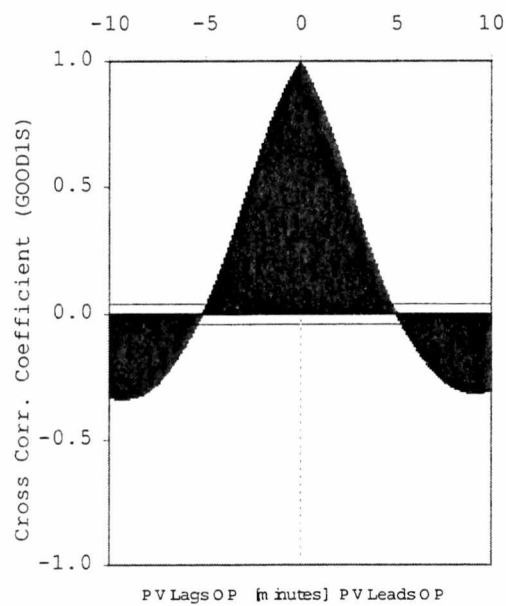


(c)

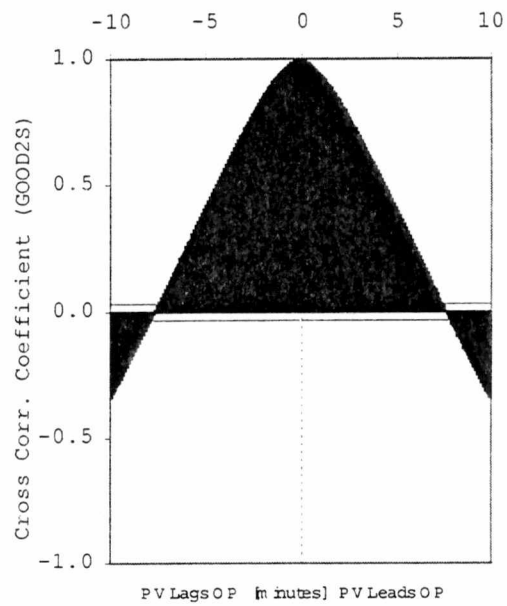


(d)

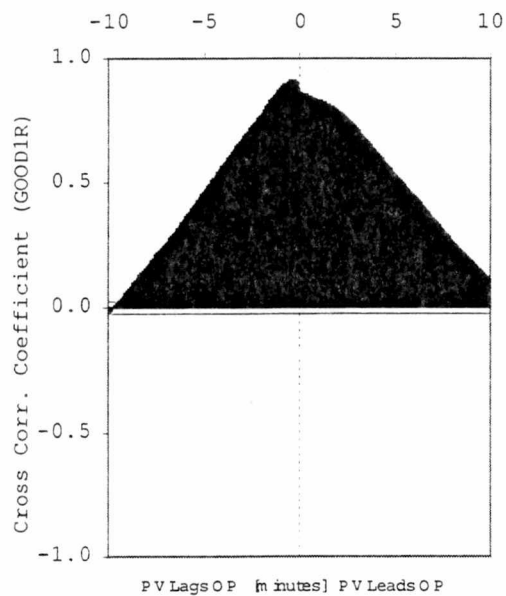
Figure 5.3. Comparison of PV/OP cross correlations (real data): (a) PV/OP correlation for STIC3R, (b) PV/OP correlation for HYST3R, (c) PV/OP correlation for TUNE1R, (d) PV/OP correlation for DIST1R.



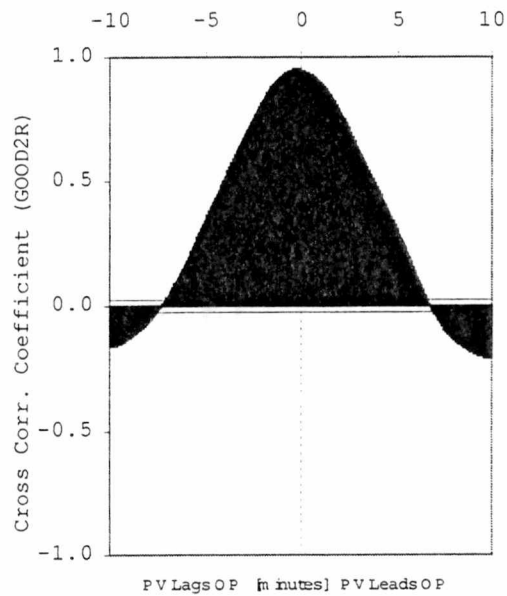
(a)



(b)



(c)



(d)

Figure 5.4. Comparison of PV/OP cross correlations (simulated/real data): (a) PV/OP correlation for GOOD1S, (b) PV/OP correlation for GOOD2S, (c) PV/OP correlation for GOOD1R, (d) PV/OP correlation for GOOD2R.

and real data). PV/OP cross correlations for all the simulated and real data sets are shown in Figures B.1-B.10.

Several parameter values which are descriptive of PV/OP cross correlation characteristics and patterns can be easily obtained from the correlation coefficients and lead/lag time data. These parameters are useful for quantitatively describing specific characteristics of the cross correlation analyses for the 4 common controller problems. These parameters are listed in Table 5.1.

5.4 Relation of PV/OP Correlation Patterns to Control Loop Problems

Based on visual inspection of the PV/OP cross correlations and the descriptive parameters listed in Table 5.1, the following inferences can be made:

- Hysteresis is the only one of the 4 common problems which has a positive correlation coefficient at time zero, and the PV/OP lag first positive peak time for hysteresis is closer to time zero than for the corresponding PV/OP lead first negative peak time.
- Stiction, disturbance, and tuning problems have a negative correlation coefficient at time zero, and their PV/OP lead first negative peak time is closer to time zero than their corresponding PV/OP lag first positive peak time.
- The correlation coefficient magnitude for the PV/OP lag first positive peak varies somewhat among the 4 common problems, as it is typically in the range of: 0.6-0.9 for stiction problems, 0.9-1.0 for hysteresis problems, and .88-1.0 for tuning problems. It is widely variable for disturbance problems.
- The time for the PV/OP lag first positive peak also varies among the 4 common problems, as would be expected. It is typically in the range of: 30 seconds to several minutes for stiction and hysteresis problems, and 15 seconds or less for tuning problems. It is widely variable for disturbance problems.

Table 5.1. Descriptive parameters from cross correlation analysis.

Loop ID	Problem	Corr. Coeff. at Time Zero	FV/OP Lag 1st + Peak	PV/OP Lag 1st + Peak [min]	PV/OP Lead 1st - Peak	PV/OP Lead 1st - Peak [min]	PV/OP Lag Peak to Osc. Freq. Ratio	PV/OP Lead Peak to Osc. Freq. Ratio
STIC1S	Stict.	-0.3713	0.8495	-1.330	-0.8612	0.817	0.610	0.375
STIC2S	Stict.	-0.7172	0.9299	-0.367	-0.9381	0.100	0.777	0.212
STIC3S	Stict.	-0.3564	0.8626	-1.320	-0.8670	0.833	0.606	0.382
STIC4S	Stict.	-0.2880	0.7056	-1.420	-0.7164	1.020	0.546	0.392
STIC5S	Stict.	-0.0435	0.7527	-0.617	-0.4010	0.500	0.334	0.270
STIC6S	Stict.	-0.5746	0.6711	-0.633	-0.6881	0.267	2.277	0.960
STIC1R	Stict.	-0.0435	0.7106	-5.010	-0.7582	4.890	0.406	0.396
STIC2R	Stict.	-0.4267	0.8878	-0.350	-0.9272	0.180	0.624	0.321
STIC3R	Stict.	-0.2672	0.8509	-3.330	-0.8546	2.330	0.581	0.407
STIC4R	Stict.	-0.3646	0.7673	-0.942	-0.8163	0.514	0.624	0.340
HYST1S	Hyst.	0.7669	0.9419	-1.100	-0.8564	4.750	0.190	0.820
HYST2S	Hyst.	0.7570	0.9151	-0.920	-0.8352	5.020	0.354	1.931
HYST3S	Hyst.	0.8216	0.9024	-1.520	-0.7311	12.350	0.157	1.272
HYST4S	Hyst.	0.8366	0.9535	-0.883	-0.8490	5.030	0.150	0.855
HYST5S	Hyst.	0.9607	0.9896	-0.450	-0.7978	5.600	0.112	1.393
HYST1R	Hyst.	0.8120	0.9601	-2.360	-0.9060	13.530	0.199	1.140
HYST2R	Hyst.	0.5585	0.7047	-1.370	-0.2035	14.530	0.895	9.497
HYST3R	Hyst.	0.7650	0.9006	-3.080	-0.2690	23.670	0.315	2.423
TUNE1S	Tuning	-0.8457	0.9153	-0.050	-0.8946	0.017	0.909	0.304
TUNE2S	Tuning	-0.2364	0.9418	-0.067	-0.9796	0.050	0.624	0.466
TUNE3S	Tuning	-0.9982	0.8809	-0.033	-0.9982	0.000	0.857	0.000
TUNE4S	Tuning	-0.1959	0.9941	-0.200	-0.9919	0.150	0.568	0.426
TUNE5S	Tuning	-0.4306	0.9363	-0.250	-0.6791	0.100	0.912	0.365
TUNE1R	Tuning	-0.9942	0.8849	-0.096	-0.9942	0.000	1.043	0.000
DIST1S	Dist.	-0.1325	0.9289	-2.670	-0.9725	2.230	0.560	0.468
DIST2S	Dist.	-0.9538	0.9793	-0.117	-0.9538	0.000	0.936	0.000
DIST3S	Dist.	-0.0933	0.9127	-2.580	-0.9377	2.280	0.585	0.517
DIST4S	Dist.	-0.3087	0.9465	-1.200	-0.9878	0.817	0.606	0.413
DIST5S	Dist.	-0.1424	0.5806	-3.000	-0.6157	2.550	1.382	1.175
DIST6S	Dist.	-0.2005	0.6110	-1.730	-0.6430	1.350	0.746	0.582
DIST1R	Dist.	-0.1922	0.8159	-0.855	-0.8258	0.633	0.545	0.403
DIST2R	Dist.	-0.3163	0.7365	-0.302	-0.8579	0.173	0.531	0.304
GOOD1S	None	0.9935	0.9935	-0.000	-0.3133	9.170	n/a	n/a
GOOD2S	None	0.9909	0.9933	-0.033	-0.6468	14.250	n/a	n/a
GOOD1R	None	0.8631	0.9109	-0.353	n/a	n/a	n/a	n/a
GOOD2R	None	0.9476	0.9512	-0.137	-0.2141	10.270	n/a	n/a

In summary, it appears that PV/OP cross correlation analyses can reliably provide a positive diagnosis for hysteresis problems, by testing for the presence of a positive correlation coefficient at time zero. Beyond this capability, it appears that PV/OP cross correlations are of limited diagnosis utility. Based on correlation parameter values, certain problems can potentially be ruled out. For example, if the correlation coefficient for the PV/OP lag first positive peak is greater than 0.9, stiction problems can probably be ruled out. As another example, if the time for the PV/OP lag first positive peak is greater than approximately 20 seconds, tuning problems can be ruled out. Thus, based on the testing performed in this work, the PV/OP cross correlation, in addition to providing a positive test for hysteresis problems, can also help to rule out potential problems (e.g., stiction, tuning) for some cases.

5.5 Other Valuable Cross Correlation Analyses

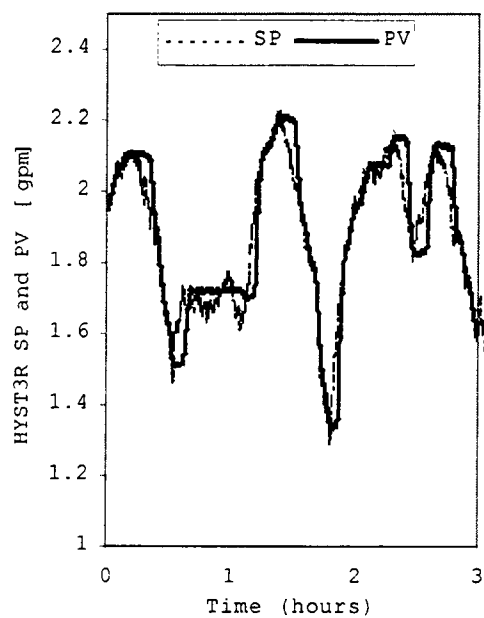
In addition to PV/OP cross correlation analyses, there are other cross correlation analyses which may have some potential use in controller problem diagnosis. The two that will be discussed here are the PV/SP and the error/delta OP (noted as $(SP-PV)/(OP_k-OP_{k-1})$) analyses.

For control loops that are cascade loops and have changing setpoints, a PV/SP cross correlation can be used to determine how well the PV is following the SP. A cascade control loop without a high positive correlation at low values of PV/SP lag can not be performing well. A comparison of PV/SP cross correlations for loops with good and poor

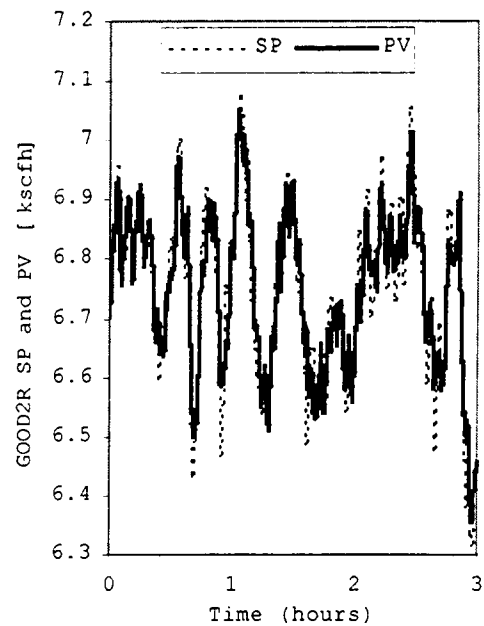
performance demonstrates the utility of this analysis (Figure 5.5). The PV/SP correlation for HYST3R (poor performance) has its highest positive peak at about -2.0 minutes, while the analogous peak for GOOD2R (good performance) is at about -0.4 minutes. The restriction of this analysis to cascade loops with changing setpoints limits its potential applicability.

The $(SP-PV)/(OP_k-OP_{k-1})$ cross correlation analysis may be useful for differentiating process disturbance problems from the other three common problems of interest. Based on observation, it appears that this particular cross correlation analysis is directly related to the Time-Shifted Distribution (TSD) measure (Hackney 1989). As such, loops with process disturbance problems may have different characteristic $(SP-PV)/(OP_k-OP_{k-1})$ cross correlations (e.g., several significant negative correlations) relative to loops with one of the other common problems. An example application of this cross correlation analysis is shown (Figure 5.6) for TUNE1R (poor tuning problem) and GOOD2R (good performance). The cross correlation is highly oscillatory and contains significant negative correlations for TUNE1R, while the correlation for GOOD2R only shows a small positive peak near zero. While these correlations look vastly different and show promise for use in diagnosis of problems, numerous other comparisons using the simulated and real data sets (not shown here) did not yield such dramatic results.

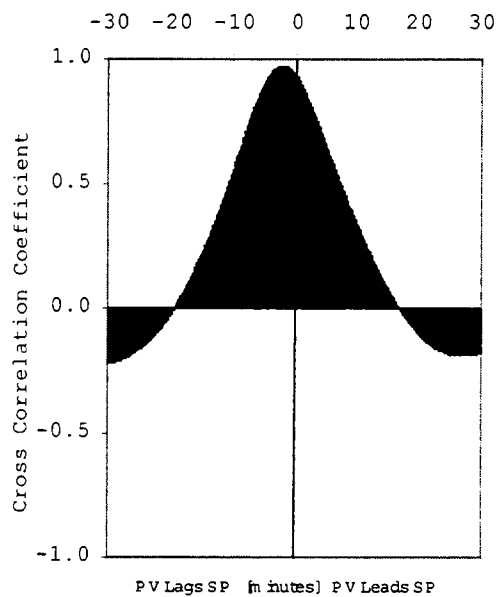
While these additional correlation analyses do not appear to hold as much promise as the PV/OP cross correlations, more investigation is



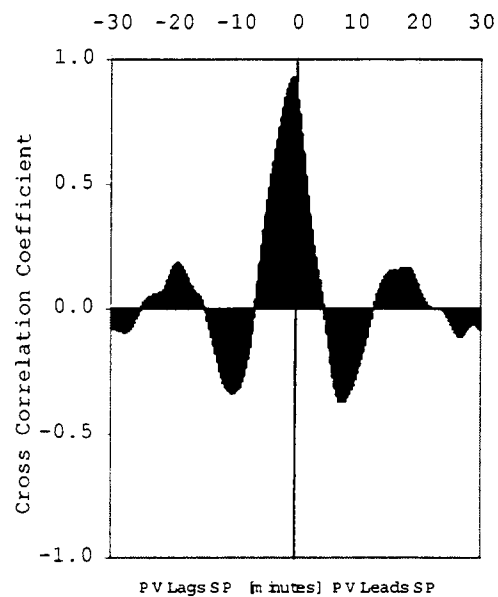
(a)



(b)

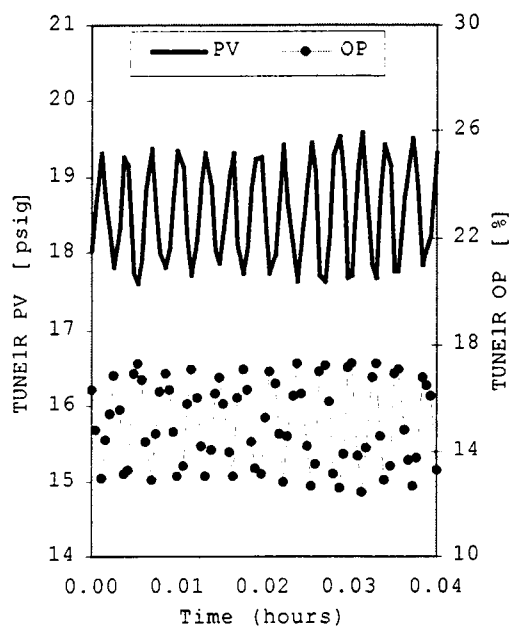


(c)

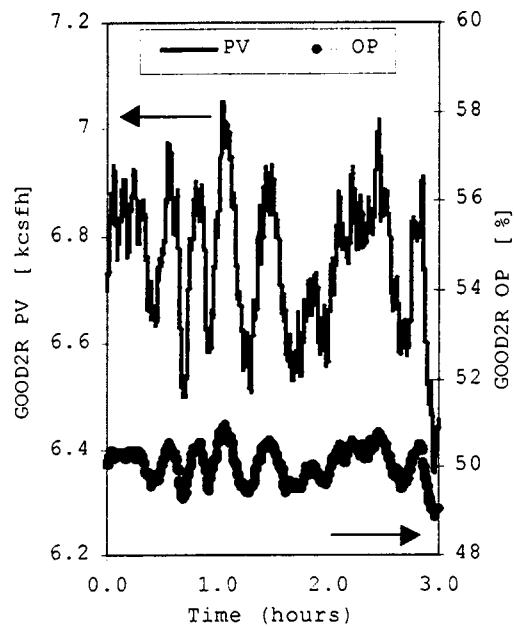


(d)

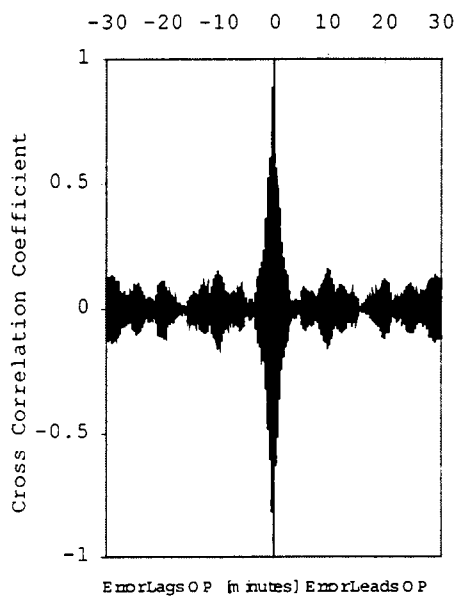
Figure 5.5. Comparison of PV/SP cross correlations for poor (HYST3R) and good (GOOD2R) performance: (a) raw PV/SP data for HYST3R, (b) raw PV/SP data for GOOD2R, (c) PV/SP cross correlation for HYST3R, (d) PV/SP cross correlation for GOOD2R.



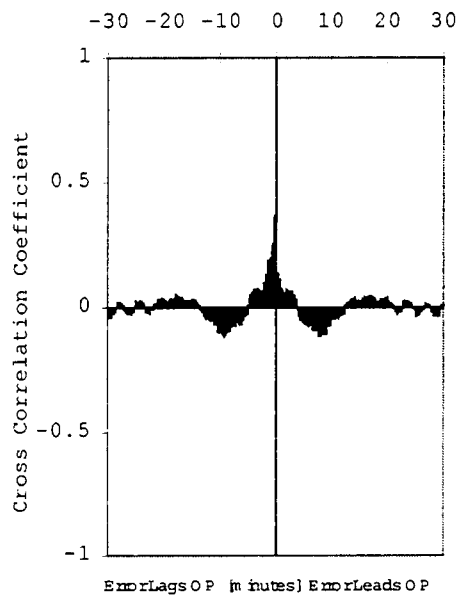
(a)



(b)



(c)



(d)

Figure 5.6. Comparison of error/delta OP cross correlations for poor (TUNE1R) and good (GOOD2R) performance: (a) subset of raw PV/OP data for TUNE1R, (b) raw PV/OP data for GOOD2R, (c) error/dOP cross correlation for TUNE1R, (d) error/dOP cross correlation for GOOD2R.

needed to determine their specific value for problem diagnosis.

CHAPTER 6

APPLICATION OF PRINCIPAL COMPONENTS ANALYSIS IN DIAGNOSIS OF COMMON CONTROLLER PROBLEMS

6.1 Introduction to Principal Components Analysis (PCA)

PCA is used to describe the variation present in a multivariate set of data. PCA finds combinations of variables which describe the major trends in a given data set. One of the strengths of PCA is in describing a data set containing many individual variables with only a few principal components; PCA can often describe the data using far fewer factors than original variables. While this work does not require the data compression of many variables to a few dimensions, the use of PCA to describe the covariance between controller variables may be beneficial in diagnosis of the 4 common controller problems (stiction, hysteresis, tuning, and disturbances). Additionally, PCA is often used in batch processing to compare the characteristics of one batch to another in order to find abnormalities present in the data. Analogously, perhaps PCA can be used in controller problem diagnosis by comparing the characteristics of controller problems with the characteristics of controllers displaying good performance.

For purposes of this work, the data variations of interest are the relations between the controller error (SP-PV) and the controller OP, as well as the relations between the controller PV and the controller OP. As seen in the simulated and real data sets used in this work, the 4 common problems often exhibit unique error/OP and PV/OP patterns

which are significantly different from the corresponding patterns for controllers with good performance. Based on visual inspection of the process data sets for the 4 common problems, the covariance relations between controller OP changes and the resulting PV and error changes appear to vary considerably. As a result, using PCA to compare the data sets for the 4 common problems against data sets for good performance may yield some useful problem diagnosis techniques.

6.2 Application of Selected PCA Methods

Based on the use of PCA in batch processing to compare data for new batches with data for previous batches (to determine abnormalities and other process characteristics), PCA will be used to compare data for the 4 common problems against data for good controller performance. To simplify the analyses, the 3 variables (SP, PV, and OP) for each controller data set will be combined into a 2 column data set: column 1 will be the controller error (SP-PV) and column 2 will be the controller OP. To precondition the data sets prior to applying PCA, the data sets are autoscaled (scaled to mean zero and unit variance). The autoscaling is done individually for each data set to prevent differences in magnitudes of variables from influencing the PCA studies, as differences in shapes and covariances in the data (not magnitudes) are the characteristics of interest. For application of PCA to the simulated data sets, a PCA model with 2 principal components is first developed for GOOD1S, a simulated controller with good performance. Then, data for a given simulated controller problem (e.g., STIC1S) are projected onto the PCA model for good performance.

The resulting new scores for the simulated controller problem are then plotted onto axes defined by the PCA model for good performance. The patterns of the scores relative to the confidence limits for the PCA model, as well as the dominant directions of the scores relative to the PCA model axes, help show the differences in characteristics between the data sets. The simulated data sets for each of the common problems are compared to the data for GOOD1S in this manner. Similarly, the real data sets for the common problems are compared to data for GOOD1R, a controller with good performance. The results of these comparisons are shown in Figures D.1-D.9. Note: analyses were performed via a user written Matlab m-file containing PLS Toolbox routines (PCA_ERR_OP_SCORES.M, Appendix C).

As a further use of PCA methods to aid in problem diagnosis, an investigation of PV/OP covariance will be made to determine if principal component percent variance explained values (from fitting of PCA models) change significantly depending on the type of controller problem. As part of this investigation, in an attempt to add some dynamic content into the data sets to be analyzed, data sets containing multiple time-shifted PV and OP values will be constructed (shifted out to a maximum of 60 sample times). Thus, the data sets will contain 61 PV columns and 61 OP columns ($PV_k, PV_{k+1}, \dots, PV_{k+60}, OP_k, OP_{k+1}, \dots, OP_{k+60}$). Construction of the data sets will be done individually for each simulated and real data set. PCA models containing 5 principal components will then be fit to each data set and the percent variance captured by each of the five components will be obtained (five principal components are arbitrarily chosen as this

should allow almost all of the variance to be explained for each data set). Results of this analysis are shown in Table 6.1. Note: analyses were performed via a user written Matlab m-file containing PLS Toolbox routines (PCA_PV_OP_MODEL.M, Appendix C).

6.3 Usefulness of Selected PCA Methods for Problem Diagnosis

For the PCA methods of projecting problem data sets onto PCA models for good performance, there appear to be some consistent patterns for many of the problem data sets. Although no specific quantitative diagnosis techniques are apparent, visual comparison of the scores plot patterns for different controller problems can be used as a valuable diagnosis tool. Based on visual inspection of the projections of the data sets (for the 4 common problems) onto the PCA models for good performance (Figures D.1-D.9), the following observations can be made. For the simulated data sets: 1) the scores for stiction problems often violate the confidence limits for the PCA model in quadrants 1 and 3, 2) the scores for hysteresis problems often violate the confidence limits in quadrants 2 and 4, 3) the scores for tuning problems, if limits are violated, typically violate the confidence limits in all 4 quadrants, and 4) the scores for disturbance problems are typically all within the confidence limits for cycling disturbances and they violate the limits in quadrants 2 and 4 for square wave disturbances (DIST5S and DIST6S). For the real data sets, somewhat similar observations can be made but the patterns for the different problems are not as consistent. Thus, for many cases, these analyses can provide diagnosis value through visual

Table 6.1. Percent variance captured by principal components for PCA models applied to PV/OP time-shifted matrices.

Loop ID	Problem	PC #1 [%]	PC #2 [%]	PC #3 [%]	PC #4 [%]	PC #5 [%]	Total [%]
STIC1S	Stict.	57.42	36.17	2.57	2.13	0.68	98.97
STIC2S	Stict.	49.79	45.00	1.42	1.30	0.54	98.05
STIC3S	Stict.	57.38	36.36	2.53	2.11	0.67	99.05
STIC4S	Stict.	52.04	34.52	5.99	4.51	1.53	98.59
STIC5S	Stict.	97.03	1.85	0.60	0.21	0.15	99.84
STIC6S	Stict.	34.75	33.34	11.54	3.45	1.72	84.80
STIC1R	Stict.	52.02	46.24	1.13	0.32	0.20	99.91
STIC2R	Stict.	43.88	42.52	2.41	2.16	1.94	92.91
STIC3R	Stict.	36.84	25.65	11.58	6.98	5.16	86.21
STIC4R	Stict.	41.75	37.17	12.31	2.17	2.05	95.45
HYST1S	Hyst.	87.79	11.37	0.46	0.21	0.04	99.87
HYST2S	Hyst.	85.44	12.98	0.68	0.54	0.10	99.74
HYST3S	Hyst.	90.22	9.25	0.31	0.10	0.01	99.89
HYST4S	Hyst.	90.88	8.35	0.36	0.20	0.03	99.82
HYST5S	Hyst.	95.72	3.85	0.07	0.04	0.01	99.69
HYST1R	Hyst.	89.43	10.22	0.24	0.08	0.02	99.99
HYST2R	Hyst.	88.31	9.58	1.77	0.18	0.07	99.91
HYST3R	Hyst.	81.72	11.40	2.34	0.80	0.41	96.67
TUNE1S	Tuning	46.94	45.50	1.65	1.56	0.47	96.12
TUNE2S	Tuning	46.49	45.65	6.54	0.22	0.21	99.11
TUNE3S	Tuning	54.88	15.41	15.08	3.19	3.03	91.59
TUNE4S	Tuning	50.28	49.66	0.01	0.01	0.00	99.96
TUNE5S	Tuning	67.99	13.21	12.56	2.82	1.32	97.90
TUNE1R	Tuning	21.95	21.80	15.75	15.45	5.47	80.42
DIST1S	Dist.	55.63	43.90	0.04	0.01	0.01	99.59
DIST2S	Dist.	49.98	48.56	0.90	0.08	0.07	99.59
DIST3S	Dist.	53.57	44.26	0.11	0.05	0.05	98.04
DIST4S	Dist.	59.42	40.30	0.06	0.02	0.00	99.80
DIST5S	Dist.	51.82	36.84	5.39	2.12	0.99	97.16
DIST6S	Dist.	52.11	35.32	5.66	2.46	1.22	96.77
DIST1R	Dist.	45.12	39.58	10.03	3.76	0.91	99.40
DIST2R	Dist.	34.34	33.23	5.01	4.82	4.40	81.80
GOOD1S	None	99.68	0.31	0.00	0.00	0.00	99.99
GOOD2S	None	99.13	0.64	0.01	0.00	0.00	99.78
GOOD1R	None	90.13	4.04	1.36	1.04	0.54	97.11
GOOD2R	None	97.29	1.38	0.61	0.34	0.13	99.75

analysis of the scores plot patterns for different controller problems.

For the PCA model fits for the time shifted PV/OP matrices (Table 6.1), the main result is that hysteresis problems do appear to be identifiable, as the first principal component captures 80 % or greater variance for all the simulated and real hysteresis data sets. For the other 3 common problems, the first principal component normally captures 30-60 % of the variance. In addition, other inferences can be made: 1) principal components 1-4 are typically all significant for stiction problems, 2) for the simulated disturbance problems due to cycling disturbances, most of the variance is typically explained by the first two principal components (but results are different for the real disturbance problems and for simulated problems due to square wave type disturbances), and 3) for tuning problems the percent variance explained by the first two principal components is often roughly the same. While these observations appear to show some promise for providing supporting data to use in problem diagnosis, the results are probably highly dependent upon the number of PV and OP time shifts incorporated into the data matrices. Much more study is needed to determine the number of time shifts to use and to determine the consistency of the results.

Based on the data sets and PCA methods studied, the PCA model fits can be used to identify hysteresis problems, and the PCA scores plot analyses can be used visually to provide supporting evidence for diagnosis of specific controller problems for many cases. Much more

study and analysis is needed to quantify the reliability of the techniques and to fully characterize the scores plot patterns for different controller problems.

CHAPTER 7

APPLICATION OF SELECTED STATISTICS IN DIAGNOSIS OF COMMON CONTROLLER PROBLEMS

7.1 Motivation for Application of Statistics

The four common control loop problems (valve stiction, valve hysteresis, poor tuning, and disturbances) often display unique PV versus OP patterns, unique controller error patterns, etc. Simple statistics calculated on a set of SP, PV, and OP data for a given control loop could be very useful for diagnosing specific control loop problems. Statistics are needed which can infer the dominant and unique patterns present in the data.

7.2 Error and Oscillation Frequency Statistics

Visual inspection of the simulated and real control loop data sets for valve stiction problems reveals that the controller PV moves or jumps very quickly once the resistance to valve movement is overcome, and that the PV often remains relatively constant (above and below the SP) for significant periods of time. As a result, the controller error for valve stiction problems is often distributed unusually, with the majority of the sampled controller errors occurring within a close proximity of the peak error for each half period oscillation. To capture these unique features, a statistic could be developed to calculate the ratio of the average error to the peak error for each

half period oscillation. To compensate for error amplitude changes and for positive/negative error variations, this statistic could be calculated for each half period oscillation and averaged separately for the positive and negative oscillations around the SP. Finally, converting the statistic to a percentage would give the following two diagnostic values: the average error/peak error % (for positive oscillations) and the average error/peak error % (for negative oscillations). For valve stiction problems, it would be expected that this average error/peak error statistic would be significantly higher (e.g., a value of 75 % or more) than the same statistic for control loops with other problems (e.g., a value of approximately 50 %).

The four common control loop problems often produce controller oscillations in different frequency regimes. For example, if the controller tuning is overaggressive, the induced oscillations will naturally be of high frequency, as the basic flow/pressure control loops exhibit negligible dead times and fast response times. The oscillations will be in the neighborhood of the ultimate frequency of the loop, which may often be between 5 and 20 seconds. As another example, if valve hysteresis or valve stiction is the problem, the oscillations will often (though there are certainly exceptions) be of lower frequency (on the order of minutes), due to the significant time periods in which the valve is in a stationary position. Thus, a statistic of the average half period oscillation frequency may be useful for diagnosing specific control loop problems.

A statistic of the average positive peak error/average negative peak error (expressed as a ratio) may be useful for problem diagnosis. Values of this statistic close to 1.0 suggest that repeatable oscillations are present in the data, while values far from 1.0 may suggest irregular or asymmetric patterns in the data. This statistic may be of utility in pointing to common problems (e.g., disturbances) more likely to produce irregular oscillations and variations around the controller SP.

In an analogous manner, a statistic of the average positive half period oscillation frequency/average negative half period oscillation frequency (expressed as a ratio) may be useful for problem diagnosis. Values of this statistic close to 1.0 suggest that regular, repeatable oscillations are present in the data, while values far from 1.0 may suggest irregular patterns in the data. Like the statistic for the ratio of the average positive peak error/average negative peak error, this statistic may be of utility in pointing to common problems (e.g., disturbances) more likely to produce irregular oscillations and variations around the controller SP.

Compilation of these error and oscillation frequency statistics for the simulated and real data sets was accomplished by programming the statistical calculations in a Matlab m-file (ERR_OSC_STATS.M, Appendix C). The results of the calculations are shown in Table 7.1.

Table 7.1. Average error and oscillation frequency statistics.

Loop ID	Problem	Avg. Error / Peak Error (+ Osc.) [%]	Avg. Error / Peak Error (- Osc.) [%]	Avg. + Peak Error to Avg. - Peak Error Ratio []	Avg. Half Per. Osc. Freq. [min]	Avg. + Osc. Freq. to Avg. - Osc. Freq. Ratio []
STIC1S	Stict.	87.53	87.92	0.88	2.180	1.14
STIC2S	Stict.	73.98	77.14	1.17	0.472	0.89
STIC3S	Stict.	91.02	90.71	0.94	2.180	1.06
STIC4S	Stict.	35.42	87.37	2.05	2.600	1.19
STIC5S	Stict.	49.78	46.90	1.11	1.850	1.27
STIC6S	Stict.	52.48	50.41	1.54	0.278	3.09
STIC1R	Stict.	76.03	79.09	0.53	12.350	2.04
STIC2R	Stict.	75.82	78.49	1.46	0.561	0.72
STIC3R	Stict.	75.82	72.87	0.88	5.730	0.90
STIC4R	Stict.	70.04	67.02	1.03	1.510	0.97
HYST1S	Hyst.	39.07	39.93	0.98	5.790	1.02
HYST2S	Hyst.	25.74	34.78	1.36	2.600	1.67
HYST3S	Hyst.	26.97	22.32	0.89	9.710	1.15
HYST4S	Hyst.	37.87	38.56	1.00	5.880	0.99
HYST5S	Hyst.	62.80	55.95	0.81	4.020	0.75
HYST1R	Hyst.	19.01	20.29	0.90	11.870	1.41
HYST2R	Hyst.	43.21	43.53	1.10	1.530	1.11
HYST3R	Hyst.	24.02	24.19	1.08	9.770	0.77
TUNE1S	Tuning	52.68	50.77	0.99	0.055	1.00
TUNE2S	Tuning	62.35	62.21	0.98	0.107	1.02
TUNE3S	Tuning	57.60	55.62	1.01	0.039	0.99
TUNE4S	Tuning	63.20	63.58	1.00	0.352	1.01
TUNE5S	Tuning	52.04	49.48	0.99	0.274	0.92
TUNE1R	Tuning	72.43	69.51	0.96	0.092	1.01
DIST1S	Dist.	54.53	54.66	1.02	4.770	1.02
DIST2S	Dist.	61.11	65.92	1.01	0.125	1.02
DIST3S	Dist.	47.16	48.26	1.00	4.410	1.10
DIST4S	Dist.	58.25	58.74	1.02	1.980	0.97
DIST5S	Dist.	27.50	22.93	0.85	2.170	1.01
DIST6S	Dist.	33.94	24.32	0.88	2.320	0.87
DIST1R	Dist.	57.25	54.89	1.07	1.570	0.90
DIST2R	Dist.	60.50	74.27	2.25	0.569	0.56
GOOD1S	None	n/a	n/a	n/a	n/a	n/a
GOOD2S	None	n/a	n/a	n/a	n/a	n/a
GOOD1R	None	n/a	n/a	n/a	n/a	n/a
GOOD2R	None	n/a	n/a	n/a	n/a	n/a

7.3 PV/OP and OP/PV Spread Statistics

Visual inspection of PV versus OP data for the 4 common problems reveals that for many cases, widely ranging controller PV values are found at virtually the same value of controller OP (i.e., there are "holes" and "gaps" within the "shapes" defined by the PV versus OP data points). Likewise, for many cases, significantly different controller OP values may result in virtually the same controller PV value. Given these observations, a statistic which estimates the spread in the PV values for a small range of OP values, as well as a statistic which estimates the spread in the OP values for a small range of PV values, may be useful in problem diagnosis. The statistic for estimating the spread in the PV values (for small ranges of OP movement) will be labeled the PV/OP spread statistic. The comparable statistic for estimating the spread in the OP values will be labeled the OP/PV spread statistic.

For stiction and hysteresis problems, it would be expected that the OP/PV spread statistic and the PV/OP spread statistic would be relatively high, due to the long periods of stagnant valve position in concert with significant controller OP movement. For tuning and disturbance problems, it would be expected that the spread statistics would be somewhat lower (possibly significantly lower for many tuning problems).

To calculate the PV/OP spread statistic, the OP data for a given controller problem will be sectioned into a specified number of sections (e.g., 20) based on OP magnitude (sections of size equal to

the difference in the minimum and maximum OP for the entire data set divided by the number of desired sections). For each OP section (representing 5 % of the total OP movement if there are 20 sections), the minimum and maximum controller PV will be found. The difference in the minimum and maximum PV (for that OP section) will be found and divided by the overall PV movement for the entire data set (to get a % PV change relative to the low/high PVs for the entire data set). The ratio of the % PV change to the % OP change for that section will then be calculated, and the similar values for each of the OP sections will be averaged. This averaged value will then be normalized by the number of sections used to get a percentage between 0 and 100. The OP/PV spread statistic will be calculated in an analogous fashion.

Compilation of these spread statistics for the simulated and real data sets was accomplished by programming the statistical calculations in a Matlab m-file (PV_OP_SPREAD_STATS.M, Appendix C). The number of sections used for both the PV/OP and OP/PV calculations was 20. The results of the calculations are shown in Table 7.2.

7.4 Usefulness of the Selected Statistics for Problem Diagnosis

Based on the calculated statistics for all of the simulated and real data sets containing the 4 control loop problems, several inferences can be made concerning the use/utility of the selected statistics for problem diagnosis. The following inferences are listed for each category of statistic.

Table 7.2. PV/OP and OP/PV spread statistics.

Loop ID	Problem	PV/OP Spread [%]	OP/PV Spread [%]
STIC1S	Stict.	28.1	74.8
STIC2S	Stict.	19.8	39.1
STIC3S	Stict.	26.5	76.4
STIC4S	Stict.	18.0	83.5
STIC5S	Stict.	3.9	11.5
STIC6S	Stict.	25.9	18.8
STIC1R	Stict.	88.5	78.1
STIC2R	Stict.	49.3	66.7
STIC3R	Stict.	41.6	23.7
STIC4R	Stict.	46.4	30.0
HYST1S	Hyst.	21.7	56.8
HYST2S	Hyst.	18.4	57.2
HYST3S	Hyst.	18.5	52.9
HYST4S	Hyst.	19.6	41.8
HYST5S	Hyst.	17.5	15.5
HYST1R	Hyst.	14.2	60.9
HYST2R	Hyst.	11.9	35.1
HYST3R	Hyst.	7.6	42.0
TUNE1S	Tuning	9.6	20.8
TUNE2S	Tuning	23.9	7.0
TUNE3S	Tuning	4.6	4.6
TUNE4S	Tuning	36.8	40.1
TUNE5S	Tuning	17.7	3.2
TUNE1R	Tuning	8.8	6.2
DIST1S	Dist.	45.3	32.1
DIST2S	Dist.	9.1	25.9
DIST3S	Dist.	47.3	32.9
DIST4S	Dist.	42.2	35.7
DIST5S	Dist.	7.8	3.3
DIST6S	Dist.	6.8	7.8
DIST1R	Dist.	38.3	45.4
DIST2R	Dist.	57.4	52.2
GOOD1S	None	6.7	4.9
GOOD2S	None	14.2	9.0
GOOD1R	None	17.2	1.5
GOOD2R	None	10.5	16.7

For the error statistics (Table 7.1), the following observations are made: 1) stiction problems typically show a large value for the average error/peak error (e.g., normally 70-90% for the typical stiction problems, note that loops STIC5S and STIC6S are atypical; STIC5S contains widely ranging mean shifts while STIC6S has only a slight amount of stiction), 2) hysteresis problems typically show low values for the average error/peak error (e.g., normally 20-40%, note that HYST5S is atypical in that it has only a slight amount of hysteresis), 3) tuning problems typically show a mid-range value for the average error/peak error (e.g., normally 50-60%, note that TUNE1R would probably have an average error/peak error closer to 60 % if the sampling frequency for the loop was increased), 4) disturbance problems typically show a low to mid-range value for the average error/peak error (e.g., normally 20-60%), and 5) tuning problems typically show a value near 1.0 for the average positive peak error / average negative peak error statistic, while the other problems show significant variability around 1.0 (note that this statistic signifies the varying degrees of stiction present in loop STIC4S by a relatively high value of 2.05).

For the oscillation frequency statistics (Table 7.1), the following observations are made: 1) tuning problems typically show a very high oscillation frequency (e.g., most are less than 15 seconds), 2) stiction and hysteresis problems typically show a low oscillation frequency (e.g., typically on the order of several minutes, STIC2S and STIC2R are exceptions), 3) disturbance problems typically show a mid-range oscillation frequency, and 4) the average positive oscillation

frequency /average negative oscillation frequency statistic is typically near 1.0 for tuning problems and can be highly variable around 1.0 for the other 3 common problems.

For the PV/OP and OP/PV spread statistics (Table 7.2), the following observations are made: 1) on average, stiction and hysteresis problems have the highest values of PV/OP spread, tuning problems have the lowest values, and disturbance problems show highly variable values, and 2) the differences in the two spread statistics for a given loop appear to be much more pronounced for the stiction and hysteresis problems when compared to the differences for the disturbance and tuning problems.

While no one statistic comes close to accurately diagnosing all of the 4 common problems, several of the statistics show promise for helping to diagnose 1 or 2 of the common problems. In summary, if a loop shows an average error/peak error statistic of greater than 75 %, the loop probably has a valve stiction problem. If a loop has an average oscillation frequency of less than 20 seconds, the problem is definitely not stiction or hysteresis, and the problem is probably tuning (although a high frequency disturbance could also be the problem). If a loop has a very high PV/OP or OP/PV spread statistic, the problem is definitely not tuning, and the problem is most likely hysteresis or stiction. Thus, the average error/peak error statistic, the average oscillation frequency, and the PV/OP and OP/PV spread statistics all give some definite value in diagnosing the 4 common problems. Refinement of some of these statistics or application of

other statistics may further enhance the utility of using descriptive statistics to diagnose the controller problems of interest.

7.5 Problem Diagnosis Through Combination of Cross Correlation, PCA, and Descriptive Statistics

For purposes of summarizing an overall problem diagnosis procedure, a flowchart has been developed (Figure 7.1). This flowchart outlines logic which could be used to diagnose a specific controller problem as one of the 4 common problems, given that a poorly-performing control loop has been identified. This logic is based entirely on quantitative inferences generated from the problem diagnosis studies covered in the scope of this work. Thus, it may be of limited accuracy, as additional testing and research is needed to prove the consistency and reliability of the outlined diagnosis techniques. Nevertheless, this logic can be a valuable controller troubleshooting aid by helping to positively identify the problem for many cases and by helping to positively rule out potential problem causes for other cases. Note: PCA methods and descriptive statistics which are not referenced in Figure 7.1 may provide supporting evidence for the diagnosis of disturbance, tuning, and stiction problems. PCA can also be coupled with cross correlation analysis as a positive test for hysteresis.

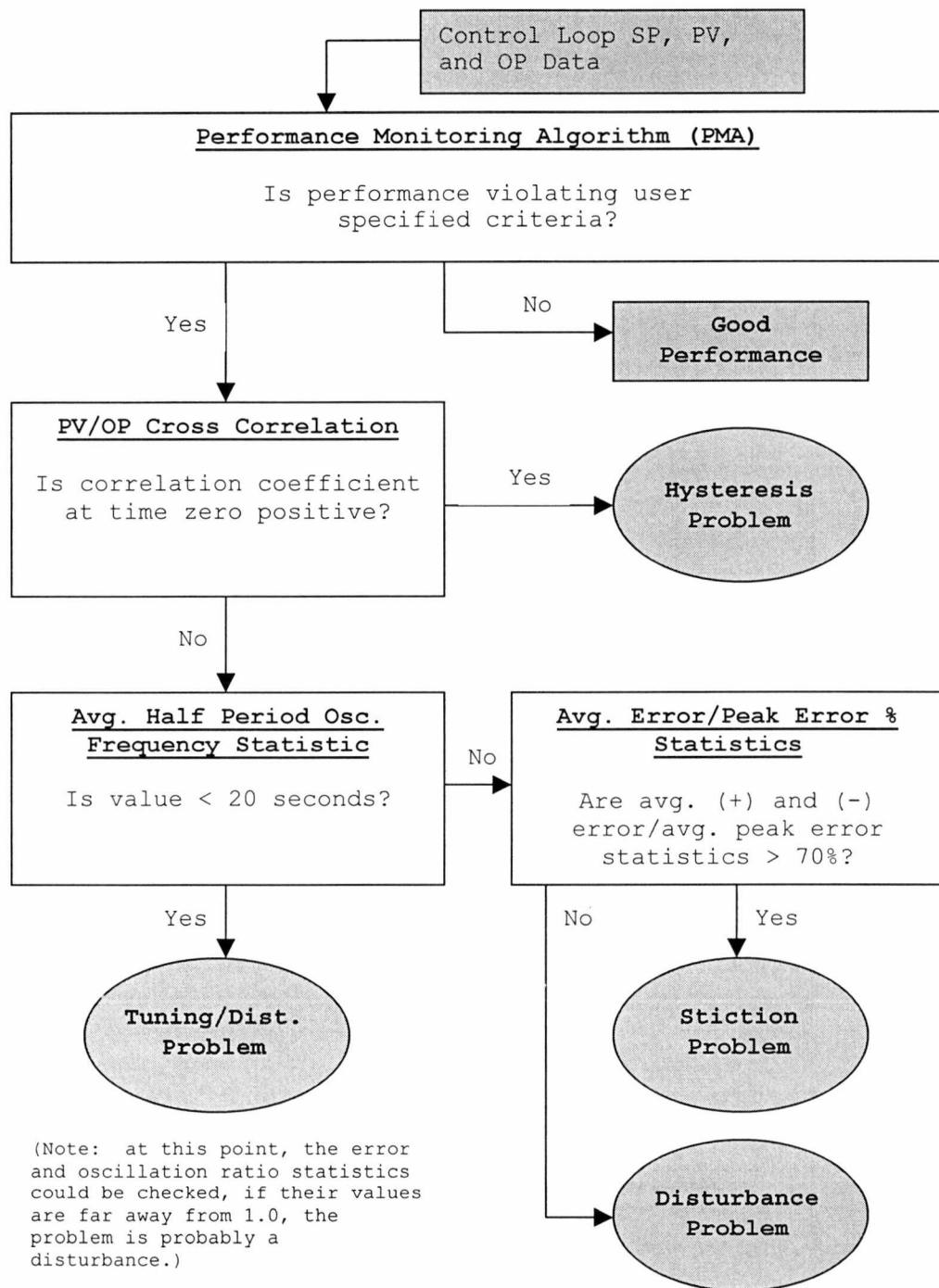


Figure 7.1. Summary flowchart for problem diagnosis logic.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE STUDY

8.1 Conclusions

In summary, simulated data sets exhibiting stiction, hysteresis, disturbance, and tuning problems have been developed using a dynamic simulator. Analogous real data sets have been obtained from chemical processes at Eastman Chemical. Using the data, a general Performance Monitoring Algorithm (PMA) has been developed for detecting PID control loops which are performing poorly based on user specified performance criteria. To identify methods which may be useful for diagnosing the 4 controller problems, cross correlation analysis, PCA, and developed descriptive statistics have been applied to the simulated and real data sets. The monitoring and diagnosis work performed is specific to PID flow/pressure controllers with a performance objective of fast setpoint tracking with no steady-state offset (i.e., controllers which typically oscillate around the setpoint when not performing well). A discussion of the results of these efforts follows.

For the monitoring algorithm portion of this work, the developed PMA is successful in detecting poor control loop performance in real time. The PMA alarming ability is independent of the controller oscillation frequency and the type of problem inducing the poor performance. The PMA is a robust detection algorithm and is easily tuned to prevent

false performance alarms. In addition, the design of the PMA provides a framework for application to large numbers (all types) of PID loops, but much more work is needed to develop default user parameters for various types of loops.

To summarize the problem diagnosis portion of this work, when the cross correlation, PCA, and descriptive statistics are used in concert, the most probable controller problem can be diagnosed with a limited degree of certainty for all cases except high frequency controller oscillations. Where a high frequency oscillation is present, no diagnosis method was found capable of differentiating between tuning and disturbance problems. More work is needed in this area to make the diagnosis ability more robust and reliable for all of the controller problems, as well as to develop methods which can differentiate between disturbance and tuning problems for high frequency oscillations.

Cross correlation analysis is found to be of limited utility in diagnosing the 4 specific controller problems. Cross correlation does provide diagnosis ability for hysteresis problems, and can help to rule out potential problem causes for other cases. In a similar fashion, the PCA model fits are useful for positively identifying hysteresis problems, and the PCA scores plot analyses can visually provide valuable supporting evidence for diagnosis of controller problems for many cases. Descriptive error and oscillation frequency statistics are demonstrated to be useful for identifying stiction

problems. These statistics are also of utility in helping to exclude specific problems as the potential cause of poor performance.

8.2 Recommendations for Future Study

In relation to the Performance Monitoring Algorithm (PMA), there are several pertinent issues requiring additional study. Although the PMA framework and design should be directly applicable to other major types of PID control loops (in addition to the flow/pressure control loops considered here), much work is needed to define good default user inputs for the other major types of PID control loops. Perhaps default user inputs could be developed for different control loop groupings or classes of control loops. For example, one class of loops might be level control loops which control large inventories and are meant for process surge capacity and disturbance dampening. For these loops, longer time window values (e.g., T_{mon} equal to 24 hours) and broader oscillation limits (e.g., PV_{lim} equal to 10.0 %) would be desirable. Classes of loops could be developed based on controlled variable type, the need for tight control, etc.

While the suggested alarm setting for the PMA is the percent time oscillating, PMA alarming could be enhanced by making it a combination of the percent time oscillating and the calculated average peak deviation. An alarming "region" could be used, with the percent time oscillating alarm limit P_{alm} being decreased from its default value as the calculated average peak deviation rises significantly (i.e., the PMA alarm limit could be made a function of the average peak

deviation, with the base point for the function being the default user inputs for P_{aim} and PV_{lim} values). For example, if the default user inputs are $P_{aim} = 50.0 \%$ and the PV_{lim} limit value is 1.0% , it might be advantageous for the P_{aim} limit to decrease to 20-30 % if the calculated average peak deviation rises above 5.0% .

The focus of the PMA is on monitoring for oscillations of the controller PV around the SP. As an extension of this, an algorithm to monitor for significant controller OP oscillations is needed. For some PID control loops, significant controller OP oscillations may be present and be of concern, even when the controller PV is not oscillating significantly. To accomplish this, robust filtering algorithms are needed for calculating controller OP reference values (so that deviations from the target or reference value can be calculated).

To make the PMA completely general and applicable to continuous chemical processes, the PMA could be extended to monitor for sustained oscillations for any process variable (i.e., adapt the PMA to monitor for all variables, not just for PID control loops). To accomplish this, robust filtering algorithms are needed for calculating process variable reference values (moving process mean values), so that deviations of the actual process value from the target can be calculated.

To further investigate and improve the controller problem diagnosis techniques, significantly more work is needed. To begin, more work is

needed to improve and quantify the diagnosis ability of the data analysis techniques and descriptive statistics. While the outlined techniques/statistics can determine the most probable controller problem for most of the cases studied, the extent of their reliability has not been determined and they may possibly give erroneous results for some cases. As a further concern, because only 4 common controller problems were studied, some additional work is necessary to determine if there are other potential controller problems which could masquerade as one of the 4 common problems.

As discussed previously, the diagnosis techniques and statistics are unable to differentiate between tuning and disturbance problems at high oscillation frequencies. As a result, additional diagnostic tools are needed for the high frequency oscillation regime.

Automation of the application of the problem diagnosis techniques and descriptive statistics would be of significant benefit. While automation of the PMA is relatively straightforward, development of an automated problem diagnosis algorithm is not. Selection of data window sizes, data sample times, and analysis input parameters are just a few of the concerns related to automating a problem diagnosis algorithm.

In relation to the PCA diagnosis methods, more study is needed to fully characterize the PCA scores plot patterns for the 4 controller problems. In addition, PCA methods may be more useful which use PCA models for individual control loops and monitor how those models

change over time. The use of multi-way PCA, which accounts for the ordered nature of the process data, may also be of utility in problem diagnosis.

A major challenge is to develop diagnosis techniques which would be applicable to all the major types of PID control loops. This would be a complicated task, as the response characteristics of the other types of PID control loops (not considered in this work) are vastly different in dead time, response time, and response order. As a starting point for addressing this challenge, recent applications by Thornhill, Oettinger, and Fedenczuk (1998) as well as by Thornhill and Hagglund (1997) propose the use of power spectrum analysis and closed loop impulse response analysis to aid in the diagnosis of controller tuning and controller valve problems.

The presence of significant PV filtering, not uncommon in industrial PID control loops, can produce misleading problem diagnosis results if the filtered PV data is unknowingly used. Formal methods are needed to address PV filtering (e.g., for what analyses should the filtered value be used?, for what analyses should the raw, unfiltered value be used?, etc.).

Finally, as a means of testing and improving both the PMA and the diagnosis techniques, a case study on all the PID control loops in a given chemical process should be performed. The case study would point out shortcomings of the algorithms and techniques and would help determine the areas in which further study is most needed.

LIST OF REFERENCES

- Belanger, P.W. and W.L. Luyben, *A New Test for Evaluation of the Regulatory Performance of Controlled Processes*, Industrial & Engineering Chemistry Research, 1996. 35: p.3447-3457.
- Desborough, L.D. and T.J. Harris, *Performance Assessment Measures for Univariate Feedback Control*. The Canadian Journal of Chemical Engineering, 1992. 70: p.1186-1197.
- Desborough, L.D. and T.J. Harris, *Performance Assessment Measures for Univariate Feedforward/Feedback Control*. The Canadian Journal of Chemical Engineering, 1993. 71: p.605-616.
- Ender, D.B., *Process Control Performance: Not as Good as You Think*. Control Engineering, 1993. 40(10): p.180-190.
- Hackney, J.E., *Cultivation of Model Predictive Control Systems*. 1989, University of Tennessee:
- Hagglund, T., *A Control Loop Performance Monitor*. Control Engineering Practice, 1995. 3(11): p.1543-1551.
- Harris, T.J., *Assessment of Control Loop Performance*. The Canadian Journal of Chemical Engineering, 1989. 67: p.856-861.
- Hunter, J.S., T.J. Harris and J.F. MacGregor, *SPC Interfaces -- Engineering and Statistical Process Control*. presented at Eastman Chemical Company, Kingsport, TN, June 28-30, 1993.
- Kozub, D.J. and C.E. Garcia, *Monitoring and Diagnosis of Automated Controllers in the Chemical Process Industries*. presented at AIChE Meeting, St. Louis, MO, November 9, 1993.
- Kozub, D.J., *Monitoring and Diagnosis of Chemical Processes with Automated Process Control*. presented at CPC V Meeting, Lake Tahoe, NV, January, 1996.
- Krauss, T.P., L. Shure and J.N. Little, *Signal Processing Toolbox User's Guide*. 1994, Natick, MA: The MathWorks, Inc.
- Kresta, J.V., J.F. MacGregor and T.E. Marlin, *Multivariate Statistical Monitoring of Process Operating Performance*. The Canadian Journal of Chemical Engineering, 1991. 69: p.35-47.
- Liptak, B., *Instrument Engineers' Handbook*. Rev. ed. 1985, Radnor, PA: Chilton Book Company.
- Liptak, B., *The Weakest Link in the Control Loop*. Control, 1998. p.21-22.
- Luyben, W.L., *Process Modeling, Simulation, and Control for Chemical Engineers*. 2nd ed. 1990, New York: McGraw-Hill, Inc.
- MacGregor, J.F. and T.J. Harris, *The Exponentially Weighted Moving Variance*. Journal of Quality Technology, 1993. 25(2): p.106-118.

MacGregor, J.F. and P.A. Taylor, *Advanced Process Control*. presented at Venture Inn, Burlington, Ontario, May 1994.

McMillan, G.K., *Tuning and Control Loop Performance*. 1st ed. 1983, Research Triangle Park, NC: Instrument Society of America.

Miller, H.D., *Application of Partial Least Squares and Principal Components Analysis to Control and Monitoring of a Reactive Distillation Column*. 1994, University of Tennessee:

Perrier, M. and A.A. Roche, *Towards Mill-Wide Evaluation of Control Loop Performance*. Control Systems, 1992. p.205-209.

Seborg, D.E., T.F. Edgar and D.A. Mellichamp, *Process Dynamics and Control*. 1st ed. 1989, New York: John Wiley & Sons.

Shinskey, F.G., *Putting Controllers to the Test*. Chemical Engineering, 1990. **97**: p.96-106.

Stanfelj, N., T.E. Marlin and J.F. MacGregor, *Monitoring and Diagnosing Process Control Performance: The Single-Loop Case*. Industrial & Engineering Chemistry Research, 1993. **32**(2): p. 301-314.

Thornhill, N.F. and T. Hagglund, *Detection and Diagnosis of Oscillation in Control Loops*. Control Engineering Practice, 1997. **5**: p.1343-1354.

Thornhill, N.F., M. Oettinger and P. Fedenczuk, *Performance Assessment and Diagnosis of Refinery Control Loops*. presented at FOCAPO Meeting, Snowbird, UT, July 1998.

Tyler, M.L. and M. Morari, *Performance Monitoring of Control Systems using Likelihood Methods*. Automatica, 1996. **32**(8): p.1145-1162.

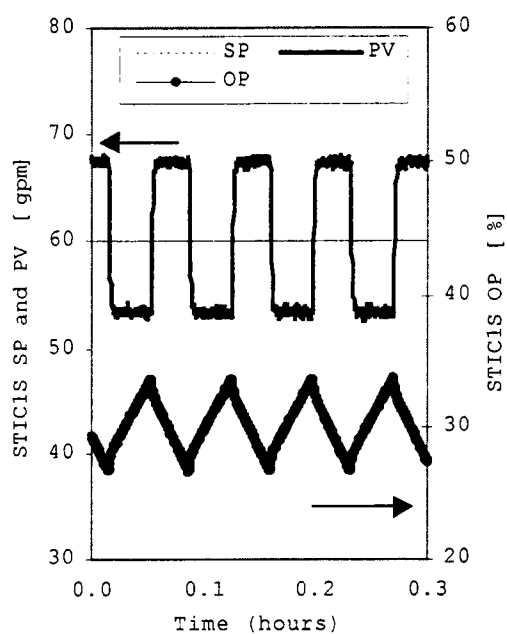
Wise, B.M. and N.B. Gallagher, *The Process Chemometrics Approach to Process Monitoring and Fault Detection*. Journal of Process Control, 1996. **6**(6): p.329-348.

APPENDICES

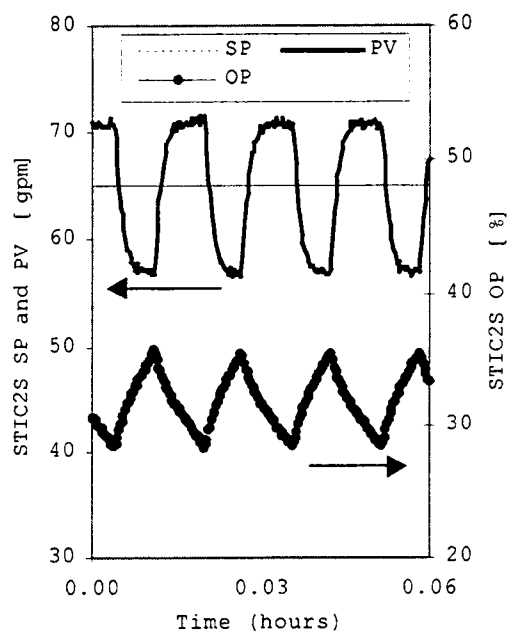
APPENDIX A

SIMULATED AND REAL PROCESS DATA SETS

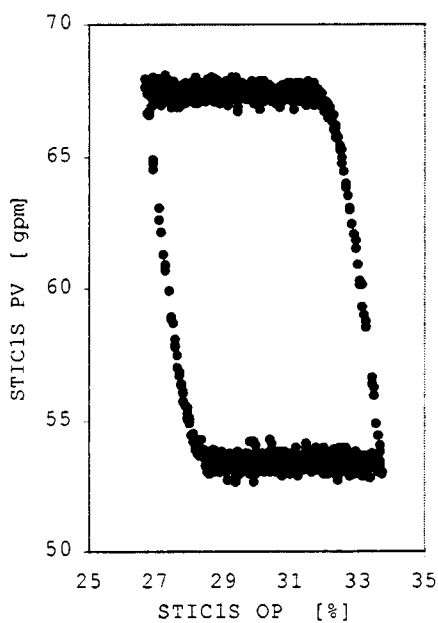
Appendix A contains SP, PV, and OP data plots for all of the simulated and real process data sets. For purposes of clarity, only portions of the data sets are plotted. The complete SP, PV, and OP data set for each simulated and real control loop is available in electronic Microsoft Excel spreadsheet format (see Dr. Charles F. Moore for access to this data). The electronic copies of the data are in individual Microsoft Excel workbook files named *loopid.xls*, where *loopid* is the tag identifier for the specific control loop (e.g., data for simulated tag STIC1S is located in file STIC1S.xls).



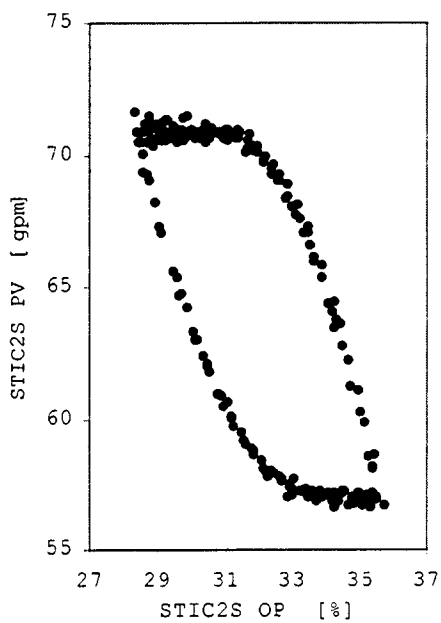
(a)



(b)



(c)



(d)

Figure A.1. Simulated data for valve stiction problem (STIC1S and STIC2S): (a) SP/PV/OP data for STIC1S, (b) SP/PV/OP data for STIC2S, (c) PV/OP characteristic plot for STIC1S, (d) PV/OP characteristic plot for STIC2S.

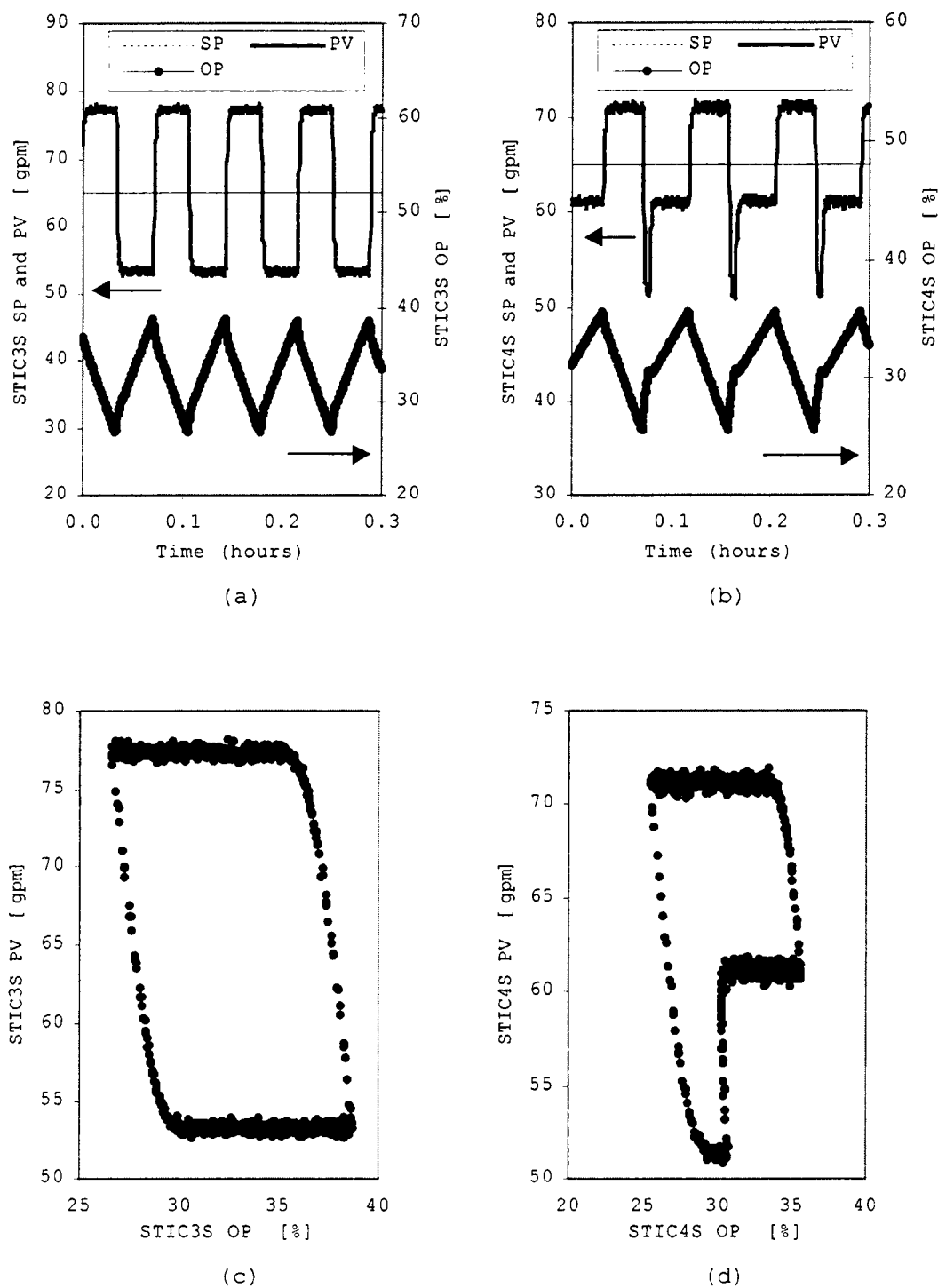


Figure A.2. Simulated data for valve stiction problem (STIC3S and STIC4S): (a) SP/PV/OP data for STIC3S, (b) SP/PV/OP data for STIC4S, (c) PV/OP characteristic plot for STIC3S, (d) PV/OP characteristic plot for STIC4S.

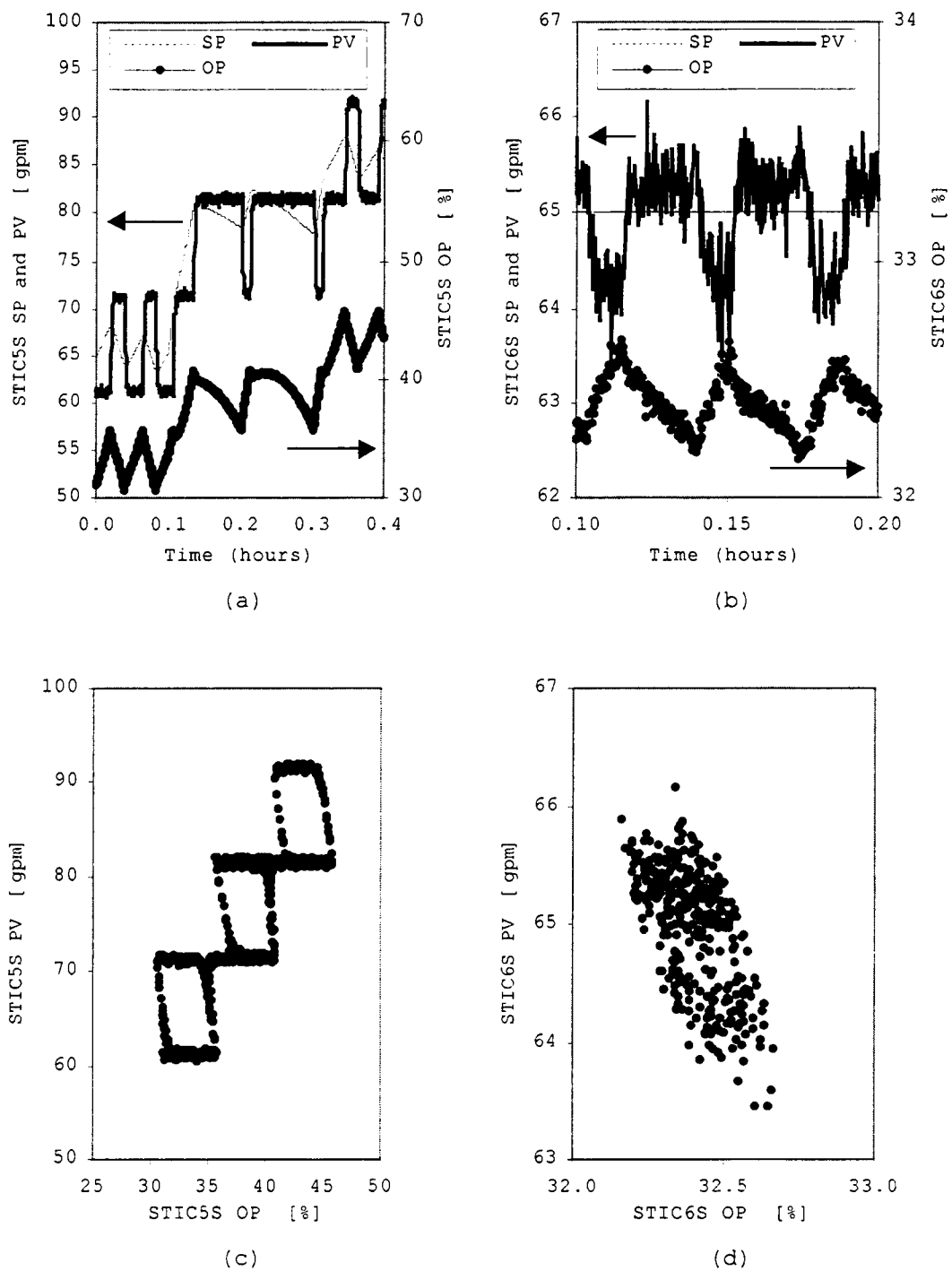


Figure A.3. Simulated data for valve stiction problem (STIC5S and STIC6S): (a) SP/PV/OP data for STIC5S, (b) SP/PV/OP data for STIC6S, (c) PV/OP characteristic plot for STIC5S, (d) PV/OP characteristic plot for STIC6S.

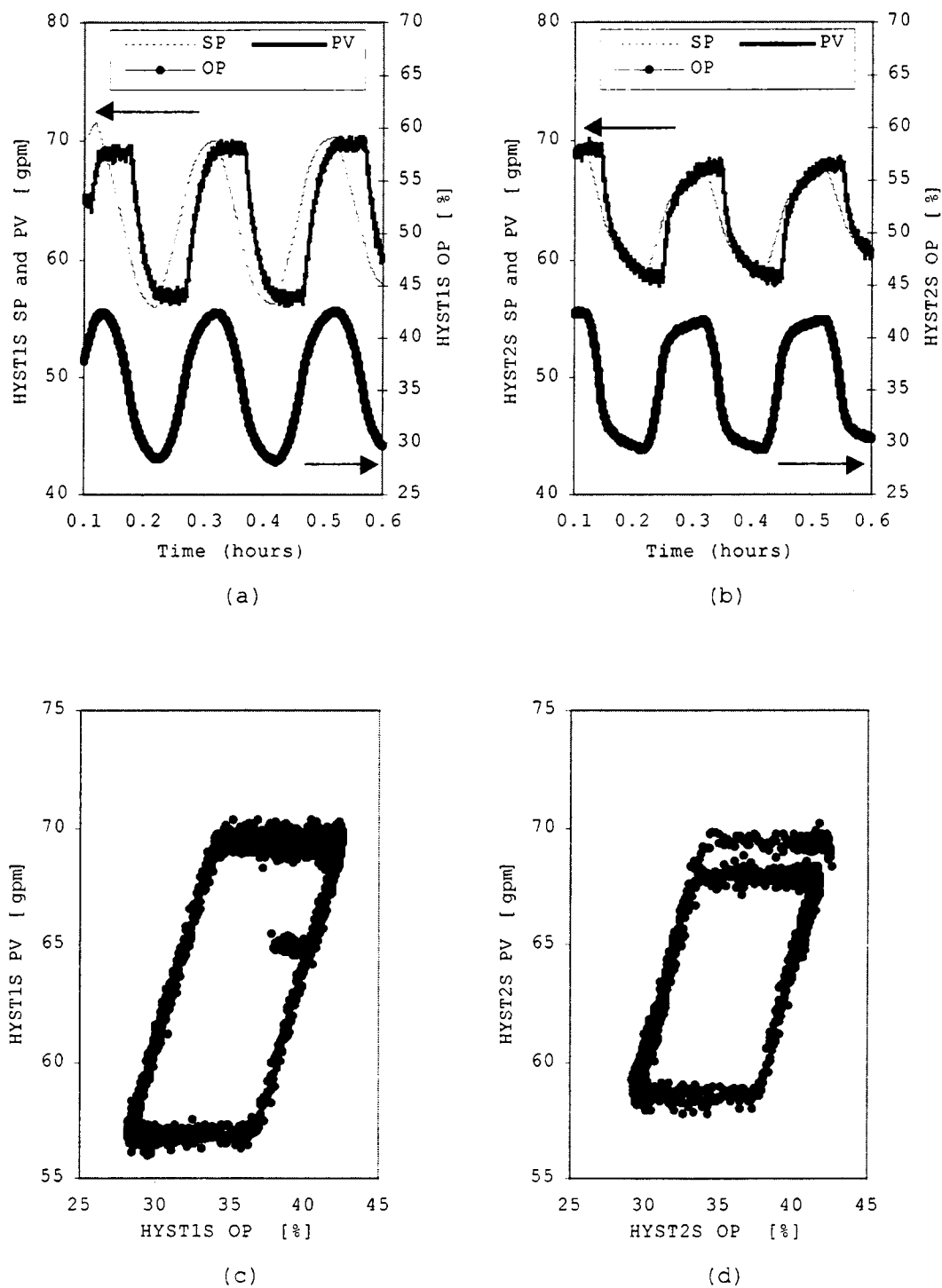
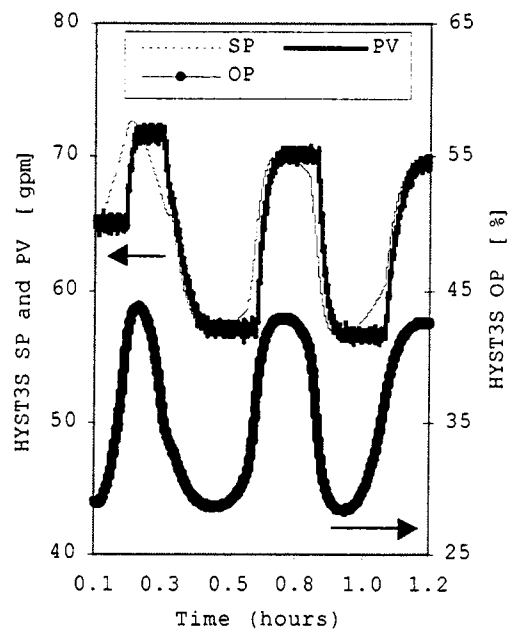
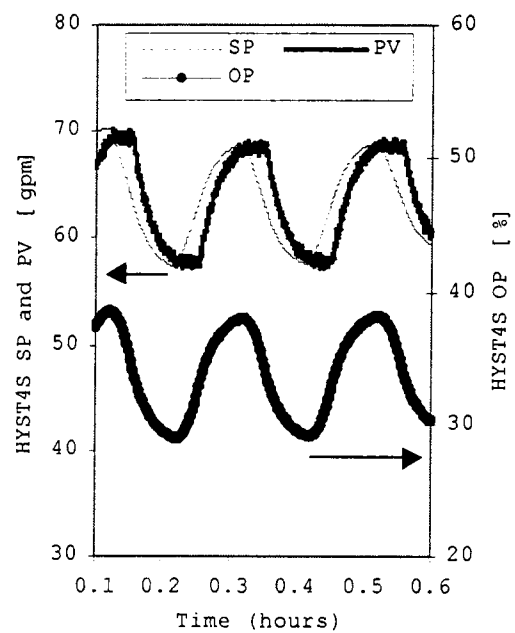


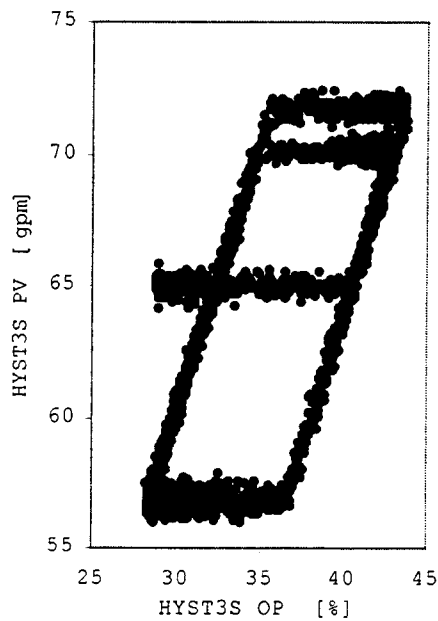
Figure A.4. Simulated data for valve hysteresis problem (HYST1S and HYST2S): (a) SP/PV/OP data for HYST1S, (b) SP/PV/OP data for HYST2S, (c) PV/OP characteristic plot for HYST1S, (d) PV/OP characteristic plot for HYST2S.



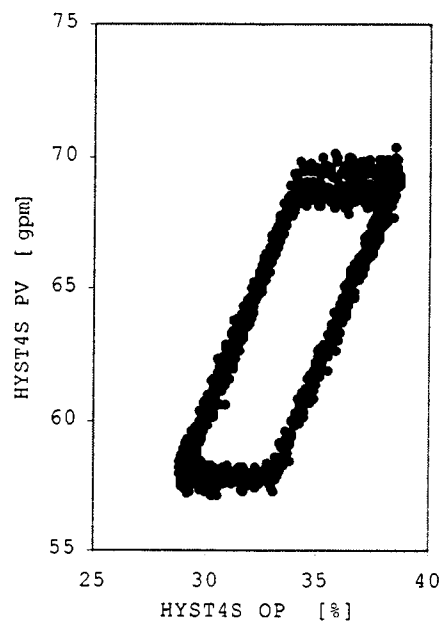
(a)



(b)

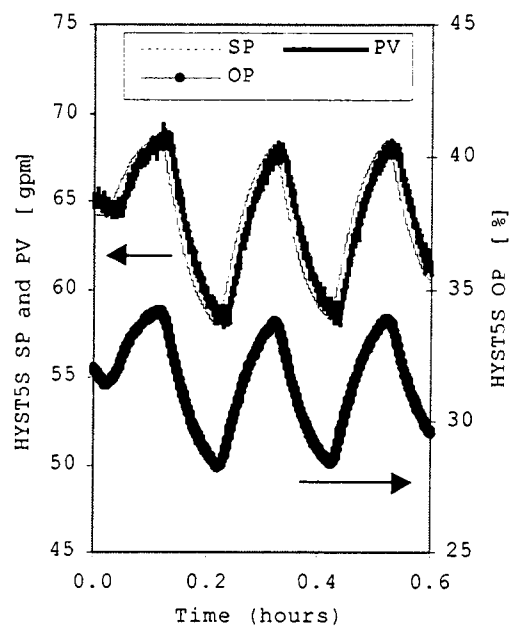


(c)

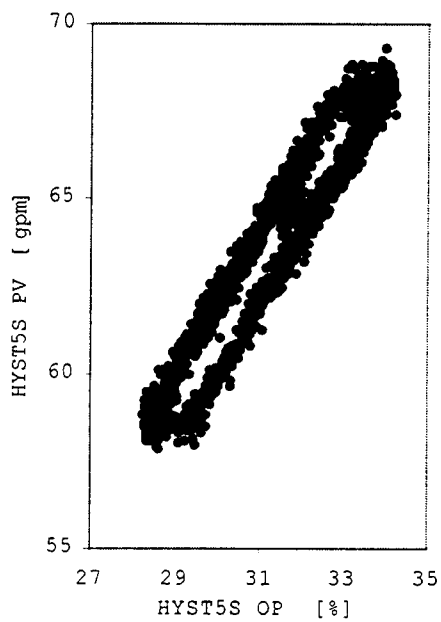


(d)

Figure A.5. Simulated data for valve hysteresis problem (HYST3S and HYST4S): (a) SP/PV/OP data for HYST3S, (b) SP/PV/OP data for HYST4S, (c) PV/OP characteristic plot for HYST3S, (d) PV/OP characteristic plot for HYST4S.

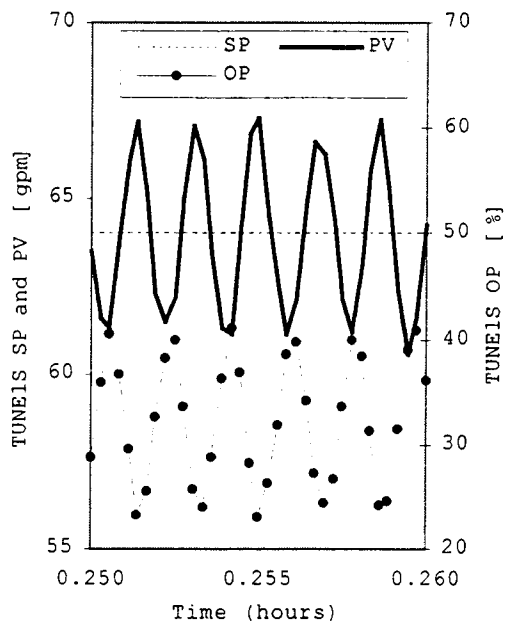


(a)

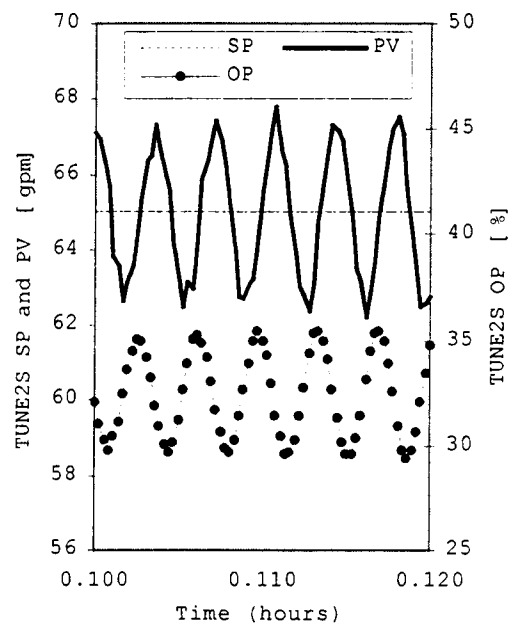


(b)

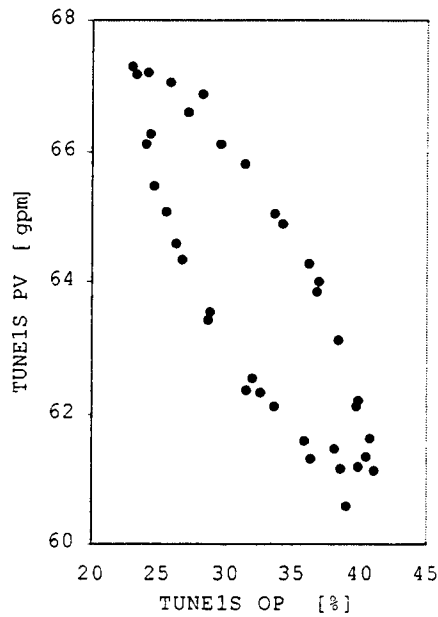
Figure A.6. Simulated data for valve hysteresis problem (HYST5S):
 (a) SP/PV/OP data for HYST5S, (b) PV/OP characteristic plot for
 HYST5S.



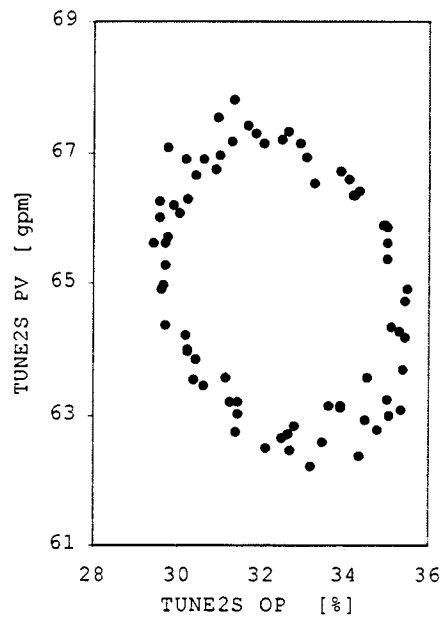
(a)



(b)

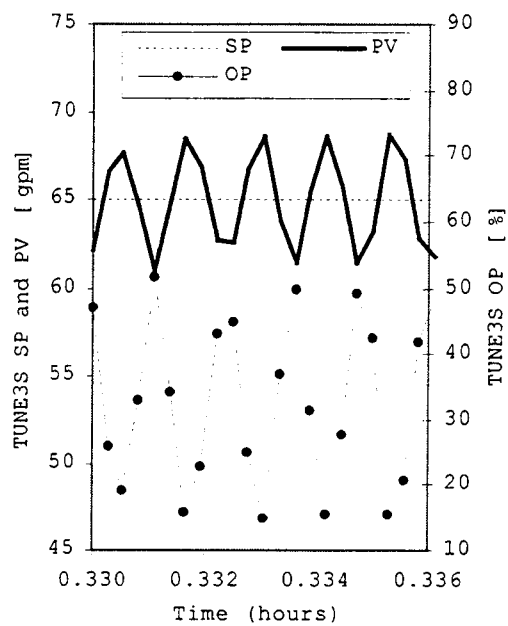


(c)

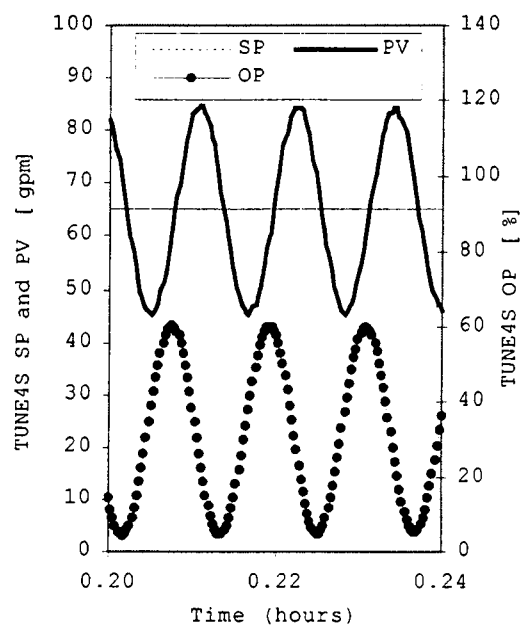


(d)

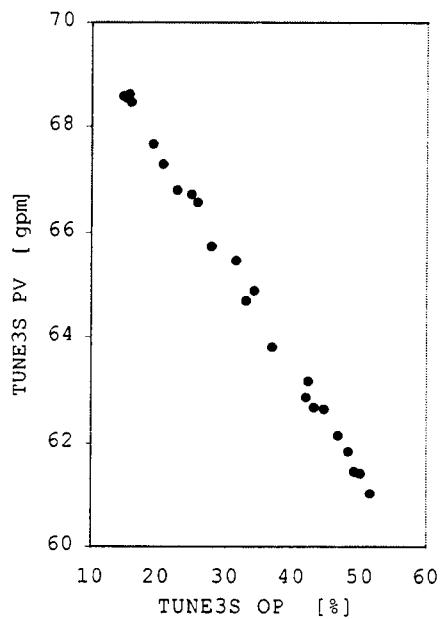
Figure A.7. Simulated data for tuning problem (TUNE1S and TUNE2S): (a) SP/PV/OP data for TUNE1S, (b) SP/PV/OP data for TUNE2S, (c) PV/OP characteristic plot for TUNE1S, (d) PV/OP characteristic plot for TUNE2S.



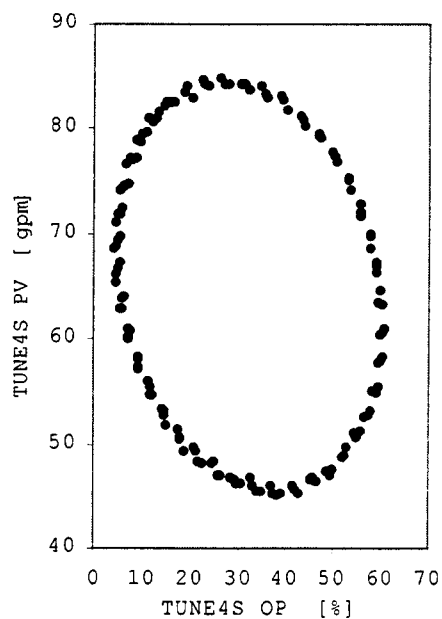
(a)



(b)

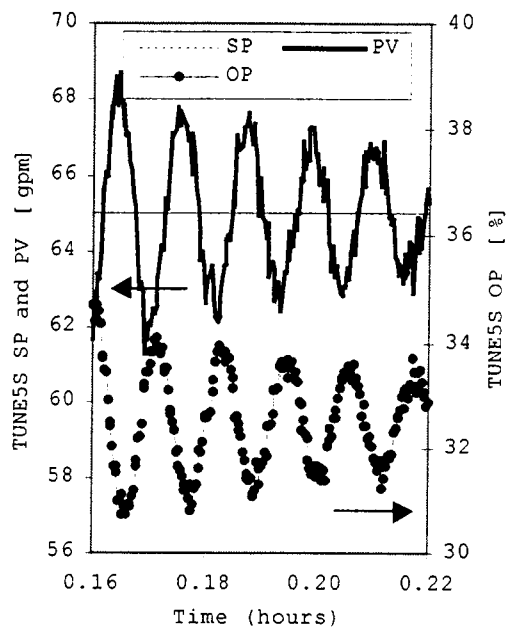


(c)

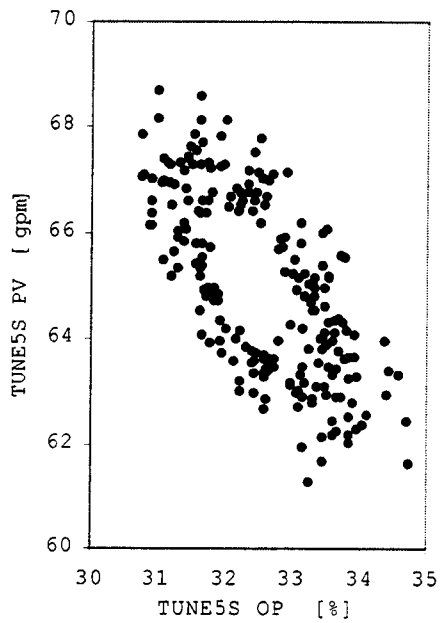


(d)

Figure A.8. Simulated data for tuning problem (TUNE3S and TUNE4S): (a) SP/PV/OP data for TUNE3S, (b) SP/PV/OP data for TUNE4S, (c) PV/OP characteristic plot for TUNE3S, (d) PV/OP characteristic plot for TUNE4S.

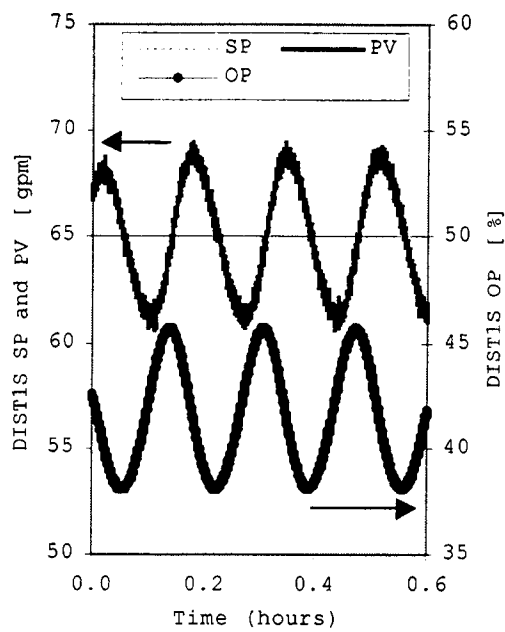


(a)

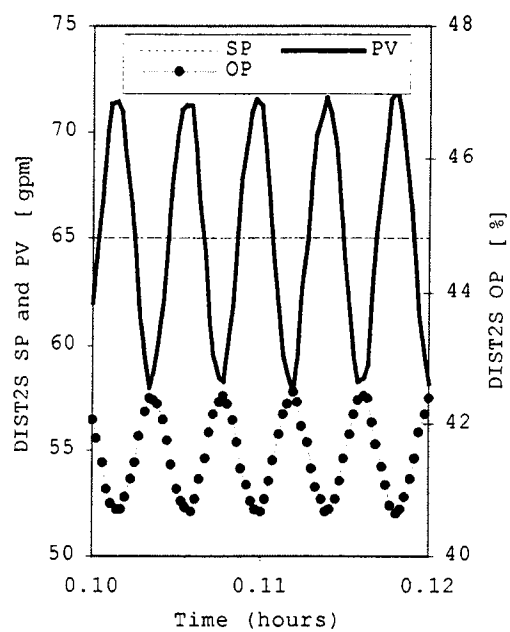


(b)

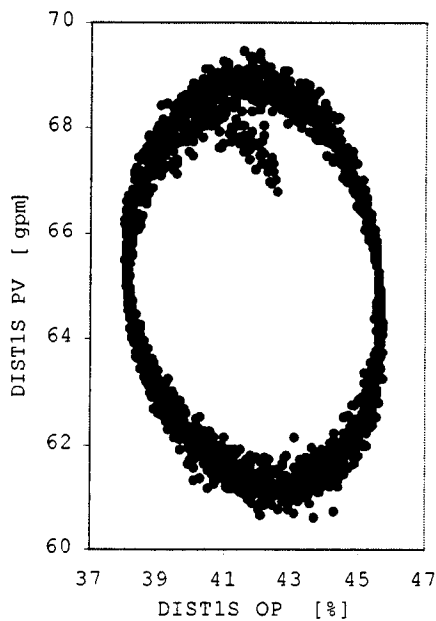
Figure A.9. Simulated data for tuning problem (TUNE5S): (a) SP/PV/OP data for TUNE5S, (b) PV/OP characteristic plot for TUNE5S.



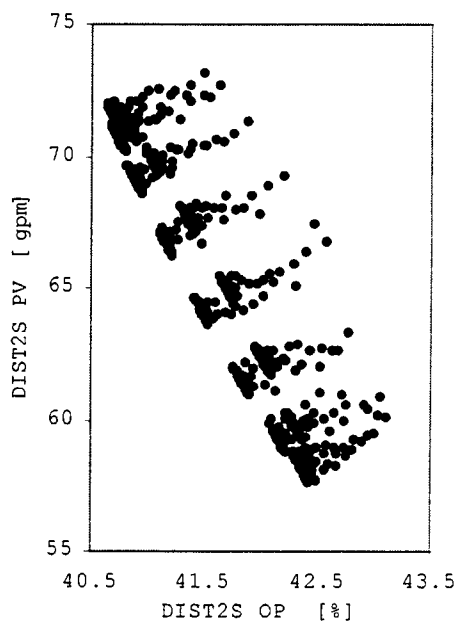
(a)



(b)

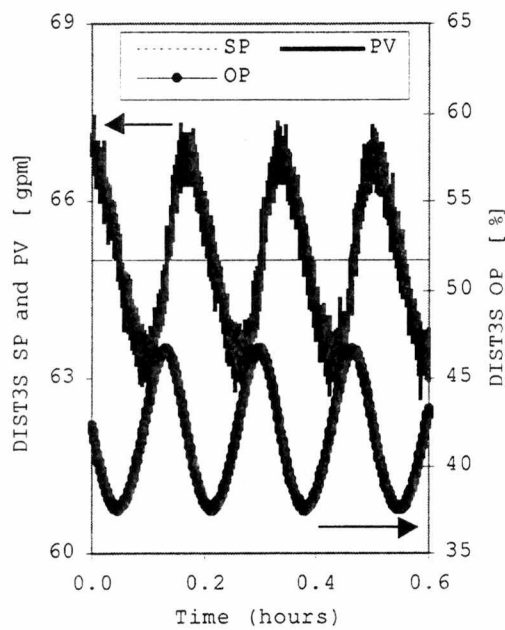


(c)

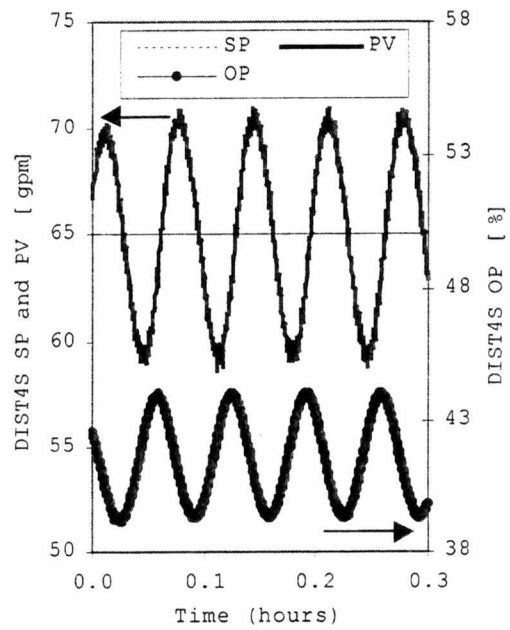


(d)

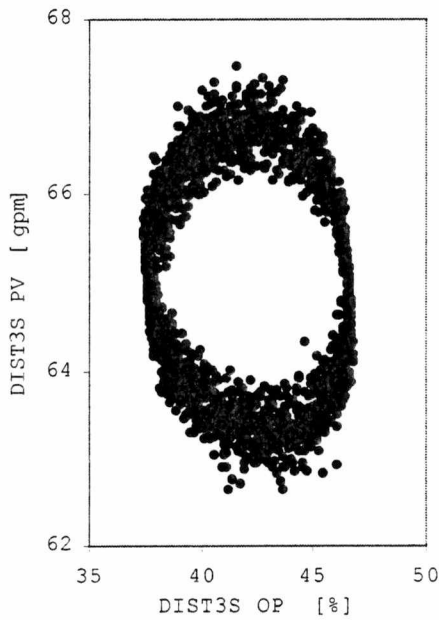
Figure A.10. Simulated data for disturbance problem (DIST1S and DIST2S): (a) SP/PV/OP data for DIST1S, (b) SP/PV/OP data for DIST2S, (c) PV/OP characteristic plot for DIST1S, (d) PV/OP characteristic plot for DIST2S.



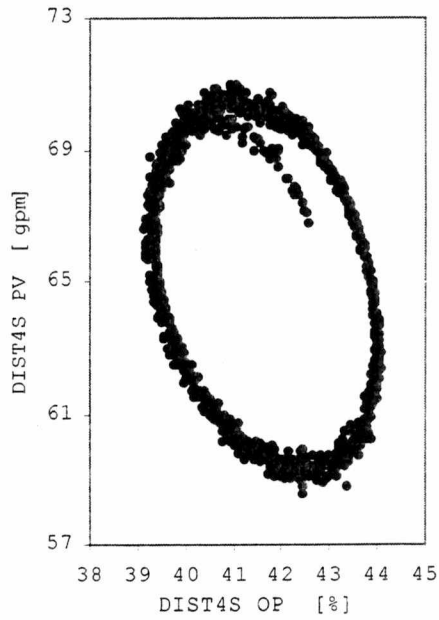
(a)



(b)

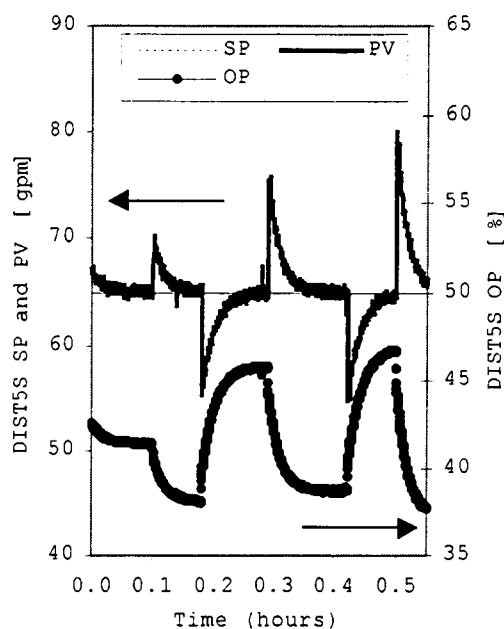


(c)

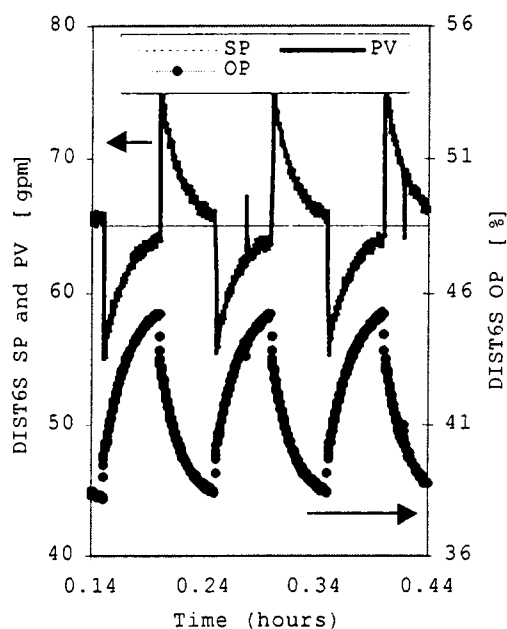


(d)

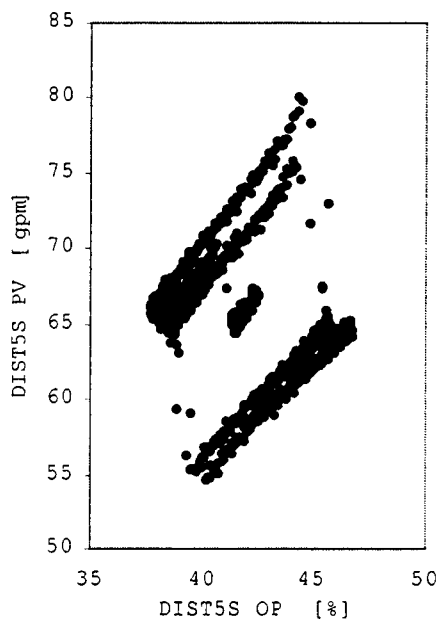
Figure A.11. Simulated data for disturbance problem (DIST3S and DIST4S): (a) SP/PV/OP data for DIST3S, (b) SP/PV/OP data for DIST4S, (c) PV/OP characteristic plot for DIST3S, (d) PV/OP characteristic plot for DIST4S.



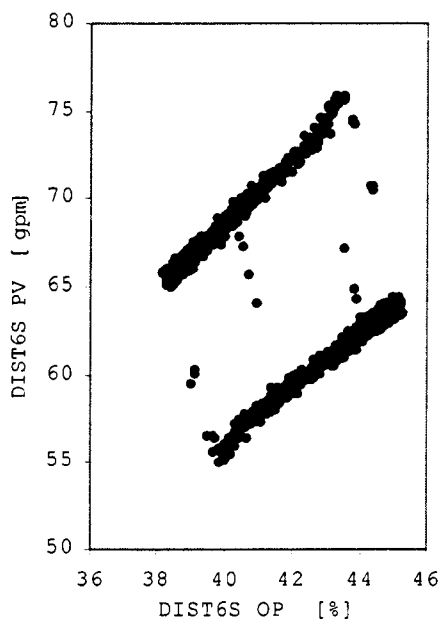
(a)



(b)

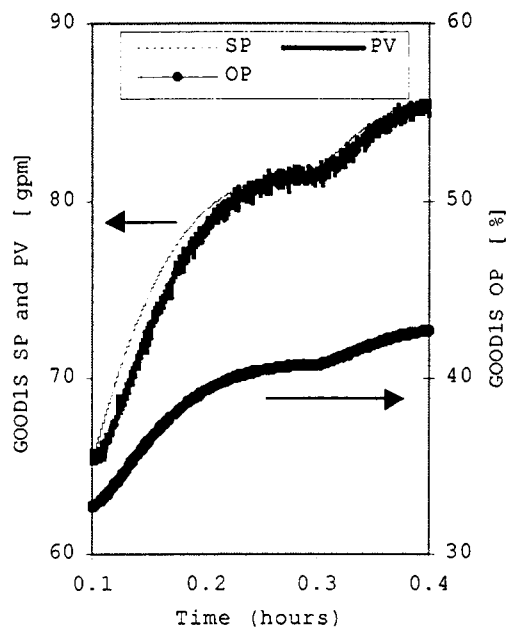


(c)

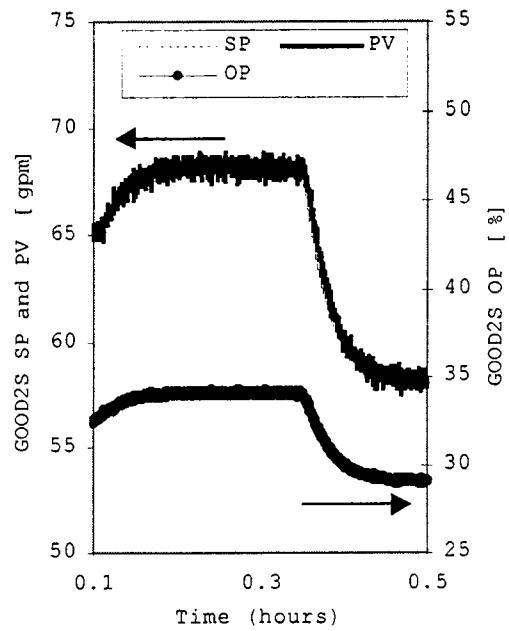


(d)

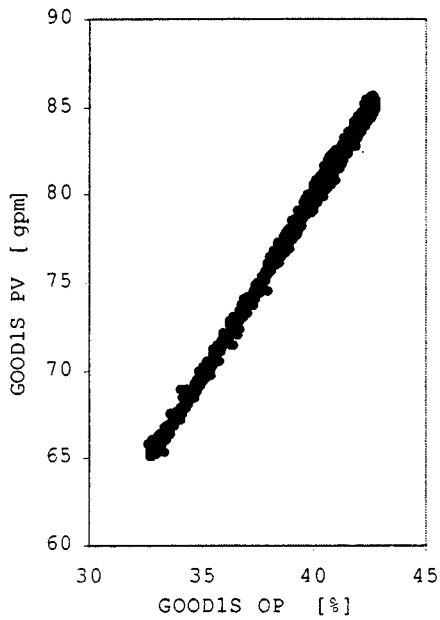
Figure A.12. Simulated data for disturbance problem (DIST5S and DIST6S): (a) SP/PV/OP data for DIST5S, (b) SP/PV/OP data for DIST6S, (c) PV/OP characteristic plot for DIST5S, (d) PV/OP characteristic plot for DIST6S.



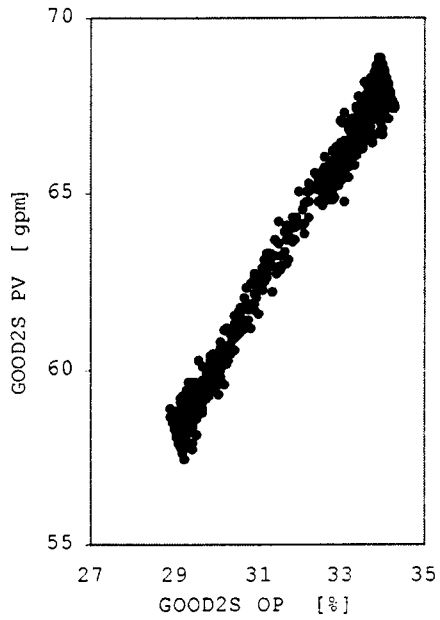
(a)



(b)

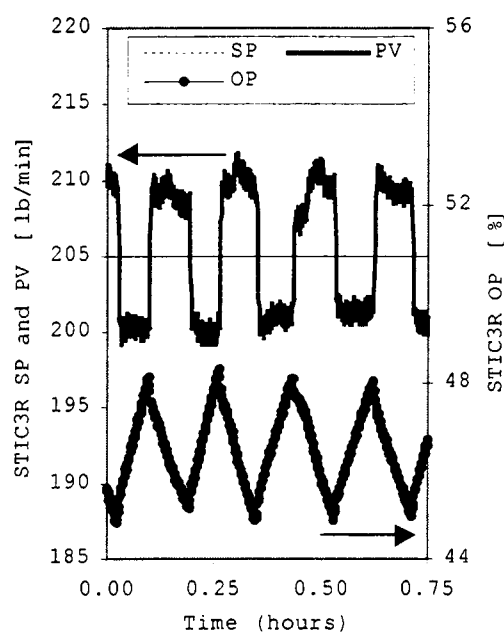


(c)

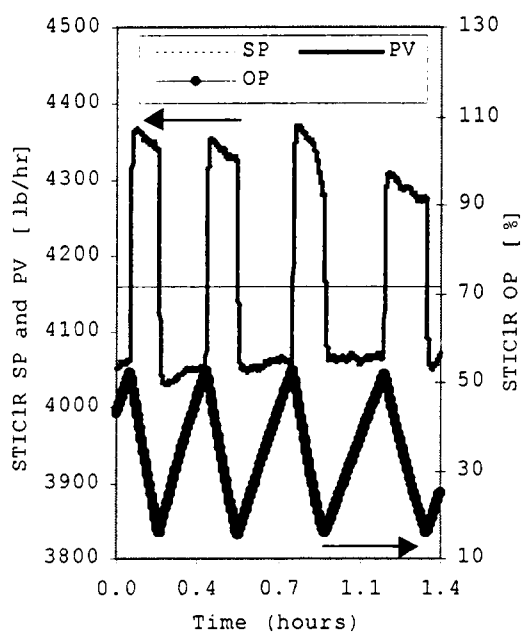


(d)

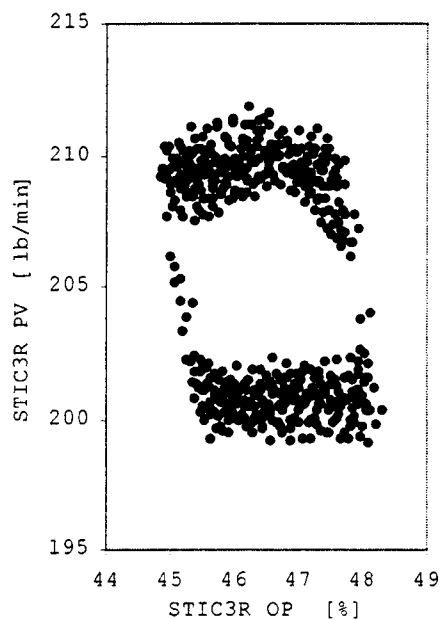
Figure A.13. Simulated data for good performance (GOOD1S and GOOD2S): (a) SP/PV/OP data for GOOD1S, (b) SP/PV/OP data for GOOD2S, (c) PV/OP characteristic plot for GOOD1S, (d) PV/OP characteristic plot for GOOD2S.



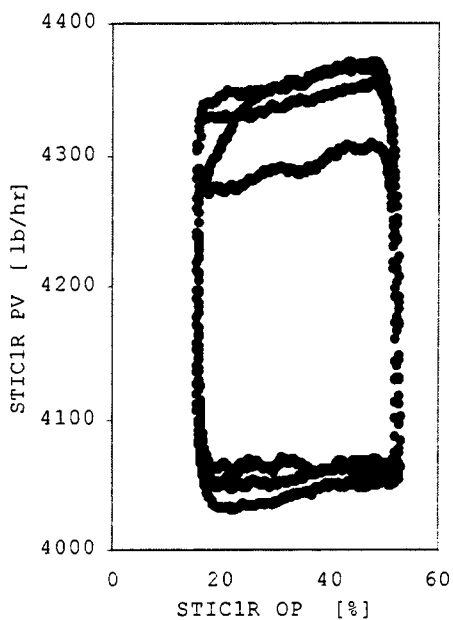
(a)



(b)

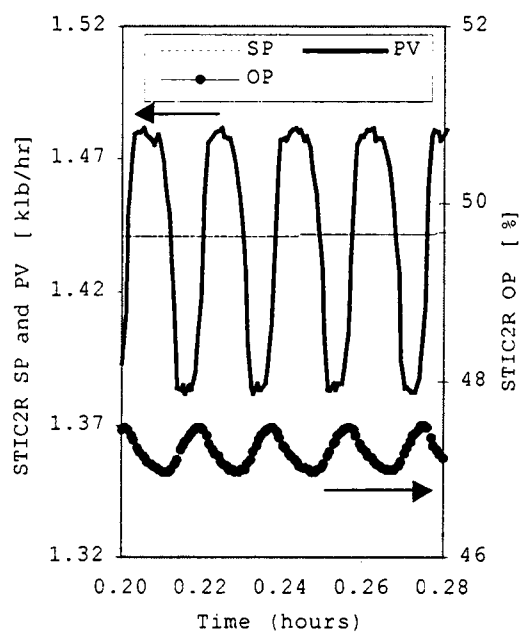


(c)

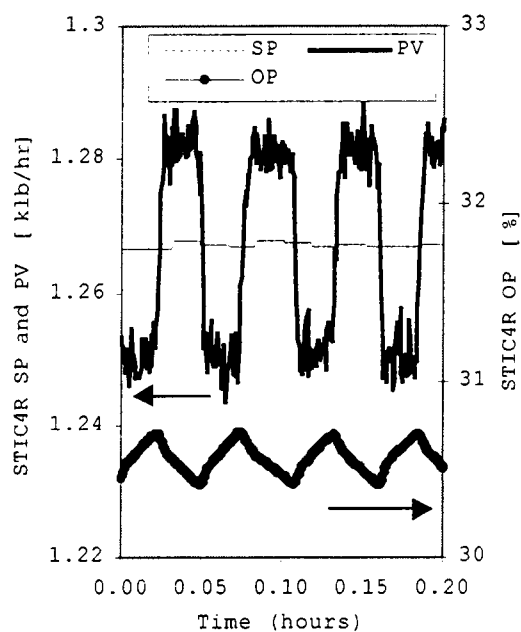


(d)

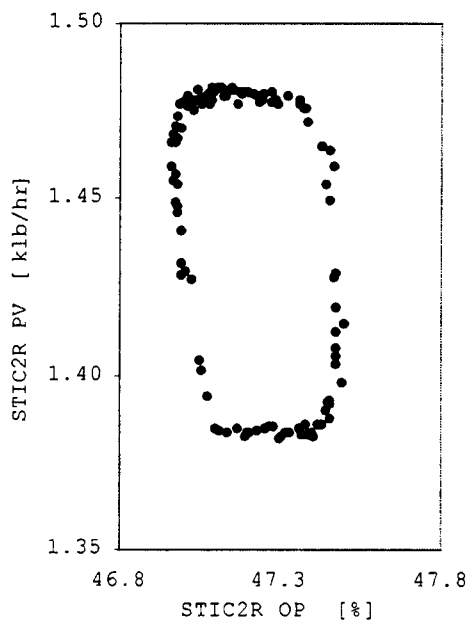
Figure A.14. Real data for valve stiction problem (STIC3R and STIC1R): (a) SP/PV/OP data for STIC3R, (b) SP/PV/OP data for STIC1R, (c) PV/OP characteristic plot for STIC3R, (d) PV/OP characteristic plot for STIC1R.



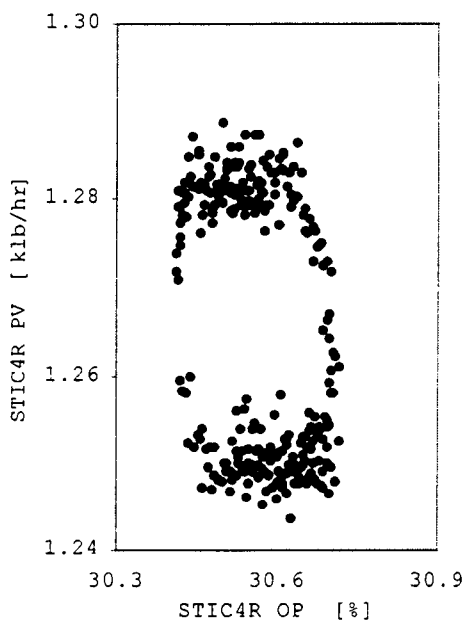
(a)



(b)



(c)



(d)

Figure A.15. Real data for valve stiction problem (STIC2R and STIC4R): (a) SP/PV/OP data for STIC2R, (b) SP/PV/OP data for STIC4R, (c) PV/OP characteristic plot for STIC2R, (d) PV/OP characteristic plot for STIC4R.

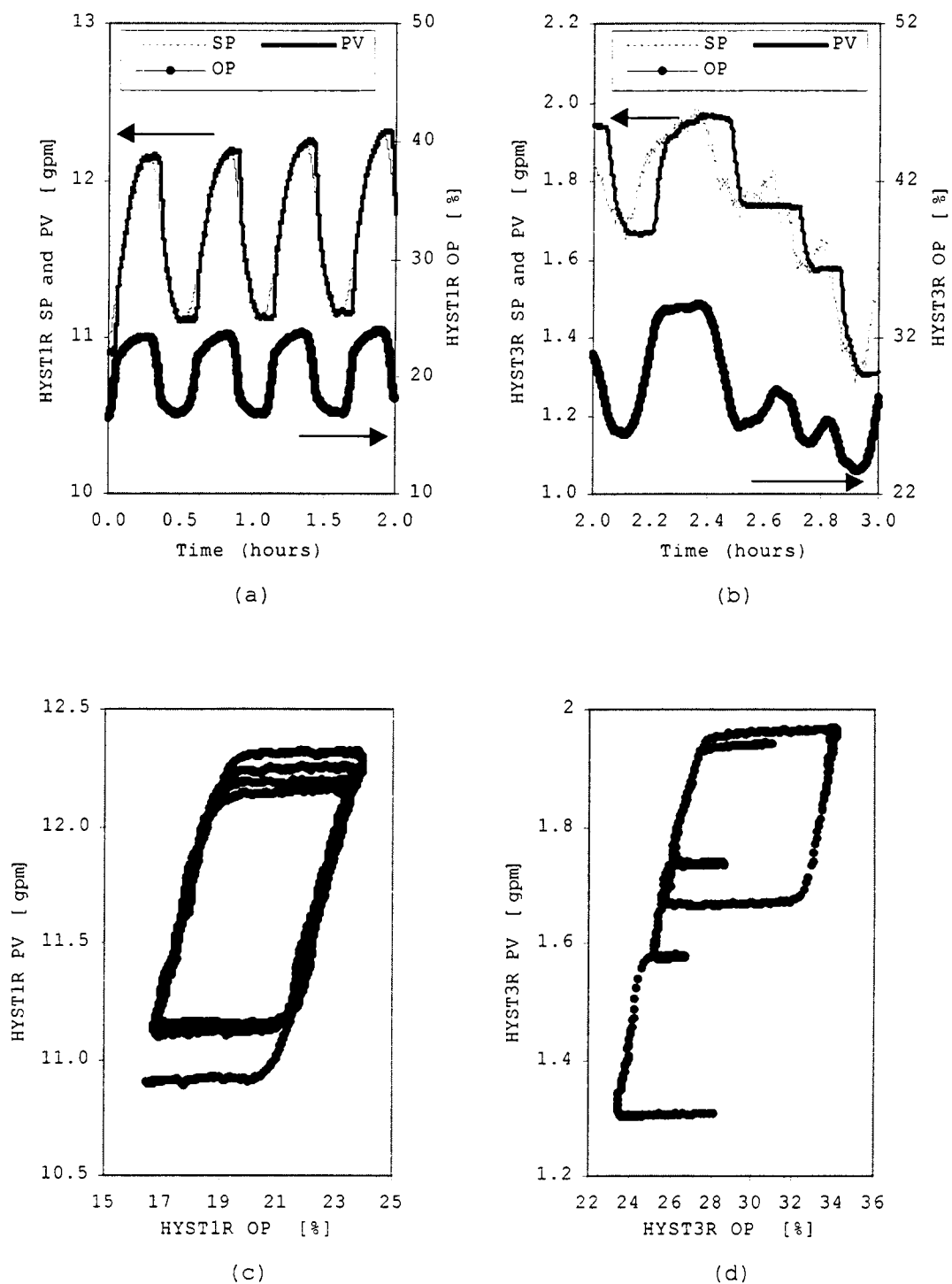
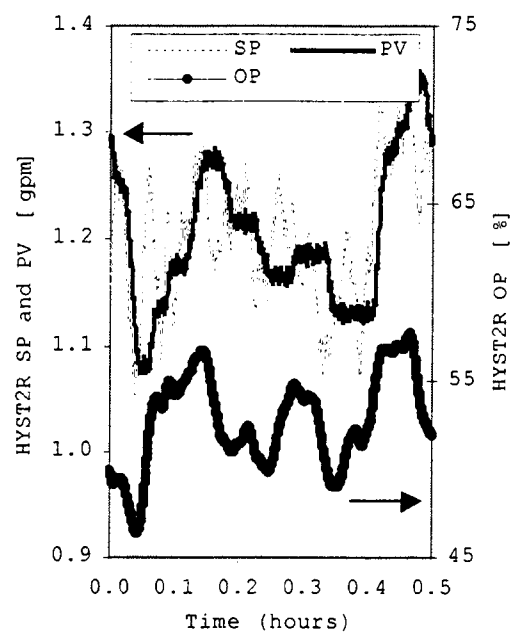
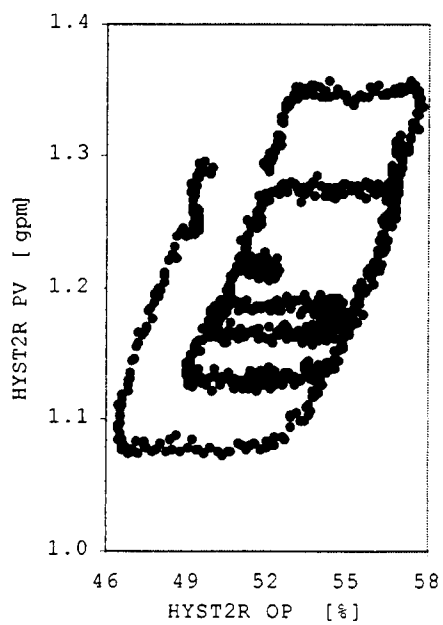


Figure A.16. Real data for valve hysteresis problem (HYST1R and HYST3R): (a) SP/PV/OP data for HYST1R, (b) SP/PV/OP data for HYST3R, (c) PV/OP characteristic plot for HYST1R, (d) PV/OP characteristic plot for HYST3R.

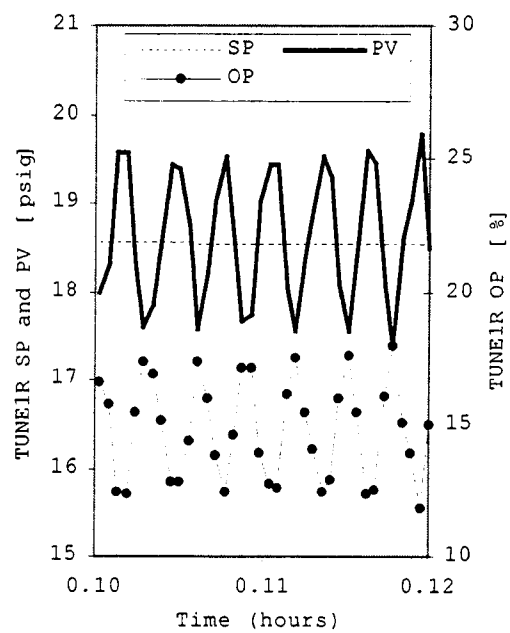


(a)

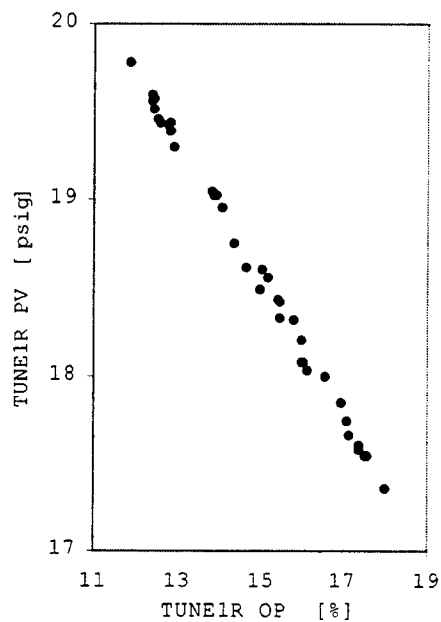


(b)

Figure A.17. Real data for valve hysteresis problem (HYST2R): (a) SP/PV/OP data for HYST2R, (b) PV/OP characteristic plot for HYST2R.

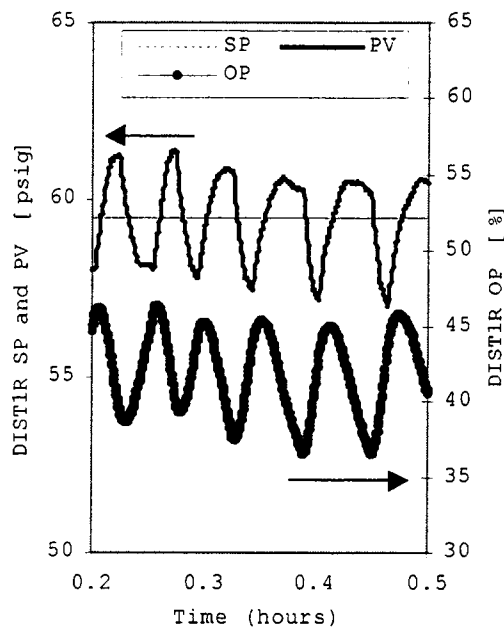


(a)

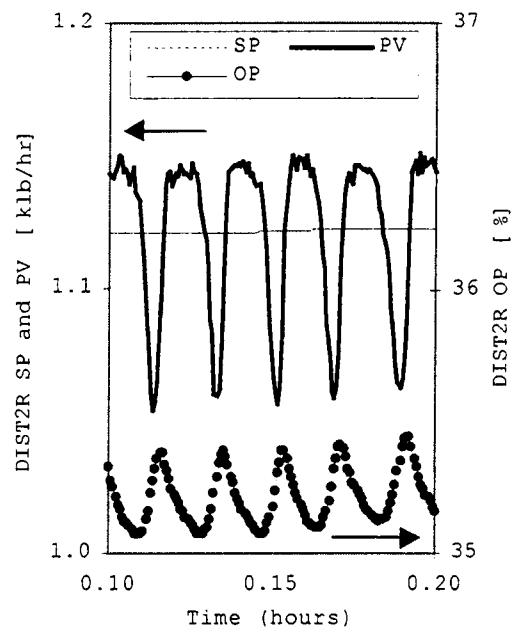


(b)

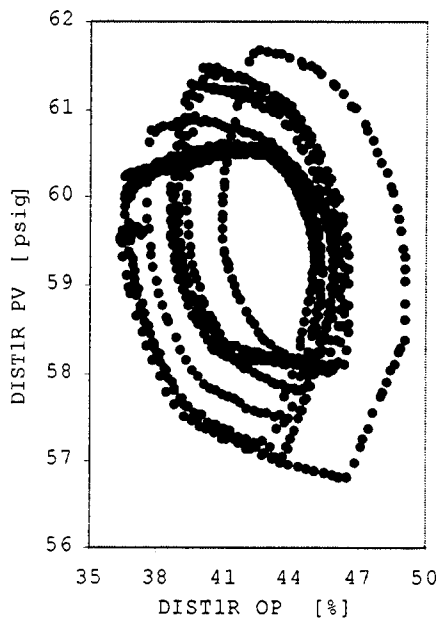
Figure A.18. Real data for tuning problem (TUNE1R): (a) SP/PV/OP data for TUNE1R, (b) PV/OP characteristic plot for TUNE1R.



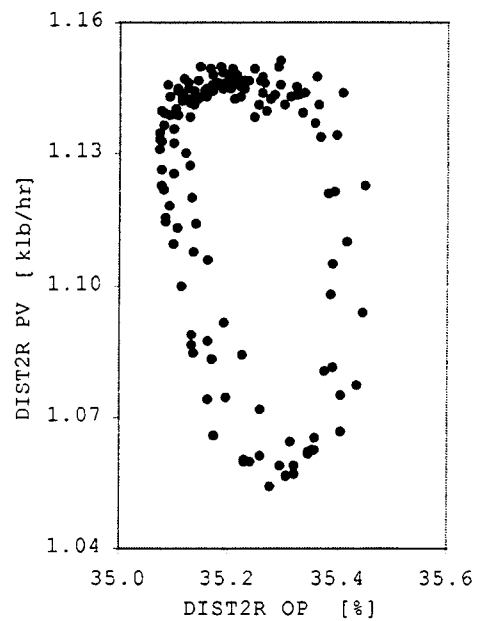
(a)



(b)

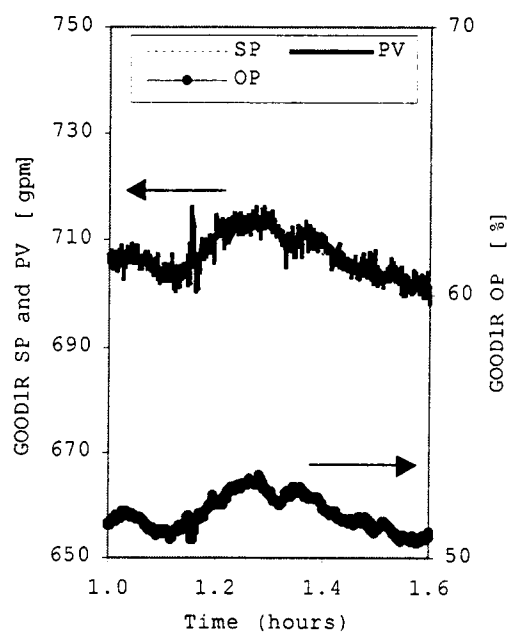


(c)

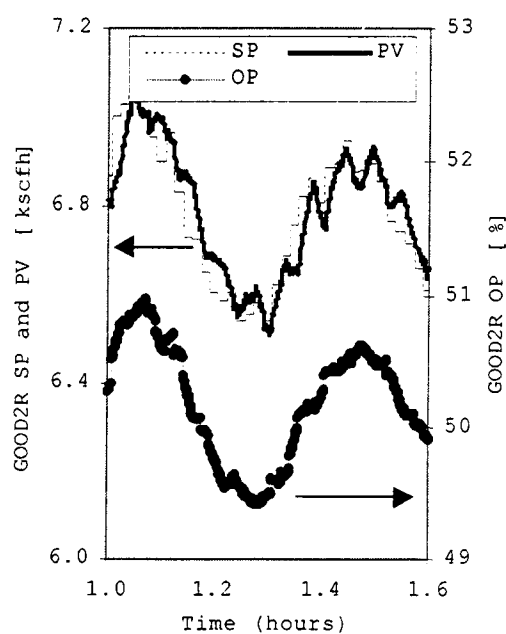


(d)

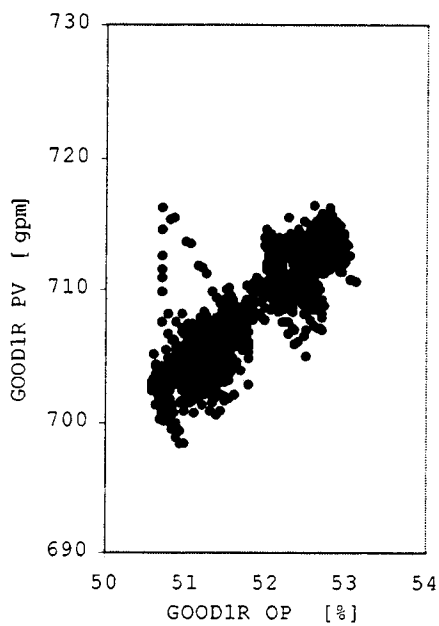
Figure A.19. Real data for disturbance problem (DIST1R and DIST2R): (a) SP/PV/OP data for DIST1R, (b) SP/PV/OP data for DIST2R, (c) PV/OP characteristic plot for DIST1R, (d) PV/OP characteristic plot for DIST2R.



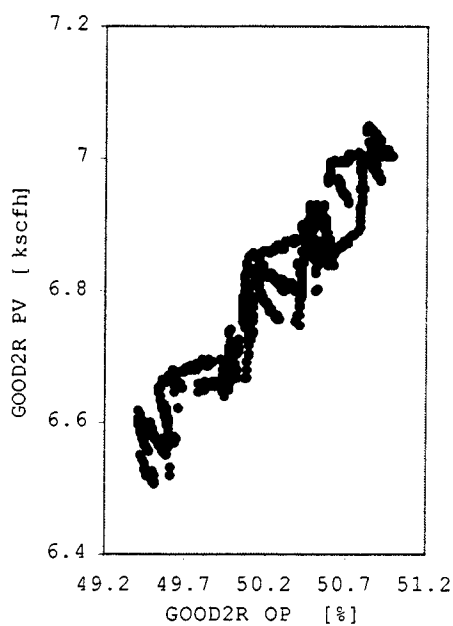
(a)



(b)



(c)



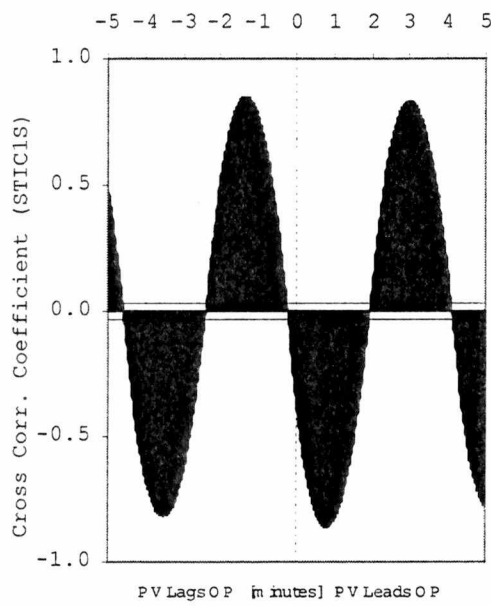
(d)

Figure A.20. Real data for good performance (GOOD1R and GOOD2R):
 (a) SP/PV/OP data for GOOD1R, (b) SP/PV/OP data for GOOD2R, (c) PV/OP
 characteristic plot for GOOD1R, (d) PV/OP characteristic plot for
 GOOD2R.

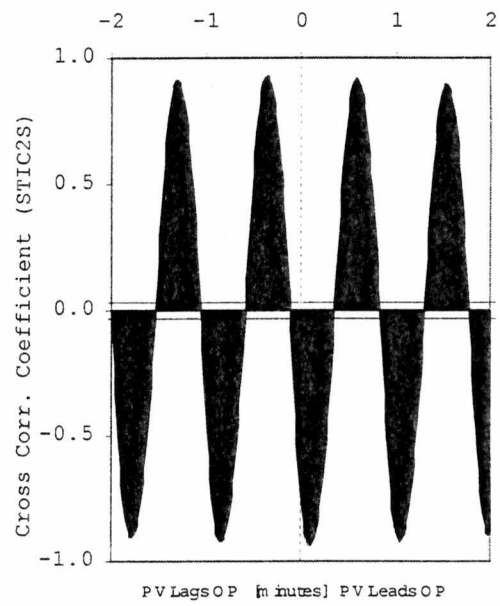
APPENDIX B

PV/OP CROSS CORRELATIONS FOR ALL SIMULATED AND REAL
PROCESS DATA SETS

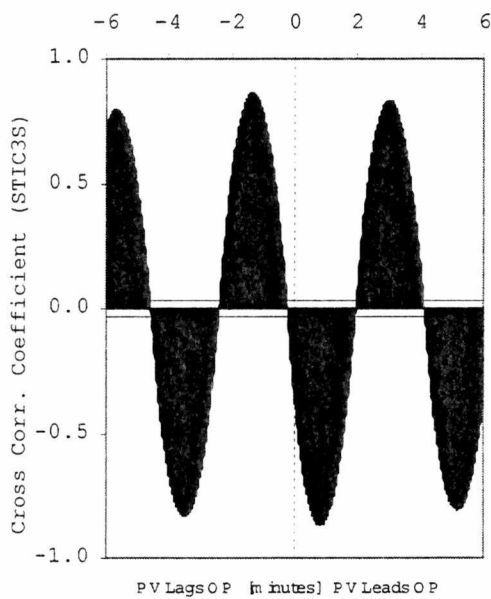
Appendix B contains PV/OP cross correlation plots for all of the simulated and real process data sets. The time periods for the plots are set to show several positive and negative correlation peaks for both lag and lead times.



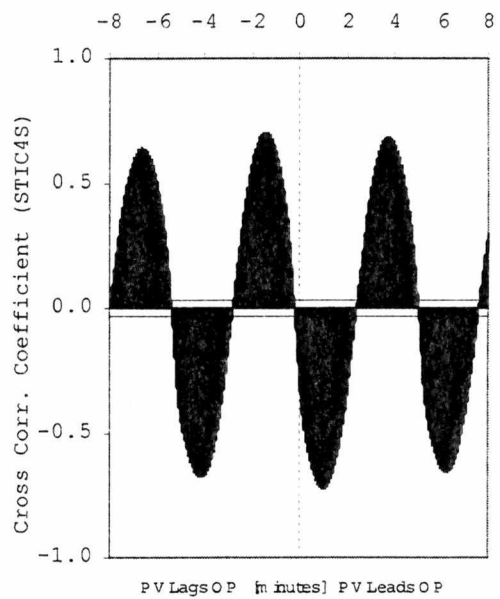
(a)



(b)

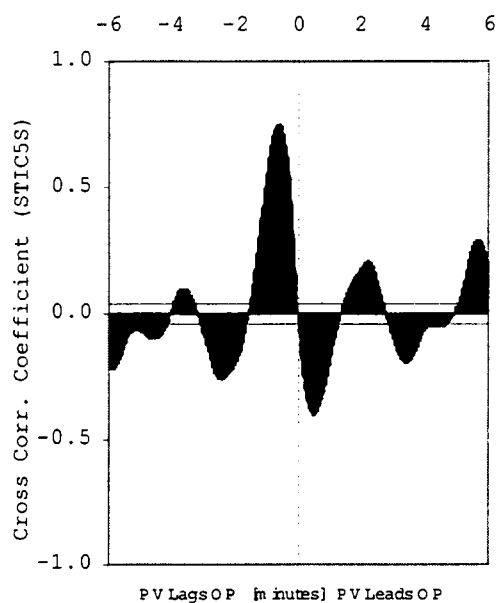


(c)

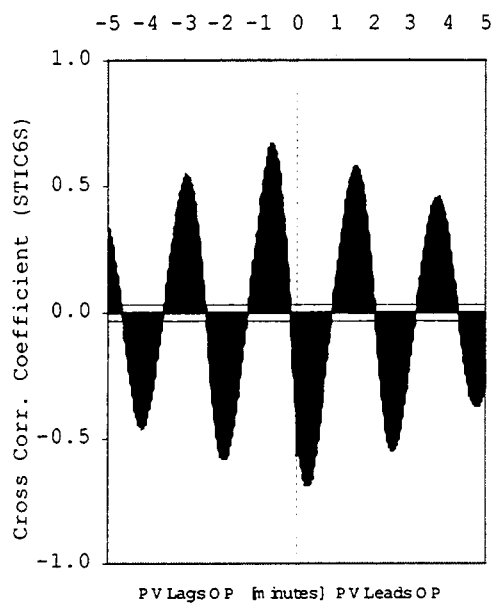


(d)

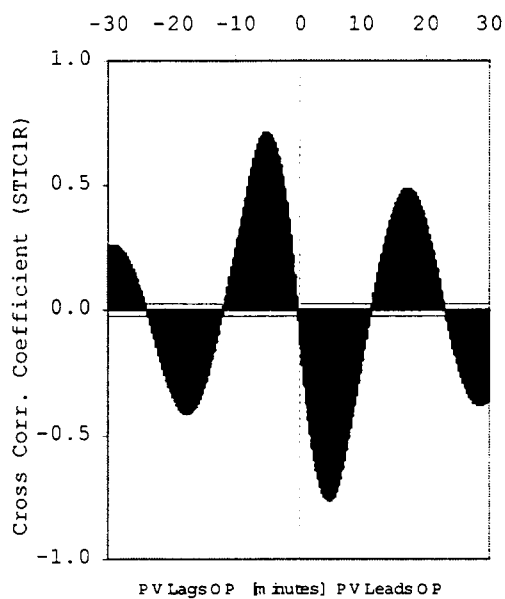
Figure B.1. PV/OP cross correlations for valve stiction problem:
 (a) PV/OP correlation for STIC1S, (b) PV/OP correlation for STIC2S,
 (c) PV/OP correlation for STIC3S, (d) PV/OP correlation for STIC4S.



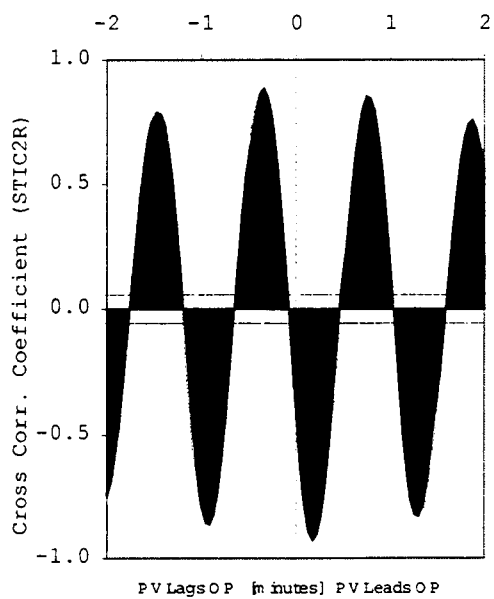
(a)



(b)

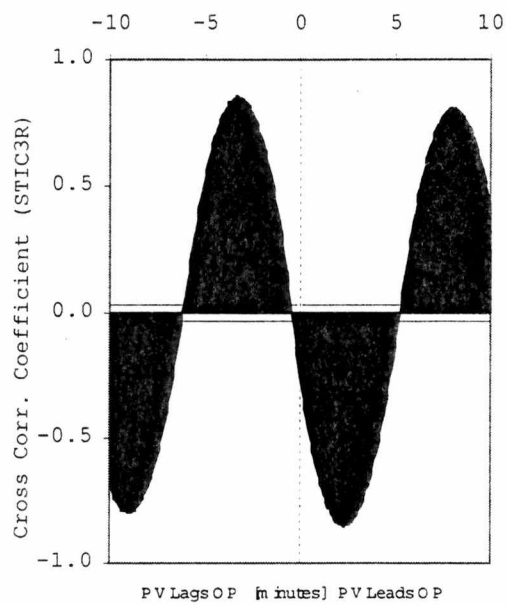


(c)

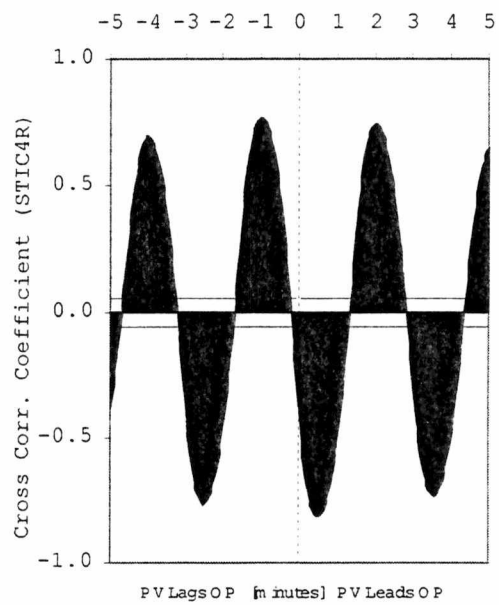


(d)

Figure B.2. PV/OP cross correlations for valve stiction problem:
 (a) PV/OP correlation for STIC5S, (b) PV/OP correlation for STIC6S,
 (c) PV/OP correlation for STIC1R, (d) PV/OP correlation for STIC2R.

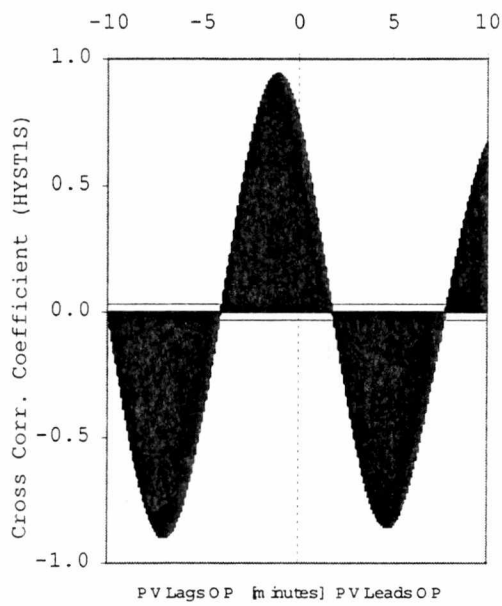


(a)

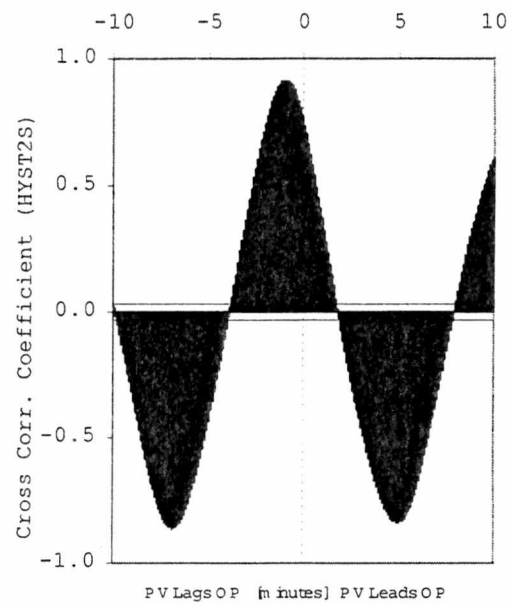


(b)

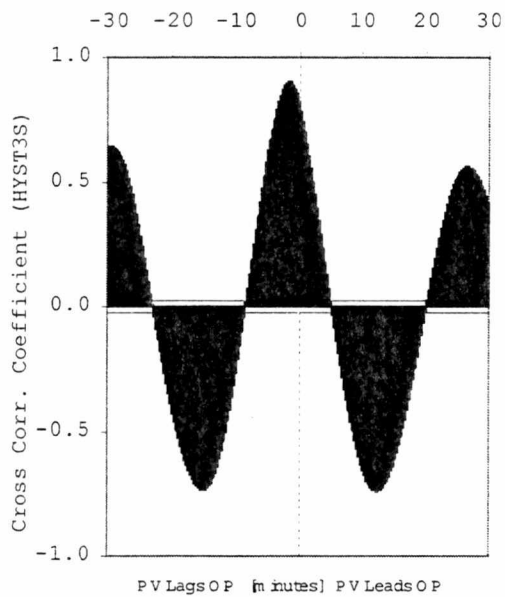
Figure B.3. PV/OP cross correlations for valve stiction problem:
 (a) PV/OP correlation for STIC3R, (b) PV/OP correlation for STIC4R.



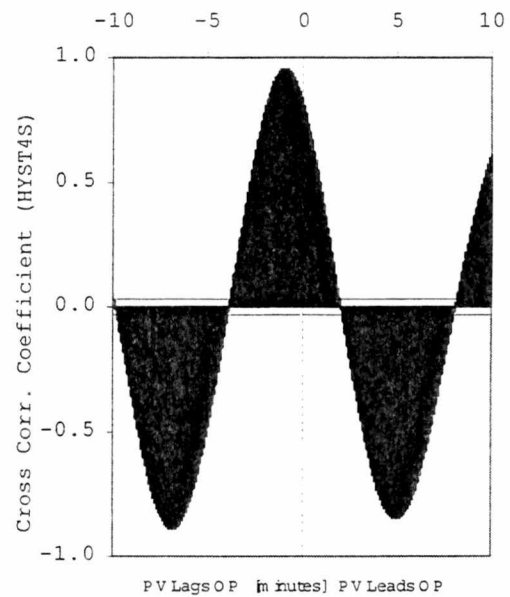
(a)



(b)

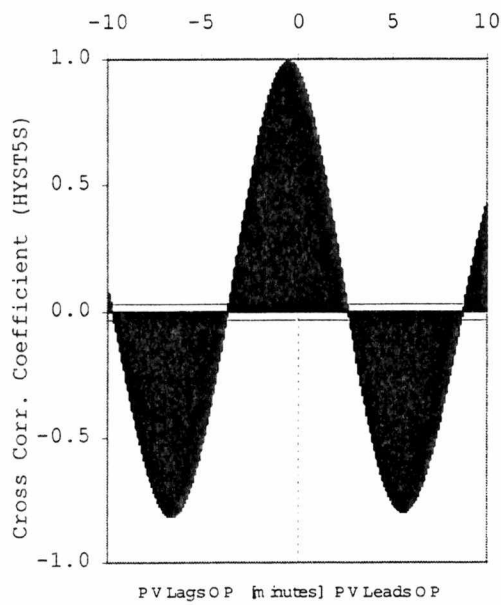


(c)

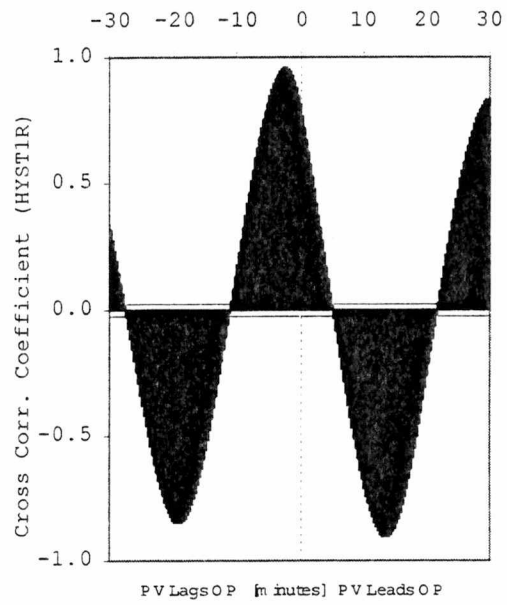


(d)

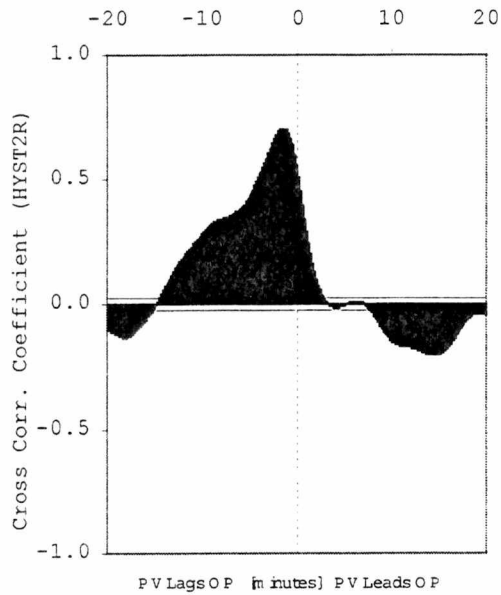
Figure B.4. PV/OP cross correlations for valve hysteresis problem:
 (a) PV/OP correlation for HYST1S, (b) PV/OP correlation for HYST2S,
 (c) PV/OP correlation for HYST3S, (d) PV/OP correlation for HYST4S.



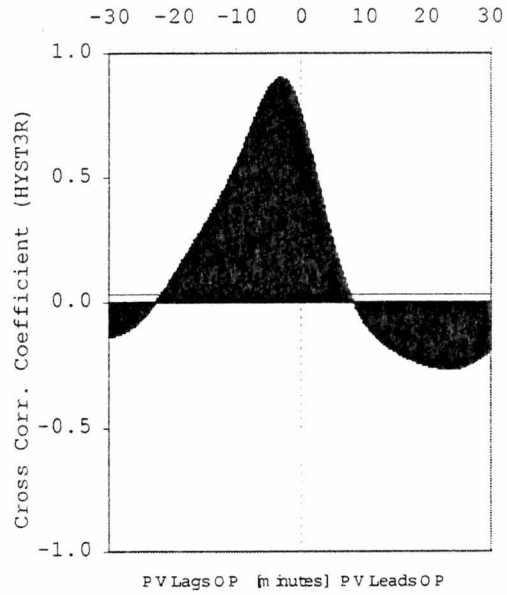
(a)



(b)

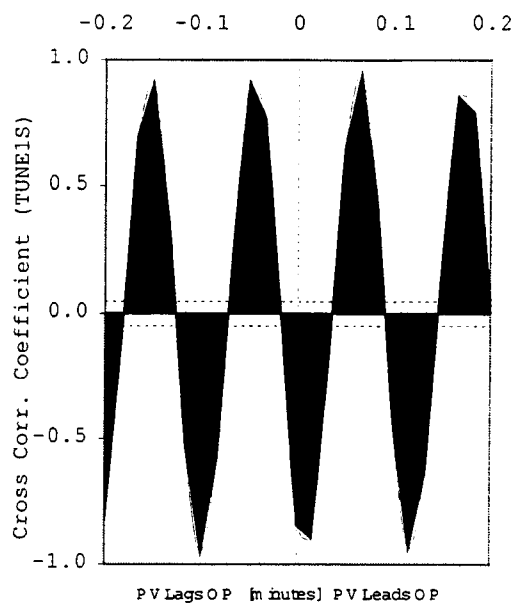


(c)

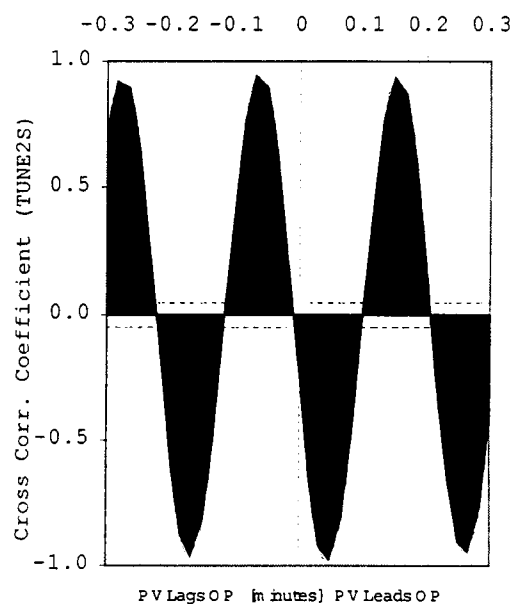


(d)

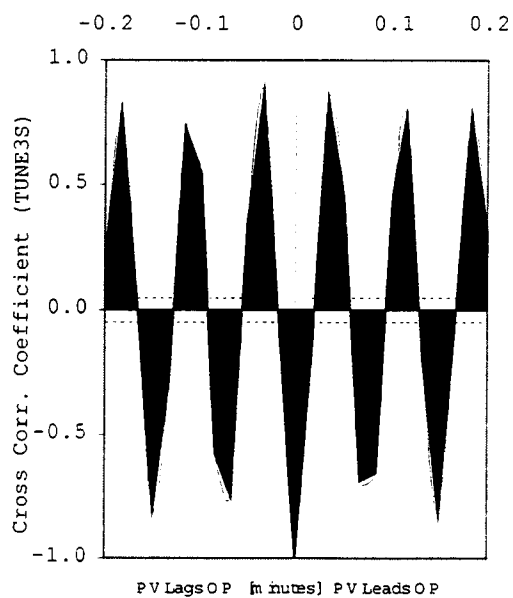
Figure B.5. PV/OP cross correlations for valve hysteresis problem:
 (a) PV/OP correlation for HYST5S, (b) PV/OP correlation for HYST1R,
 (c) PV/OP correlation for HYST2R, (d) PV/OP correlation for HYST3R.



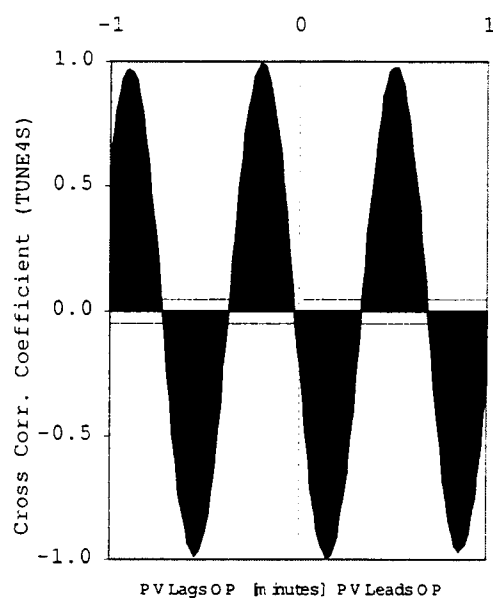
(a)



(b)

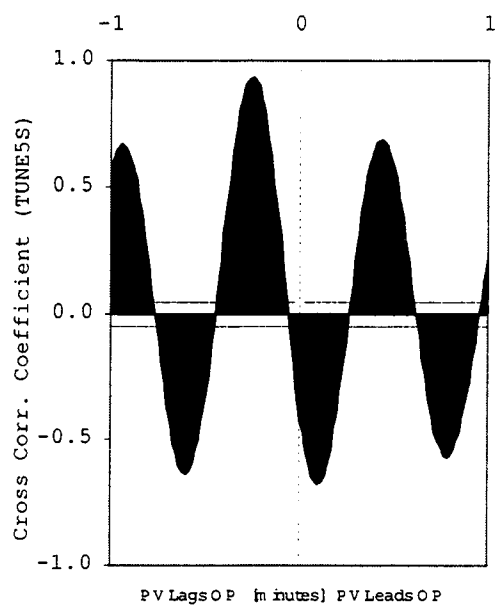


(c)

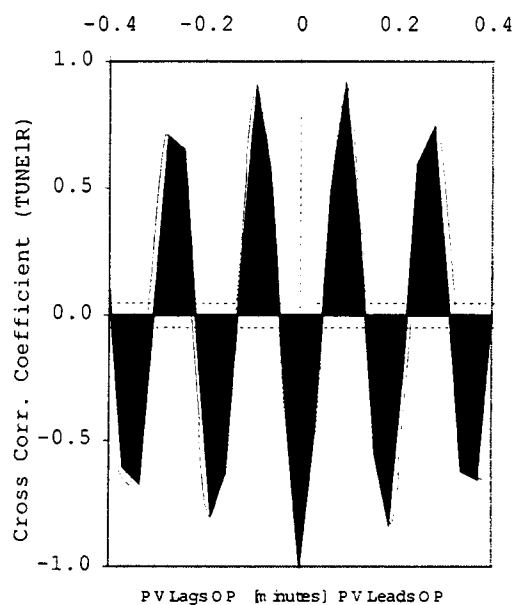


(d)

Figure B.6. PV/OP cross correlations for tuning problem: (a) PV/OP correlation for TUNE1S, (b) PV/OP correlation for TUNE2S, (c) PV/OP correlation for TUNE3S, (d) PV/OP correlation for TUNE4S.

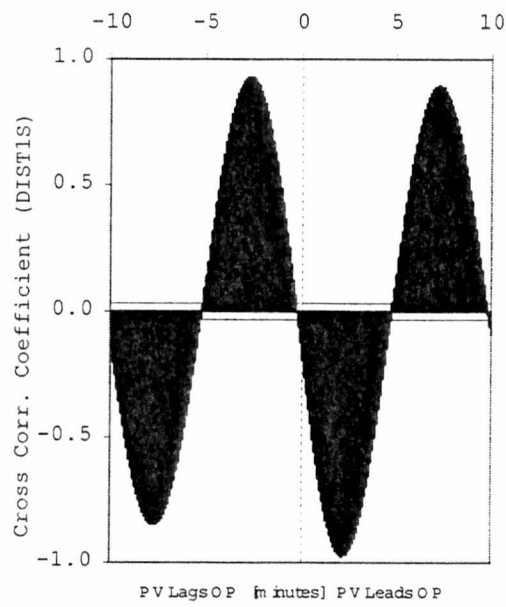


(a)

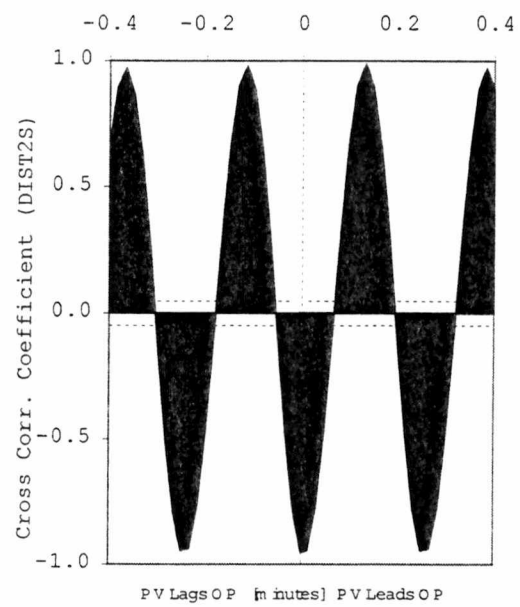


(b)

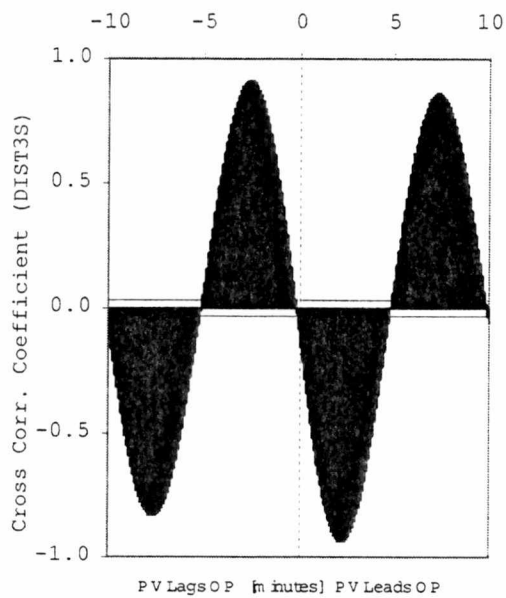
Figure B.7. PV/OP cross correlations for tuning problem: (a) PV/OP correlation for TUNE5S, (b) PV/OP correlation for TUNE1R.



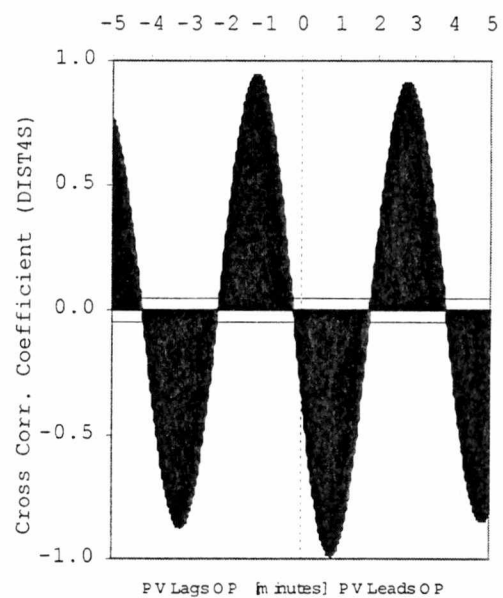
(a)



(b)

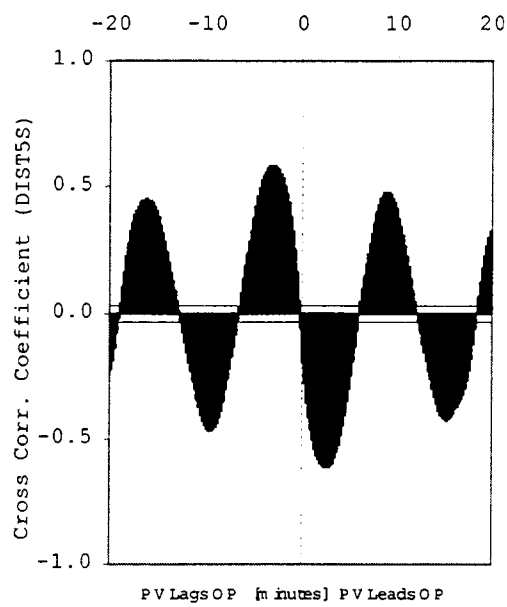


(c)

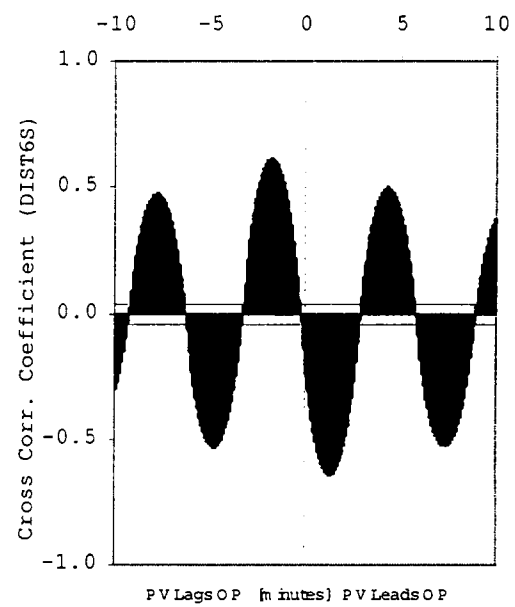


(d)

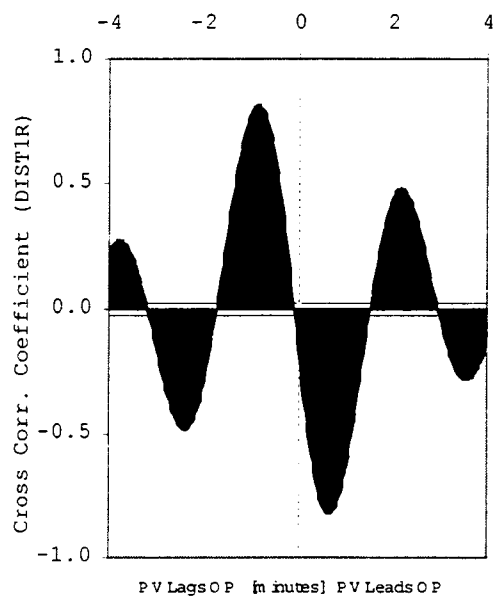
Figure B.8. PV/OP cross correlations for disturbance problem: (a) PV/OP correlation for DIST1S, (b) PV/OP correlation for DIST2S, (c) PV/OP correlation for DIST3S, (d) PV/OP correlation for DIST4S.



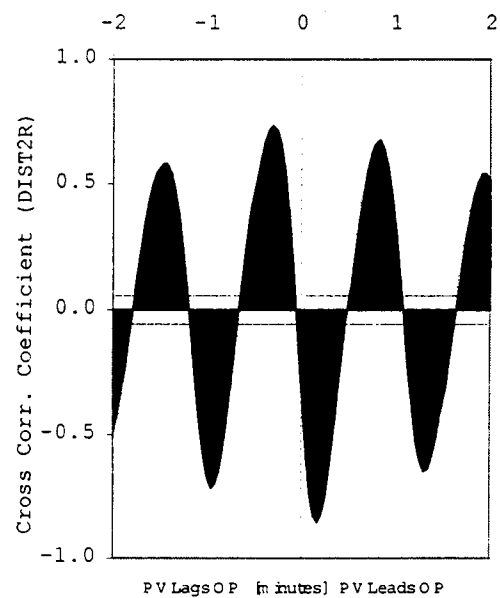
(a)



(b)

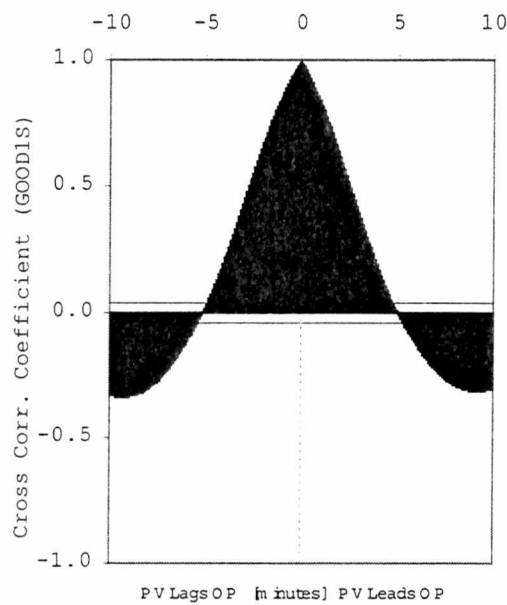


(c)

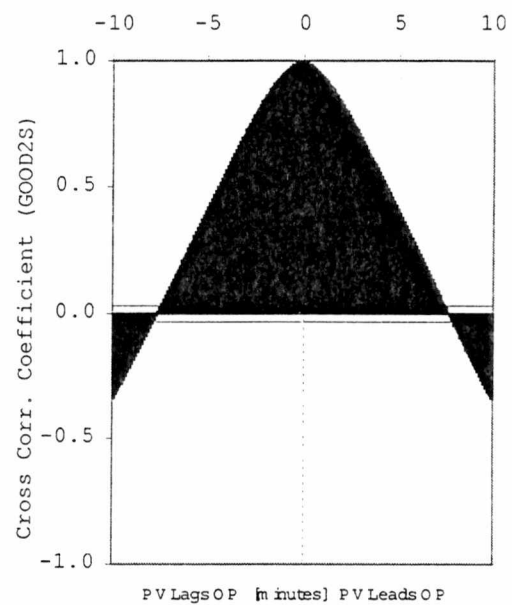


(d)

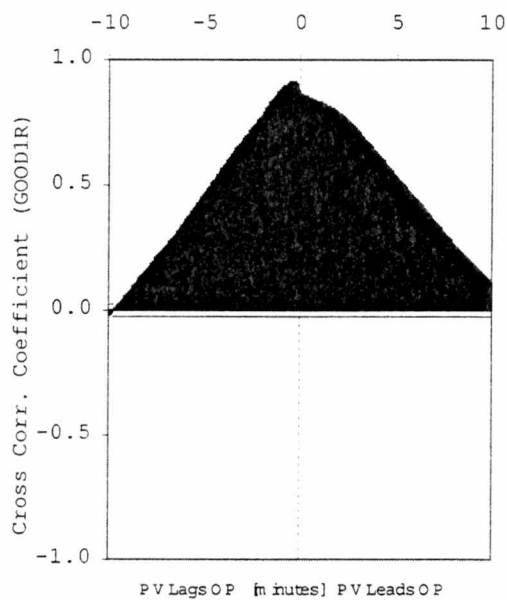
Figure B.9. PV/OP cross correlations for disturbance problem: (a) PV/OP correlation for DIST5S, (b) PV/OP correlation for DIST6S, (c) PV/OP correlation for DIST1R, (d) PV/OP correlation for DIST2R.



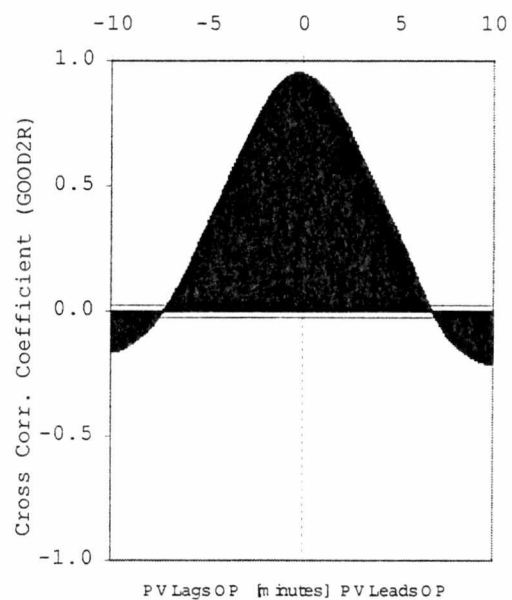
(a)



(b)



(c)



(d)

Figure B.10. PV/OP cross correlations for good performance: (a) PV/OP correlation for GOOD1S, (b) PV/OP correlation for GOOD2S, (c) PV/OP correlation for GOOD1R, (d) PV/OP correlation for GOOD2R.

APPENDIX C

**MATLAB M-FILES USED FOR MONITORING ALGORITHM APPLICATION
AND DATA ANALYSIS APPLICATION**

CLPM_HAGG.M

```
%
% CLPM_HAGG.M
% Program to perform the Hagglund Control Loop Performance
% Monitor (CLPM) algorithm.
% Program produces 3 plots (SP/PV/OP for entire data
% range, SP/PV/OP for small data range, and CLPM plot
% showing the CLPM rate of oscillation relative to the
% alarm limit).
%

figure(1);
clf;
figure(1);
subplot(3,1,1);

clear x;
clear y1;
clear y2;
clear z;

y1column=input('Column # for y1 data - PV: '); % User prompts
y2column=input('Column # for y2 data - OP: ');
zcolumn=input('Column # for z data - SP: ');
start_t=input('Enter start time: ');
t_length=input('Length of time: ');

b=eval(fname);
y1 = b( round((start_t*3600)/samp_time)+1: ...
        round((start_t*3600)/samp_time) + 1 + ...
        round((t_length*3600)/samp_time), ...
        y1column);
y2 = b( round((start_t*3600)/samp_time)+1: ...
        round((start_t*3600)/samp_time) + 1 + ...
        round((t_length*3600)/samp_time), ...
        y2column);
z = b( round((start_t*3600)/samp_time)+1: ...
        round((start_t*3600)/samp_time) + 1 + ...
        round((t_length*3600)/samp_time), ...
        zcolumn);

[m,n]=size(y1);
x(1,1)=start_t; % Calculate time variable
for i = 2:m;
    x(i,1) = x(i-1,1) + samp_time/3600;
end;

[ax,h1,h2]=plotyy(x,y1,x,y2); % Plot PV and OP
hold on;

plot(x,z,'r--'); % Plot SP on same plot

set(get(ax(1),'xlabel'),'string','Time (hours)'); % Plot labels

set(get(ax(1),'ylabel'),'string','SP (red) and PV (blue)');
set(get(ax(2),'ylabel'),'string','OP (green)');
set(h1,'linestyle','-','color','b');
set(h2,'linestyle',':','color','g');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot SP, PV, and OP data over short time range to allow visual
% inspection.
%
subplot(3,1,2);

clear xzoom;
clear ylzoom;
```

• CLPM_HAGG.M (continued)

```

clear y2zoom;
clear zzoom;

st_time=input('Enter start time for data zoom (hours): ');
dur_time=input('Enter total plot time for data zoom (hours): ');

xzzoom=x((st_time*3600)/samp_time:(st_time*3600)/samp_time + ...
          (dur_time*3600/samp_time));
ylzoom=y1((st_time*3600)/samp_time:(st_time*3600)/samp_time + ...
          (dur_time*3600/samp_time));
y2zoom=y2((st_time*3600)/samp_time:(st_time*3600)/samp_time + ...
          (dur_time*3600/samp_time));
zzoom=z((st_time*3600)/samp_time:(st_time*3600)/samp_time + ...
          (dur_time*3600/samp_time));

[ax,h1,h2]=plotyy(xzzoom,ylzoom,xzzoom,y2zoom); % Plot PV and OP
hold on;

plot(xzzoom,zzoom,'r--'); % Plot SP on same plot

set(get(ax(1),'xlabel'),'string','Time (hours)'); % Plot labels
set(get(ax(1),'ylabel'),'string','SP (red) and PV (blue)');
set(get(ax(2),'ylabel'),'string','OP (green)');
set(h1,'linestyle','-','color','b');
set(h2,'linestyle','-','color','g');
title('SP/PV/OP Over Short Time Period');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Perform CLPM algorithm
% User sets oscillation limit and estimate of oscillation frequency
%
subplot(3,1,3);
clear rosc;

% User prompts

pvlo=input('PVLO: '); % PV low
pvhi=input('PVHI: '); % PV high
aperc=input('Acceptable % Osc. Limit: '); % Error limit in % of PV range
tu=input('Estimate of osc. freq in seconds: '); % tu/2 is half period osc.
nlim=input('Limit of number of oscillations: '); % Alarm limit

% Initialize algorithm variables
loopiae=0.0; % Loop cumulative IAE during potential osc.
iaelim=(2 * aperc)/(2 * 3.1415 / tu); % Limit for counting as an oscillation
% Value set based on sine wave approx. with
% error limit and estimate of osc. period
tsup=5*nlim*tu; % Supervisory time used for filtering rate of oscillation
oload=0.0; % Set to 1 when countable oscillation occurs
oflag=0.0; % Flag for oscillation during a given deviation
olderr=0.0; % Initial value for controller error

[m,n]=size(y1);

rosc(1,1)=0.0; % Initialize the rate of oscillation variable array

for i=2:m; % Loop through set of data sample by sample to perform
% algorithm calculations

looperr=((z(i,1) - y1(i,1))/(pvhi - pvlo)) * 100.0; % Error in % of PV range

if looperr * olderr >= 0.0; % No controller error sign change
loopiae = loopiae + abs(looperr) * samp_time; % Increment loop IAE
if loopiae > iaelim; % Is limit for counting as an oscillation exceeded?

```

CLPM_HAGG.M (continued)

```

if oflag < 0.5;
    oload = 1.0;    % Load variable for adding "1" to rate of oscillation
    oflag = 1.0;    % Flag so that only 1 oscillation can be counted
end;              % for a given deviation above/below setpoint
end;
else;
    % Controller error sign change
    loopiae = abs(looperr) * samp_time; % Reset loop IAE
    oflag = 0.0;
end;

rosc(i,1)=(1-(samp_time / tsup)) * rosc(i-1,1) + oload; % rosc set each sample time
olderr = looperr; % and is filtered to effect
oload = 0.0;      % a moving time window
%
end;
%
%
plot(x,rosc,'g-');

v=axis;           % Find nice axis/plot limits
if v(1,4) > nlim;
    ym=max(rosc)*1.5;
else;
    ym=nlim*2;
end;

hold on;

set(gca,'XLim',[min(x) max(x)]); % Plot labels
set(gca,'YLim',[0,ym]);
plot(x,rosc,'g-'); % Plot rate of oscillation as function of time
hold on;
plot([v(1,1) v(1,2)],[nlim nlim],'m--'); % Plot alarm limit line

ylabel('PV Osc. Rate (green)'); % More plot labels
xlabel('Time ( hours)');
title('Hagglund CLPM');

```

PMA_ALG.M

```

%
% PMA_ALG.M
% Program to perform the Performance Monitoring Algorithm (PMA).
% This program calculates the percent time oscillating for a
% control loop based on user specified criteria.
% Program produces 4 plots (SP/PV/OP for entire time range,
% SP/PV/OP for short time period, percent time oscillating
% (with avg. peak deviation), and rate of oscillation
% (with avg. frequency).
%
figure(1);
clf;
figure(1);
subplot(2,2,1);

clear x;
clear y1;
clear y2;
clear z;

% User prompts
ylcolumn=input('Column # for y1 data - PV: '); % PV data
y2column=input('Column # for y2 data - OP: '); % OP data
zcolumn=input('Column # for z data - SP: '); % SP data
start_t=input('Enter start time: ');
t_length=input('Length of time: ');

b=eval(fname);
y1 = b( round((start_t*3600)/samp_time)+1: ...
        round((start_t*3600)/samp_time) + 1 + ...
        round((t_length*3600)/samp_time), ...
        ylcolumn);
y2 = b( round((start_t*3600)/samp_time)+1: ...
        round((start_t*3600)/samp_time) + 1 + ...
        round((t_length*3600)/samp_time), ...
        y2column);
z = b( round((start_t*3600)/samp_time)+1: ...
        round((start_t*3600)/samp_time) + 1 + ...
        round((t_length*3600)/samp_time), ...
        zcolumn);

[m,n]=size(y1);
x(1,1)=start_t; % Calculate time variable
for i = 2:m;
    x(i,1) = x(i-1,1) + samp_time/3600;
end;

[ax,h1,h2]=plotyy(x,y1,x,y2); % Plot PV and OP
hold on;

plot(x,z,'r--'); % Plot SP on same plot

set(get(ax(1),'xlabel'),'string','Time (hours)'); % Plot labels
set(get(ax(1),'ylabel'),'string','SP (red) and PV (blue)');
set(get(ax(2),'ylabel'),'string','OP (green)');
set(h1,'linestyle','-','color','b');
set(h2,'linestyle',':','color','g');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot data for short time period
subplot(2,2,2);

clear xzoom;
clear y1zoom;
clear y2zoom;
clear zzoom;

```

PMA_ALG.M (continued)

```
% User prompts

st_time=input('Enter start time for data zoom (hours): ');
dur_time=input('Enter total plot time for data zoom (hours): ');
xzoom=x((st_time*3600)/samp_time:(st_time*3600)/samp_time + ...
        (dur_time*3600)/samp_time));
ylzoom=y1((st_time*3600)/samp_time:(st_time*3600)/samp_time + ...
        (dur_time*3600)/samp_time));
y2zoom=y2((st_time*3600)/samp_time:(st_time*3600)/samp_time + ...
        (dur_time*3600)/samp_time));
zzoom=z((st_time*3600)/samp_time:(st_time*3600)/samp_time + ...
        (dur_time*3600)/samp_time));

[ax,h1,h2]=plotyy(xzoom,ylzoom,xzoom,y2zoom); % Plot PV and OP
hold on;

plot(xzoom,zzoom,'r--'); % Plot SP on same plot

set(get(ax(1),'xlabel'),'string','Time (hours)'); % Plot labels
set(get(ax(1),'ylabel'),'string','SP (red) and PV (blue)');
set(get(ax(2),'ylabel'),'string','OP (green)');
set(h1,'linestyle','-','color','b');
set(h2,'linestyle',':','color','g');
title('SP/PV/OP Over Short Time Period');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Perform PMA calculations and plot the percent time oscillating
% (with avg. peak deviation).
%
subplot(2,2,3);

clear filtpv;
clear filtsp;
clear tm_osc;
clear num_osc;
clear pc_tm_osc;
clear sm_pk_dev;
clear av_pk_dev;
clear fl_av_pk_dev;
clear av_osc_freq;

% User prompts
pvlo=input('Enter low PV limit: '); % Low PV limit
pvhi=input('Enter high PV limit: '); % High PV limit

% Filter time to effect moving time window
tm_range=input('Enter time range in hours for osc. window: ');

% Limit for counting as an oscillation
osc_base_lim=input('Enter low limit (fraction, eg 0.01) for osc. counting: ');

pvevmfilt=input('Enter ewma filter value for PV: ');
spevmfilt=input('Enter ewma filter value for SP: ');

%Initialize algorithm variables
hlfper_osc=0.0;
tosc_init=start_t; % Start time for a given oscillation
olderr=0.0; % Controller error one sample time ago
max_dev=0.0; % Max. error for given oscillation from setpoint
start_flag = 0.0; % Flag to handle initial oscillation
tm_osc(1,1) = 0.0; % Total time oscillating during time window
num_osc(1,1) = 0.0; % Number of oscillations during time window
pc_tm_osc(1,1) = 0.0; % Percent time oscillating during time window
sm_pk_dev(1,1) = 0.0; % Sum of peak deviations during time window
av_pk_dev(1,1) = 0.0; % Average of peak deviations over time window
av_osc_freq(1,1) = 0.0; % Average half period oscillation freq. for time window
```

PMA_ALG.M (continued)

```

[m,n]=size(y1);
x(1,1)=start_t;

filtpv(1,1)=y1(1,1);
filtsp(1,1)=z(1,1);

for i=2:m; % Loop through entire data set sample by sample to perform calcs.

% Filter PV and SP as desired
filtpv(i,1)=(1-pvewmafilt)*filtpv(i-1,1)+ pvewmafilt*y1(i,1);
filtsp(i,1)=(1-spewmafilt)*filtsp(i-1,1)+ spewmafilt*z(i,1);

x(i,1)=x(i-1,1) + samp_time/3600; % Calculate time variable

looperr=((filtsp(i,1) - filtpv(i,1))/(pvhi - pvlo)) * 100.0; % Controller error

if looperr * olderr >= 0.0; % Still in given oscillation; controller error sign
    % is unchanged
    num_osc(i,1) = num_osc(i-1,1)*(1-(samp_time/(tm_range*3600))) + 0.0;
    tm_osc(i,1) = tm_osc(i-1,1)*(1-(samp_time/(tm_range*3600))) + 0.0;
    pc_tm_osc(i,1) = tm_osc(i,1)/(tm_range*60.0) * 100.0;
    sm_pk_dev(i,1) = sm_pk_dev(i-1,1)*(1-(samp_time/(tm_range*3600))) + 0.0;
    if abs(filtpv(i,1)-filtsp(i,1)) > max_dev;
        max_dev = abs(filtpv(i,1)-filtsp(i,1));
    end;

elseif max_dev > (pvhi-pvlo)*osc_base_lim; % Controller error sign change
    % and limit for counting as an
    % oscillation has been exceeded

    hlfrper_osc=x(i,1)-tosc_init;
    tosc_init=x(i,1);
    if start_flag < 0.5;
        num_osc(i,1) = num_osc(i-1,1)*(1-(samp_time/(tm_range*3600))) + 0.0;
        tm_osc(i,1) = tm_osc(i-1,1)*(1-(samp_time/(tm_range*3600))) + 0.0;
        pc_tm_osc(i,1) = tm_osc(i,1)/(tm_range*60.0) * 100.0;
        sm_pk_dev(i,1) = sm_pk_dev(i-1,1)*(1-(samp_time/(tm_range*3600))) + 0.0;
        start_flag = 1.0;
    else; % Countable oscillation detected, adjust variables appropriately
        num_osc(i,1) = num_osc(i-1,1)*(1-(samp_time/(tm_range*3600))) + 1.0;
        tm_osc(i,1) = tm_osc(i-1,1)*(1-(samp_time/(tm_range*3600))) + ...
            hlfrper_osc*60.0;
        pc_tm_osc(i,1) = tm_osc(i,1)/(tm_range*60.0) * 100.0;
        sm_pk_dev(i,1) = sm_pk_dev(i-1,1)*(1-(samp_time/(tm_range*3600))) + ...
            (max_dev/(pvhi-pvlo))*100;
    end;
    max_dev = 0.0;

else; % Controller error sign change, but no countable oscillation

    max_dev = 0.0;
    tosc_init=x(i,1);
    num_osc(i,1) = num_osc(i-1,1)*(1-(samp_time/(tm_range*3600))) + 0.0;
    tm_osc(i,1) = tm_osc(i-1,1)*(1-(samp_time/(tm_range*3600))) + 0.0;
    pc_tm_osc(i,1) = tm_osc(i,1)/(tm_range*60.0) * 100.0;
    sm_pk_dev(i,1) = sm_pk_dev(i-1,1)*(1-(samp_time/(tm_range*3600))) + 0.0;

end;

% Calculate average oscillation frequency and peak deviation
if num_osc(i,1) > 0.5;
    av_osc_fq(i,1) = tm_osc(i,1)/num_osc(i,1);
    av_pk_dev(i,1) = sm_pk_dev(i,1)/num_osc(i,1);
else;
    av_osc_fq(i,1) = av_osc_fq(i-1,1);
    av_pk_dev(i,1) = av_pk_dev(i-1,1);

```

PMA_ALG.M (continued)

```
end;

olderr=looperr; % Store current error as old error for next loop execution

end;

[ax,h1,h2]=plotyy(x,av_pk_dev,x,pc_tm_osc); % Plot % time oscillating and
hold on; % avg. peak deviation

% Plot oscillation limit (for counting as oscillation)
plot([min(x) max(x)],[(osc_base_lim*100) (osc_base_lim*100)],'m--');

ylabel('Avg. Peak % Deviation from SP (blue)'); % Plot labels
xlabel('Time (hours)');
axes(ax(2));
ylabel('% Time Oscillating (green)');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot rate of oscillation and average half period oscillation frequency
%
subplot(2,2,4);

[ax,h1,h2]=plotyy(x,num_osc,x,av_osc_fq); % Plot rate of oscillation and
% avg. oscillation frequency

ylabel('Rate of Oscillation (blue)'); % Plot labels
xlabel('Time (hours)');
axes(ax(2));
ylabel('Avg Half-osc Freq (minutes)');
```

CCORR_PLOT.M

```
%
% CCORR_PLOT.M
% Program to calculate the PV/OP cross correlation.
% High pass filtering of PV/OP data may be used
% as needed. Program produces 3 plots (SP/PV/OP,
% filtered PV/OP, PV/OP cross correlation).
%

%
% Get PV/OP data and plot
%
figure(1);
clf;
figure(2);
clf;
figure(1);
subplot(3,1,1);

clear x;
clear y1;
clear y2;
clear z;

ylcolumn=input('Column # for y1 data - PV: '); % User prompts
y2column=input('Column # for y2 data - OP: ');
zcolumn=input('Column # for z data - SP: ');
start_t=input('Enter start time: ');
t_length=input('Length of time: ');

b=eval(fname);
y1 = b( round((start_t*3600)/samp_time)+1: ...
        round((start_t*3600)/samp_time) + 1 + ...
        round((t_length*3600)/samp_time), ...
        ylcolumn);
y2 = b( round((start_t*3600)/samp_time)+1: ...
        round((start_t*3600)/samp_time) + 1 + ...
        round((t_length*3600)/samp_time), ...
        y2column);
z = b( round((start_t*3600)/samp_time)+1: ...
        round((start_t*3600)/samp_time) + 1 + ...
        round((t_length*3600)/samp_time), ...
        zcolumn);

[m,n]=size(y1); % Calculate time variable
x(1,1)=start_t;
for i = 2:m;
    x(i,1) = x(i-1,1) + samp_time/3600;
end;

[ax,h1,h2]=plotyy(x,y1,x,y2); % Plot PV and OP

hold on;
plot(x,z,'r--'); % Plot SP on same plot

set(get(ax(1),'xlabel'),'string','Time (hours)'); % Plot labels
set(get(ax(1),'ylabel'),'string','SP (red) and PV (blue)');
set(get(ax(2),'ylabel'),'string','OP (purple)');
set(h1,'linestyle','-','color','b');
set(h2,'linestyle',':','color','m');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% High pass filter the data and plot it
%
subplot(3,1,2);
clear y1flt;
clear y2flt;
```

CCORR_PLOT.M (continued)

```

hpfilt=input('Enter high pass filter in minutes: '); % User prompts

ylflt(1,1)=0.0;
y2flt(1,1)=0.0;
[m,n]=size(y1);
mny1 = mean(y1);
mny2 = mean(y2);
for i = 2:m;
    ylflt(i,1) = exp(-1.0*(samp_time/60.0)/hpfilt)*ylflt(i-1,1) ...
        + y1(i,1) - y1(i-1,1);
    y2flt(i,1) = exp(-1.0*(samp_time/60.0)/hpfilt)*y2flt(i-1,1) ...
        + y2(i,1) - y2(i-1,1);
end;

[ax,h1,h2]=plotyy(x,ylflt,x,y2flt); % Plot high pass filtered PV/OP data
hold on;

set(get(ax(1),'xlabel'),'string','Time (hours)'); % Plot labels
set(get(ax(1),'ylabel'),'string','PV (blue)');
set(get(ax(2),'ylabel'),'string','OP (purple)');
set(h1,'linestyle','-','color','b');
set(h2,'linestyle',':','color','m');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Actually find PV/OP cross correlations and plot
%
subplot(3,1,3);
clear c1;
clear c2;

plot_back=input('CC Time Back (min): '); % User prompts
plot_forw=input('CC Time Forward (min): ');

[m,n]=size(y1);

c1=xcov(ylflt,y2flt,'coeff'); % Call "xcov" to find correlations

c2(:,2) = c1(:,1);
c2(m,1) = 0.0;

for i = 1:m-1; % Calculate corresponding lag/lead times
    c2(m-i,1) = c2(m,1) - i * (samp_time/60);
    c2(m+i,1) = c2(m,1) + i * (samp_time/60);
end;

nsback = round( (plot_back*60) / samp_time ); % Plot ranges based on
nsforw = round( (plot_forw*60) / samp_time ); % user inputs

bar(c2((m-nsback:m+nsforw),1),c2((m-nsback:m+nsforw),2),'b'); % Use "bar"
                                                                % function to
                                                                % get fill effect

ylabel('Cross-correlation'); % Plot labels
xlabel...
('PV Lags OP (minutes) ----- PV Leads OP (minutes)');
set(gca,'Ylim',[-1,1]);
set(gca,'Xlim',[-1*plot_back,(1*plot_forw)]);
hold on;

plot([0 0 0],[-1 0 1],'r-'); % Line at t=0 to separate lag/lead
                             % portions of plot

mxc2=0.0;
mnc2=0.0;

for i = (m-nsback):(m+nsforw); % Find max and min correlation
    if c2(i,2) > mxc2; % values and corresponding times
        mxc2 = c2(i,2);
    end
end

```

CCORR_PLOT.M (continued)

```
        tmmxc2 = c2(i,1);
    end;
    if c2(i,2) < mnc2;
        mnc2 = c2(i,2);
        tmmnc2 = c2(i,1);
    end;
end;

set(gcf,'DefaultTextColor','green'); % More plot labels
out=[ num2str(tmmxc2,3) '/' num2str(mxc2,3)];
text((0.4*plot_forw),0.7,out);
out=[ num2str(tmmnc2,3) '/' num2str(mnc2,3)];
text((-0.9*plot_back),-0.7,out);
set(gcf,'DefaultTextColor','black');
```

ERROR_OSC_STATS.M

```

%
% ERROR_OSC_STATS.M
% Program to calculate controller error and oscillation
% frequency statistics.
% Program produces 2 SP/PV/OP plots to allow visual
% inspection. Program outputs various statistics for avg.
% error, peak error, and avg. oscillation frequencies.
%
%

figure(1);
clf;
figure(1);
subplot(2,1,1);
clear x;
clear y1;
clear y2;
clear z;

% User prompts
ylcolumn=input('Column # for y1 data - PV: '); % PV data
y2column=input('Column # for y2 data - OP: '); % OP data
zcolumn=input('Column # for z data - SP: '); % SP data
start_t=input('Enter start time: ');
t_length=input('Length of time: ');

b=eval(fname);
y1 = b( round((start_t*3600)/samp_time)+1: ...
        round((start_t*3600)/samp_time) + 1 + ...
        round((t_length*3600)/samp_time), ...
        ylcolumn);
y2 = b( round((start_t*3600)/samp_time)+1: ...
        round((start_t*3600)/samp_time) + 1 + ...
        round((t_length*3600)/samp_time), ...
        y2column);
z = b( round((start_t*3600)/samp_time)+1: ...
        round((start_t*3600)/samp_time) + 1 + ...
        round((t_length*3600)/samp_time), ...
        zcolumn);

[m,n]=size(y1);
x(1,1)=start_t; % Calculate time variable
for i = 2:m;
    x(i,1) = x(i-1,1) + samp_time/3600;
end;

[ax,h1,h2]=plotyy(x,y1,x,y2); % Plot PV and OP
hold on;

plot(x,z,'r--'); % Plot SP on same plot

set(get(ax(1),'xlabel'),'string','Time (hours)'); % Plot labels
set(get(ax(1),'ylabel'),'string','SP (red) and PV (blue)');
set(get(ax(2),'ylabel'),'string','OP (green)');
set(h1,'linestyle','-','color','b');
set(h2,'linestyle',':','color','g');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot SP/PV/OP data over short time range
%
subplot(3,1,2);
clear xzoom;
clear ylzoom;
clear y2zoom;
clear zzoom;

```

ERROR_OSC_STATS.M (continued)

```
% User prompts
st_time=input('Enter start time for data zoom (hours): ');
dur_time=input('Enter total plot time for data zoom (hours): ');

xzoom=x((st_time*3600)/samp_time:(st_time*3600)/samp_time + ...
        (dur_time*3600/samp_time));
ylzoom=y1((st_time*3600)/samp_time:(st_time*3600)/samp_time + ...
        (dur_time*3600/samp_time));
y2zoom=y2((st_time*3600)/samp_time:(st_time*3600)/samp_time + ...
        (dur_time*3600/samp_time));
zzoom=z((st_time*3600)/samp_time:(st_time*3600)/samp_time + ...
        (dur_time*3600/samp_time));

[ax,h1,h2]=plotyy(xzoom,ylzoom,xzoom,y2zoom); % Plot PV and OP
hold on;

plot(xzoom,zzoom,'r--'); % Plot SP on same plot

set(get(ax(1),'xlabel'),'string','Time (hours)'); % Plot labels
set(get(ax(1),'ylabel'),'string','SP (red) and PV (blue)');
set(get(ax(2),'ylabel'),'string','OP (green)');
set(h1,'linestyle','-','color','b');
set(h2,'linestyle',':','color','g');
title('SP/PV/OP Over Short Time Period');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate error and oscillation frequency statistics
%
%
clear filtpv;
clear filtsp;
clear filttop;
clear tm_osc;
clear num_osc;
clear pc_tm_osc;
clear sm_pk_dev;
clear av_pk_dev;
clear fl_av_pk_dev;
clear av_osc_fq;
clear hlfper_arr;
clear max_dev_pos_arr;
clear max_dev_neg_arr;
clear osc_pos_arr;
clear osc_neg_arr;
clear min_pv;
clear max_pv;
clear min_op;
clear max_op;
clear pkdev_arr_gn;
clear opmn_arr;
clear opmx_arr;
clear pvmn_arr;
clear pvmx_arr;

% User prompts
pvlo=input('Enter low PV limit: ');
pvhi=input('Enter high PV limit: ');
osc_base_lim=input('Enter low limit (fraction, eg 0.01) for osc. counting: ');
pvewmafilt=input('Enter ewma filter value for PV: ');
spewmafilt=input('Enter ewma filter value for SP: ');

% Initialize calculation variables
osc_count = 0;
hlfper_osc=0.0;
```

ERROR_OSC_STATS.M (continued)

```
tosc_init=start_t;
olderr=0.0;
max_dev=0.0;
start_flag = 0.0;

sum_neg_err = 0.0;
sum_pos_err = 0.0;
osc_pos_cnt = 0.0;
osc_neg_cnt = 0.0;
osc_tot_cnt = 0.0;

maxdev_pos_cnt = 0.0;
maxdev_neg_cnt = 0.0;
pos_err_count = 0.0;
neg_err_count = 0.0;

[m,n]=size(y1);
x(1,1)=start_t;

filtpv(1,1)=y1(1,1);
filtop(1,1)=y2(1,1);
filtsp(1,1)=z(1,1);

for i=2:m;

% Filter SP/PV/OP data as desired
filtop(i,1)=(1-pvewmafilt)*filtop(i-1,1)+ pvewmafilt*y2(i,1);
filtpv(i,1)=(1-pvewmafilt)*filtpv(i-1,1)+ pvewmafilt*y1(i,1);
filtsp(i,1)=(1-spewmafilt)*filtsp(i-1,1)+ spewmafilt*z(i,1);

x(i,1)=x(i-1,1) + samp_time/3600; % Calculate time variable

looperr=((filtsp(i,1) - filtpv(i,1))/(pvhi - pvlo)) * 100.0; % Controller error

if looperr > 0.0; % Sum positive errors (PV below SP)
    pos_err_count = pos_err_count + 1;
    sum_pos_err = sum_pos_err + abs(looperr);
end;

if looperr < 0.0; % Sum negative errors (PV above SP)
    neg_err_count = neg_err_count + 1;
    sum_neg_err = sum_neg_err + abs(looperr);
end;

if looperr * olderr >= 0.0; % Still on same side of controller error

    if abs(filtpv(i,1)-filtsp(i,1)) > max_dev; % Set max. deviations for each osc.
        max_dev = abs(filtpv(i,1)-filtsp(i,1));
    end;

elseif max_dev > (pvhi-pvlo)*osc_base_lim; % Limit exceeded for counting current
    hlfrper_osc=x(i,1)-tosc_init; % deviation in the statistics
    tosc_init=x(i,1);
    if start_flag < 0.5;
        start_flag = 1.0;
    else;
        osc_count = osc_count + 1;
        hlfrper_arr(osc_count) = hlfrper_osc*60.0;

        if olderr > 0.0; % Osc. limit exceeded, count as a positive error
            osc_pos_cnt = osc_pos_cnt + 1;
            osc_pos_arr(osc_pos_cnt) = hlfrper_osc;
            maxdev_pos_cnt = maxdev_pos_cnt + 1;
            max_dev_pos_arr(maxdev_pos_cnt) = (max_dev/(pvhi-pvlo))*100.0;
            osc_tot_cnt = osc_tot_cnt + 1;
        end;
    end;
end;
```

ERROR_OSC_STATS.M (continued)

```
if olderr < 0.0; % Osc. limit exceeded, count as a negative error
    osc_neg_cnt = osc_neg_cnt + 1;
    osc_neg_arr(osc_neg_cnt) = hlfper_osc;
    maxdev_neg_cnt = maxdev_neg_cnt + 1;
    max_dev_neg_arr(maxdev_neg_cnt) = (max_dev/(pvhi-pvlo))*100.0;
    osc_tot_cnt = osc_tot_cnt + 1;
end;

end;

max_dev = 0.0;

else; % Controller error sign change but limits not exceeded for counting
    max_dev = 0.0;
    tosc_init=x(i,1);
end;

olderr=looperr; % Set olderr equal to current error for the next
                % execution of the loop

end;

%
% Calculate/output statistics
%

dev_hld = [max_dev_pos_arr max_dev_neg_arr];
'[Avg. peak error (%)] = ',mean(dev_hld)

'[Avg. (+) peak error (%)] = ',mean(max_dev_pos_arr)

'[Avg. (-) peak error (%)] = ',mean(max_dev_neg_arr)

pct_pos = (sum_pos_err/pos_err_count)/mean(max_dev_pos_arr)*100.0;
'[Avg. (+) error / avg. (+) peak error (%)] = ',pct_pos

pct_neg = (sum_neg_err/neg_err_count)/mean(max_dev_neg_arr)*100.0;
'[Avg. (-) error / avg. (-) peak error (%)] = ',pct_neg

err_rat = mean(max_dev_pos_arr)/mean(max_dev_neg_arr);
'[Ratio of Avg. (+) peak error / avg. (-) peak error] = ',err_rat

osc_rat = mean(osc_pos_arr)/mean(osc_neg_arr);
'[Ratio of Avg. (+) osc. freq / avg. (-) osc. freq] = ',osc_rat

'[Avg. (+) osc. freq (min)] = ',mean(osc_pos_arr)

'[Avg. (-) osc. freq (min)] = ',mean(osc_neg_arr)

freq_mn = mean(hlfper_arr);
'[Avg. half period osc. freq (min)] = ',freq_mn
```

PV_OP_SPREAD_STATS.M

```

%
% PV_OP_SPREAD_STATS.M
% Program to calculate the PV/OP spread in %/%.
% OP and PV data are divided into sections and the relative
% variations are calculated for each section.
%

figure(1);
clf;
figure(1);
subplot(2,1,1);

clear x;           % Time variable
clear y1;          % PV variable
clear y2;          % OP variable
clear z;           % SP variable

y1column=input('Column # for y1 data - PV: '); % User prompts
y2column=input('Column # for y2 data - OP: ');
zcolumn=input('Column # for z data - SP: ');
start_t=input('Enter start time: ');
t_length=input('Length of time: ');

b=eval(fname);
y1 = b( round((start_t*3600)/samp_time)+1: ...
        round((start_t*3600)/samp_time) + 1 + ...
        round((t_length*3600)/samp_time), ...
        y1column);
y2 = b( round((start_t*3600)/samp_time)+1: ...
        round((start_t*3600)/samp_time) + 1 + ...
        round((t_length*3600)/samp_time), ...
        y2column);
z = b( round((start_t*3600)/samp_time)+1: ...
        round((start_t*3600)/samp_time) + 1 + ...
        round((t_length*3600)/samp_time), ...
        zcolumn);

[m,n]=size(y1);
x(1,1)=start_t;      % Calculate time variable
for i = 2:m;
    x(i,1) = x(i-1,1) + samp_time/3600;
end;

[ax,h1,h2]=plotyy(x,y1,x,y2); % Plot PV and OP

hold on;
plot(x,z,'r--');          % Plot SP on same plot

set(get(ax(1),'xlabel'),'string','Time (hours)'); % Plot labels
set(get(ax(1),'ylabel'),'string','SP (red) and PV (blue)');
set(get(ax(2),'ylabel'),'string','OP (purple)');
set(h1,'linestyle','-','color','b');
set(h2,'linestyle',':','color','m');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Perform calculation to find spread of PV relative to OP
%
clear opbin;
clear binmin;
clear binmax;
clear pvopsp;

numbinspv=input('Enter number of bins for PV spread calc: ');

[m,n]=size(y1);

```

PV_OP_SPREAD_STATS.M (continued)

```

mxop = max(y2);
mnop = min(y2);
mxpv = max(y1);
mnpv = min(y1);
opstep = (mxop-mnop)/numbinspv;    % Find OP step size

for i=1:numbinspv;
    opbin(i) = mnop+ (i*opstep);
    binmin(i) = 10000000;
    binmax(i) = -10000000;
end;

for pp = 1:numbinspv;    % Find max and min PV values for each OP section (bin)
    for i=1:m;
        if y2(i,1) <= opbin(pp);
            if y2(i,1) > ( opbin(pp)-opstep ) ;
                if y1(i,1) < binmin(pp);
                    binmin(pp) = y1(i,1);
                end;
                if y1(i,1) > binmax(pp);
                    binmax(pp) = y1(i,1);
                end;
            end;
        end;
    end;
end;

for i=1:numbinspv;
    pvopsp(i) = ( (binmax(pp)-binmin(pp))/(mxpv-mnpv) *100.0 )/ (100.0/numbinspv);
end;

pvopspread = mean(pvopsp);

'[PV/OP Spread Index (unfiltered data) %] = ', (pvopspread/numbinspv) * 100.0

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Perform similar calculation to find spread of OP relative to PV
%
clear pvbin;
clear binmin;
clear binmax;
clear oppvsp;

mxop = max(y2);
mnop = min(y2);
mxpv = max(y1);
mnpv = min(y1);
pvstep = (mxpv-mnpv)/numbinspv;    % PV step size

for i=1:numbinspv;
    pvbin(i) = mnpv+ (i*pvstep);
    binmin(i) = 10000000;
    binmax(i) = -10000000;
end;

for pp = 1:numbinspv;    % Find max and min OP values for each PV section (bin)
    for i=1:m;
        if y1(i,1) <= pvbin(pp);
            if y1(i,1) > ( pvbin(pp)-pvstep ) ;
                if y2(i,1) < binmin(pp);
                    binmin(pp) = y2(i,1);

```

PV_OP_SPREAD_STATS.M (continued)

```

        end;
        if y2(i,1) > binmax(pp);
            binmax(pp) = y2(i,1);
        end;
    end;
end;
end;

end;

for i=1:numbinspv;
    oppvsp(i) = ( (binmax(pp)-binmin(pp))/(mxop-mnop) *100.0 )/ (100.0/numbinspv);
end;

oppvspread = mean(oppvsp);

'[OP/PV Spread Index (unfiltered data) %] = ',(oppvspread/numbinspv) * 100.0

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot PV vs. OP to view shape and spread visually
subplot(2,1,2);
plot(y2,y1,'ro');
```

PCA_ERR_OP_SCORES.M

```
%
% PCA_ERR_OP_SCORES.M
% Program to project scores for new data onto
% existing PCA model for base case data
% (data for good performance).
%
% Program produces scores plot with 95%
% confidence limits, via interactive user
% window.
%

% User prompts
bspvcol=input('Enter column number for PV (base data): ');
bsopcol=input('Enter column number for OP (base data): ');
bsspcol=input('Enter column number for SP (base data): ');
newfile=input('Enter name for new file (to project onto existing model): ');
nwpvcol=input('Enter column number for PV (new data): ');
nwopcol=input('Enter column number for OP (new data): ');
nwspcol=input('Enter column number for SP (new data): ');

bf=eval(basefile);          % Get base data
base_pv_data=bf(:,bspvcol);
base_op_data=bf(:,bsopcol);
base_sp_data=bf(:,bsspcol);

nf=eval(newfile);          % Get new data
new_pv_data=nf(:,nwpvcol);
new_op_data=nf(:,nwopcol);
new_sp_data=nf(:,nwspcol);

base_err = base_sp_data - base_pv_data;    % Calculate controller error
new_err = new_sp_data - new_pv_data;

bs_mat = [base_err base_op_data];          % Form base matrix for error/op
nw_mat = [new_err new_op_data];            % Form new matrix for error/op

[sc_bs_mat,mx_bs,stdx_bs]=auto(bs_mat);     % Scale base data to mean zero
                                           % and unit variance

[scores,loads,ssq,res,q,tsq]=pca(sc_bs_mat,0,[],2); % Calculate PCA model
                                           % for base data

sc_nw_mat = auto(nw_mat);                  % Scale new data to mean zero and unit variance

% Project new scores data onto PCA model for base data
[nw_scores,nw_res,nw_tsq]=pcapro(sc_nw_mat,loads,ssq,q,tsq,0);

% Plot the scores
scrpltr(nw_scores,ssq);
```

PCA_PV_OP_MODEL.M

```
%
% PCA_PV_OP_MODEL.M
% Program to fit PCA model to a time-shifted
% PV/OP data matrix.
% The percent variance captured by each of the
% 5 principal components is output.
%
clear nw_mat;           % Clear matrices
clear sc_nw_mat;

% User prompts
newfile=input('Enter name for new file: ');
samp_time=input('Enter sample time for new data in seconds: ');
nwpvcol=input('Enter column number for PV (new data): ');
nwopcol=input('Enter column number for OP (new data): ');
sec_lag=input('Enter total time for PV/OP lag in seconds: ');

nf=eval(newfile);
new_pv_data=nf(:,nwpvcol);
new_op_data=nf(:,nwopcol);

% Create time-shifted matrix
[m,n]=size(new_pv_data);
lg_delt = round(sec_lag/samp_time);

for i =1:(lg_delt+1);
    nw_mat(:,i)=new_pv_data(i:m-lg_delt+(i-1));
    nw_mat(:,i+lg_delt+1)=new_op_data(i:m-lg_delt+(i-1));
end;

sc_nw_mat = auto(nw_mat); % Scale to mean zero and unit variance

[scores,loads,ssq,res,q,tsq]=bigpca(sc_nw_mat,5,0); % Calculate PCA model
```

APPENDIX D

PCA COMPARISONS FOR SIMULATED AND REAL PROBLEM DATA SETS
(RELATIVE TO GOOD PERFORMANCE)

Appendix D contains the scores plots for projection of controller error/OP data (for the 4 common problems) onto PCA models for good performance.

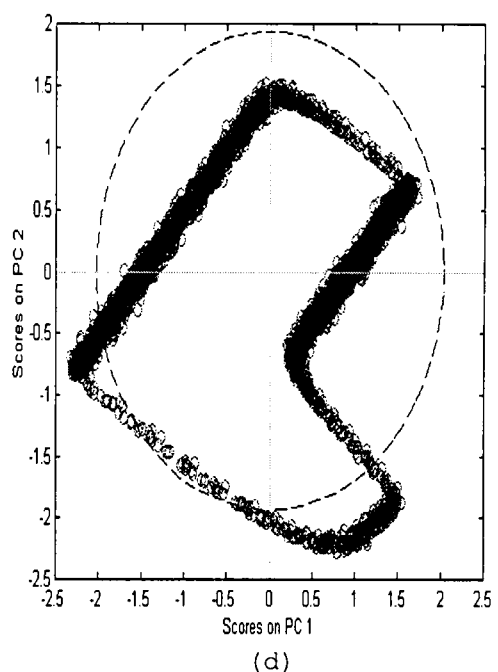
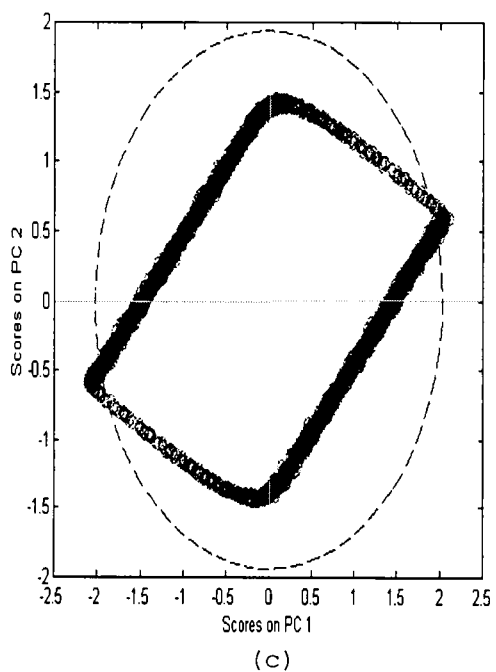
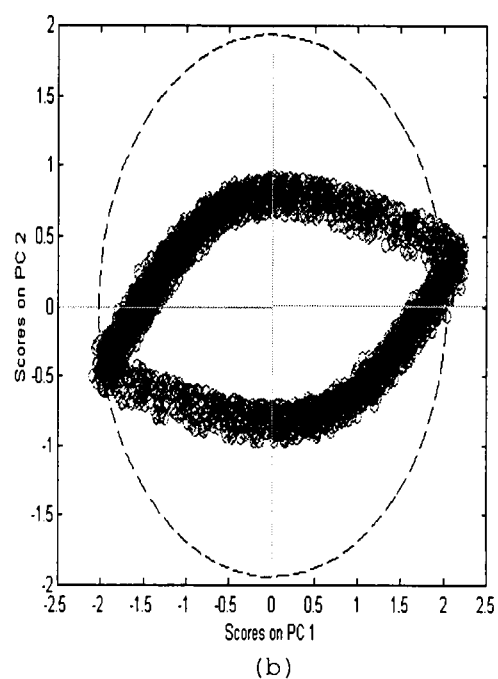
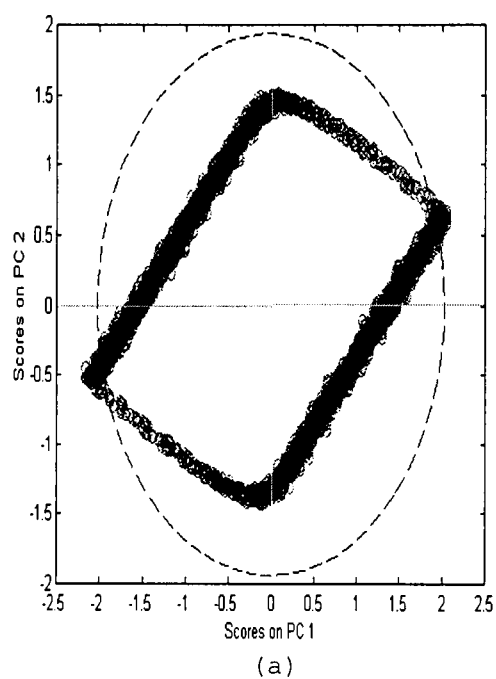
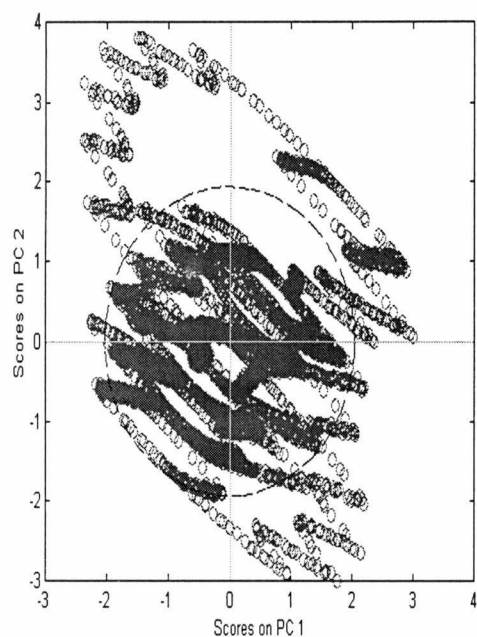
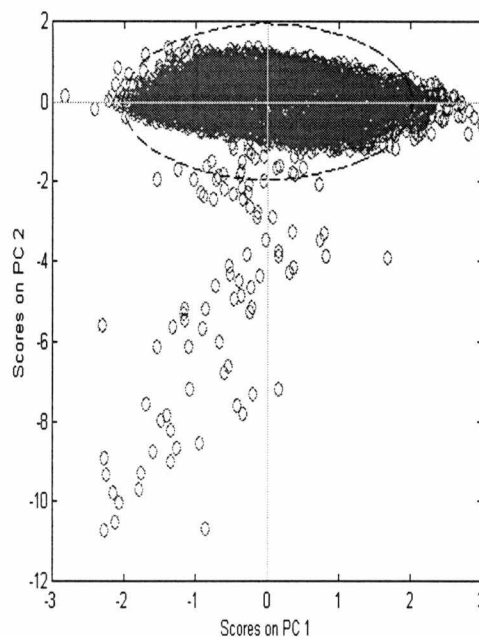


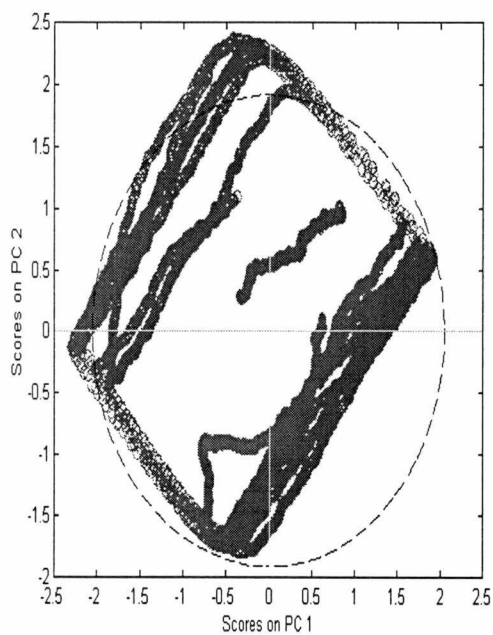
Figure D.1. Scores plots for stiction data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for STIC1S, (b) scores for STIC2S, (c) scores for STIC3S, (d) scores for STIC4S.



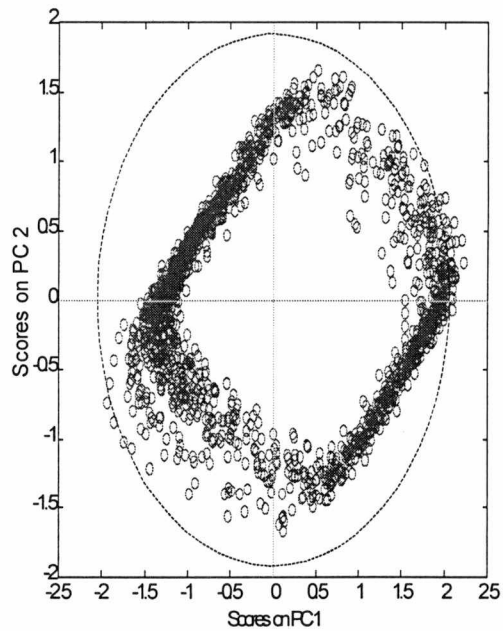
(a)



(b)



(c)



(d)

Figure D.2. Scores plots for stiction data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for STIC5S, (b) scores for STIC6S, (c) scores for STIC1R, (d) scores for STIC2R.

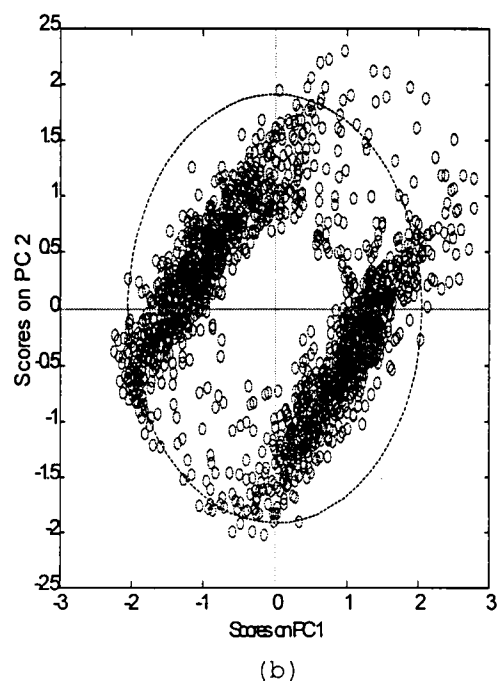
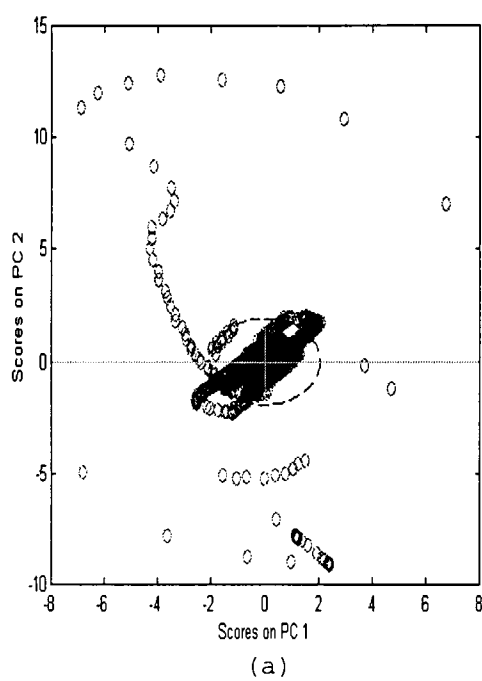


Figure D.3. Scores plots for stiction data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for STIC3R, (b) scores for STIC4R.

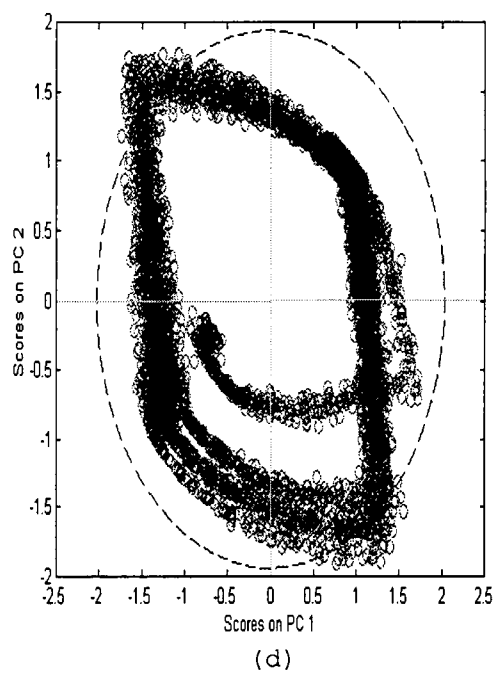
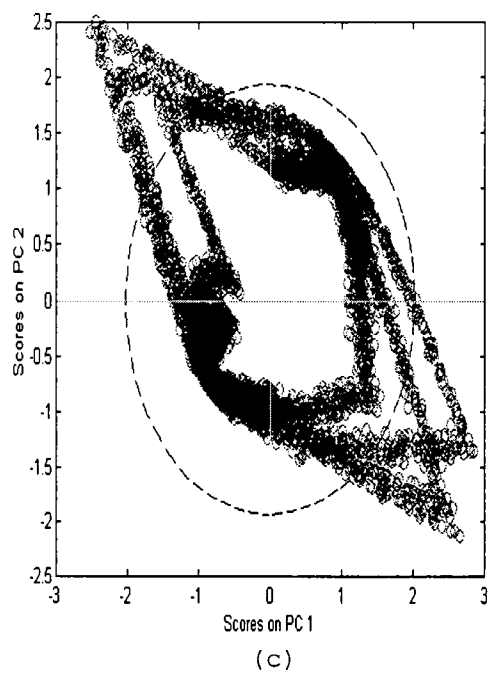
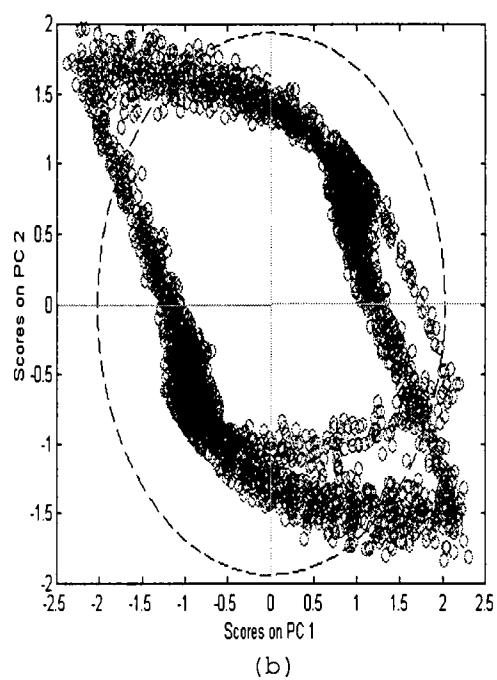
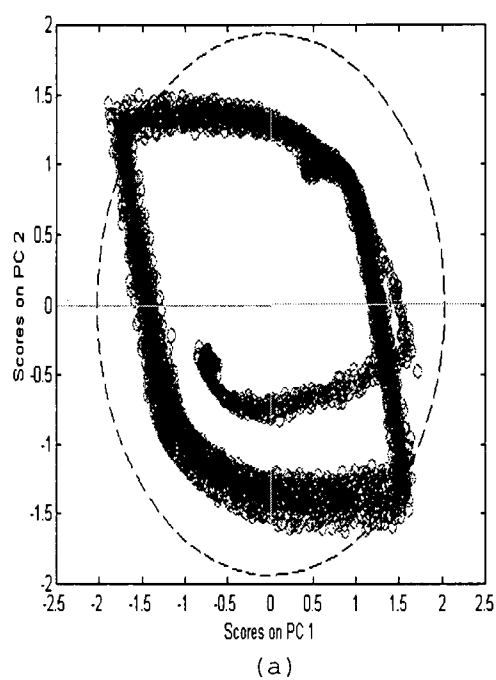
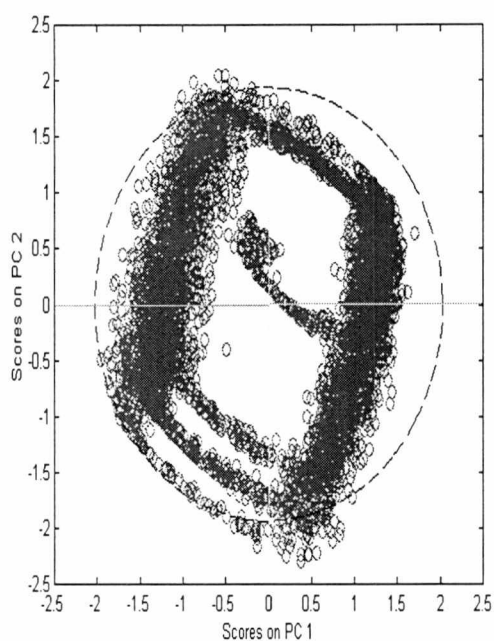
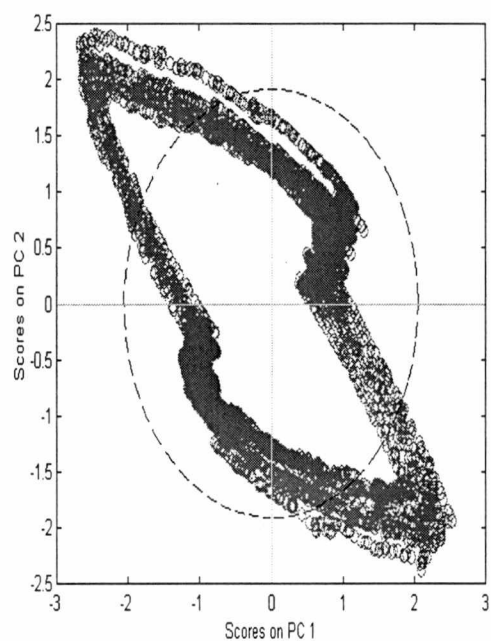


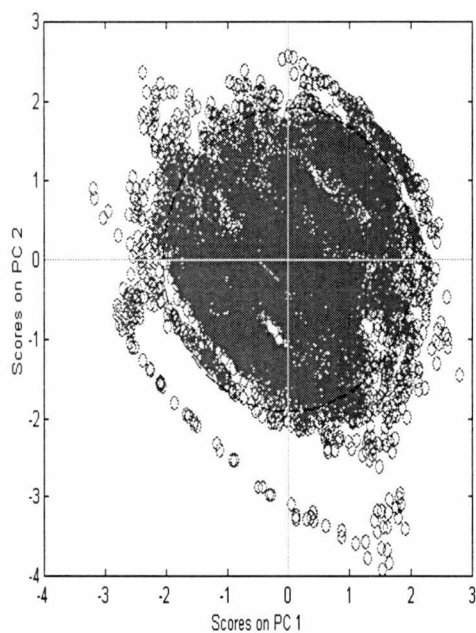
Figure D.4. Scores plots for hysteresis data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for HYST1S, (b) scores for HYST2S, (c) scores for HYST3S, (d) scores for HYST4S.



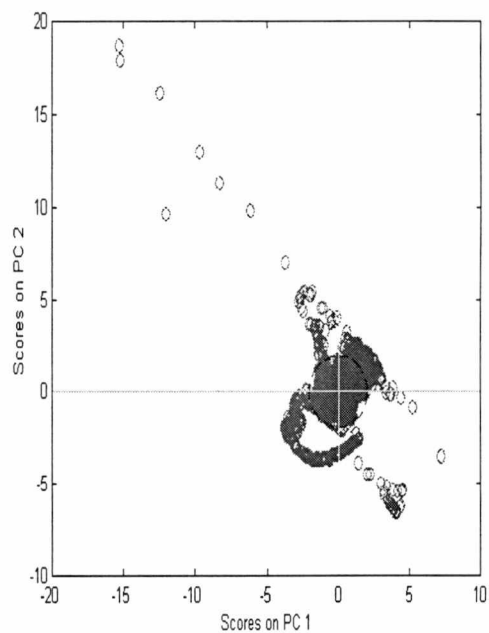
(a)



(b)

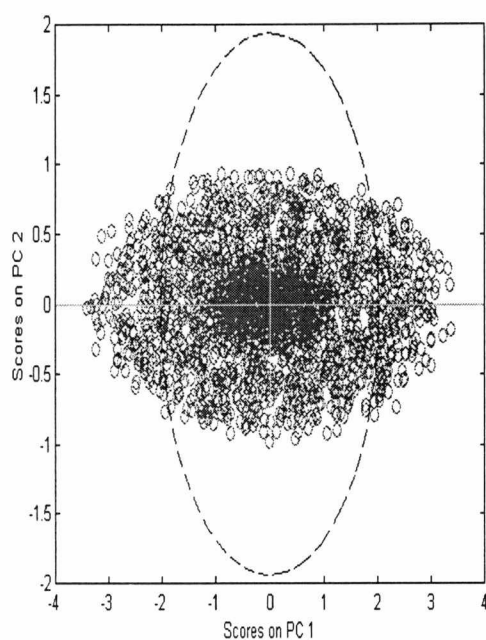


(c)

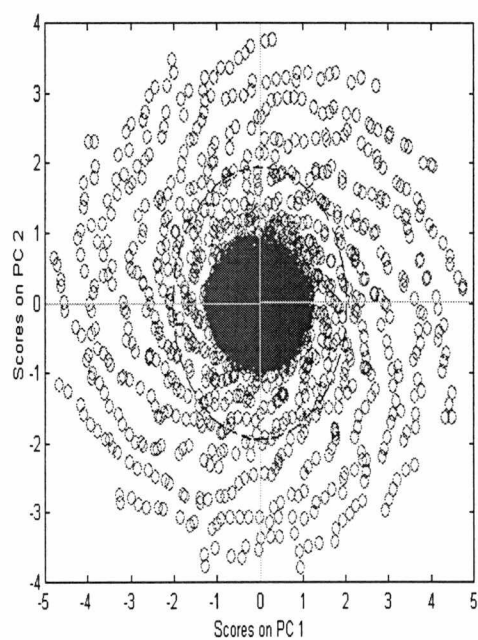


(d)

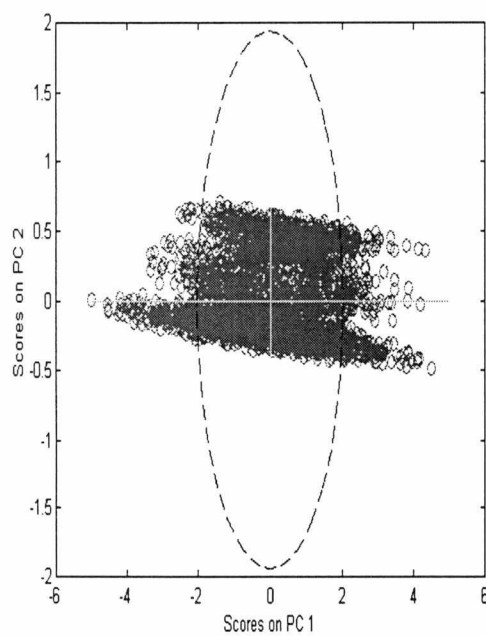
Figure D.5. Scores plots for hysteresis data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for HYST5S, (b) scores for HYST1R, (c) scores for HYST2R, (d) scores for HYST3R.



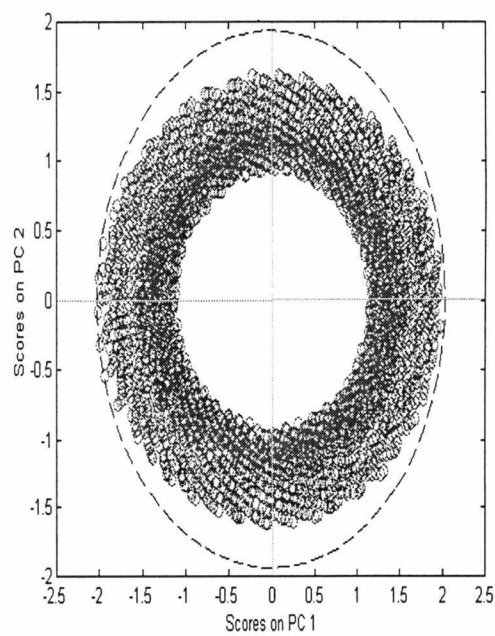
(a)



(b)



(c)



(d)

Figure D.6. Scores plots for tuning data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for TUNE1S, (b) scores for TUNE2S, (c) scores for TUNE3S, (d) scores for TUNE4S.

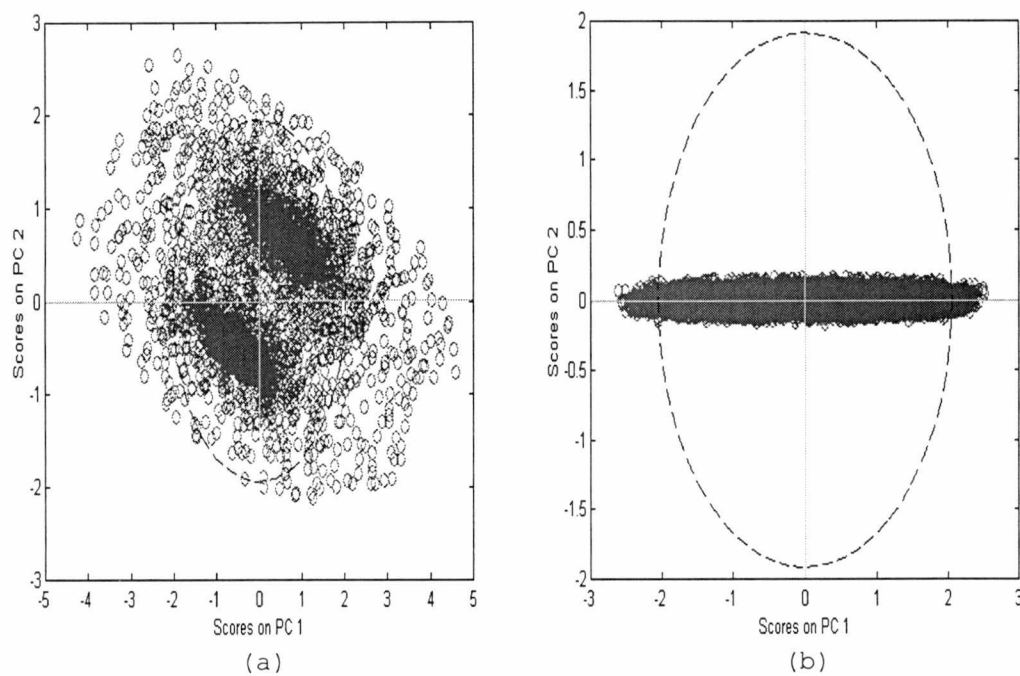
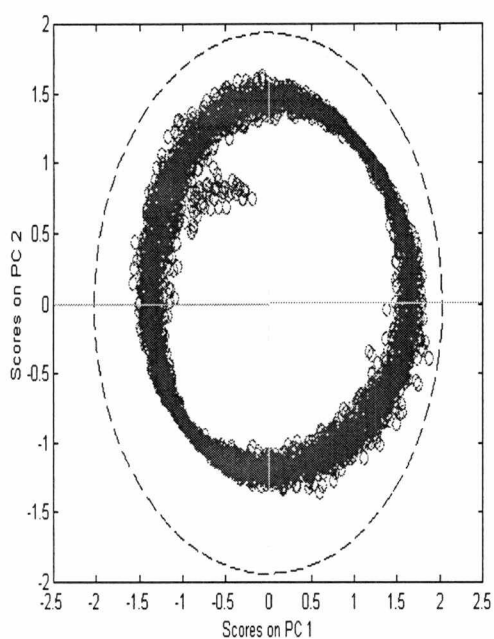
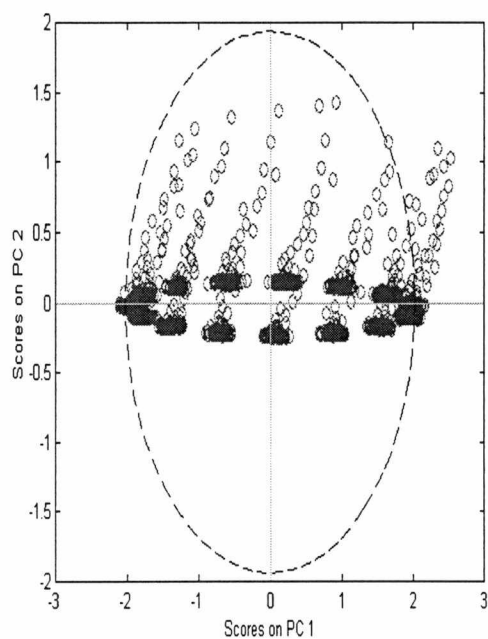


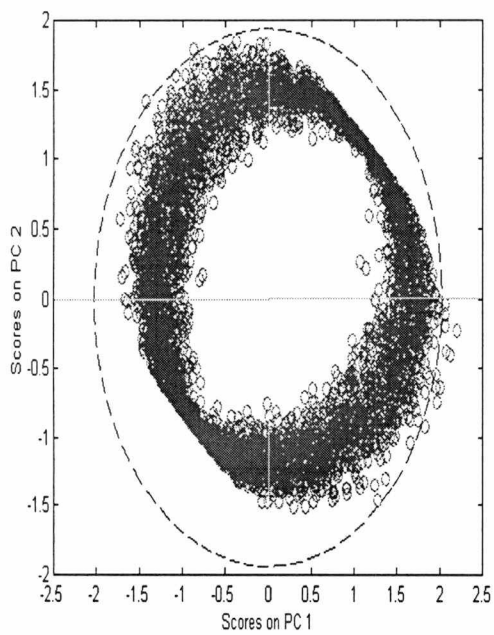
Figure D.7. Scores plots for tuning data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for TUNE5S, (b) scores for TUNE1R.



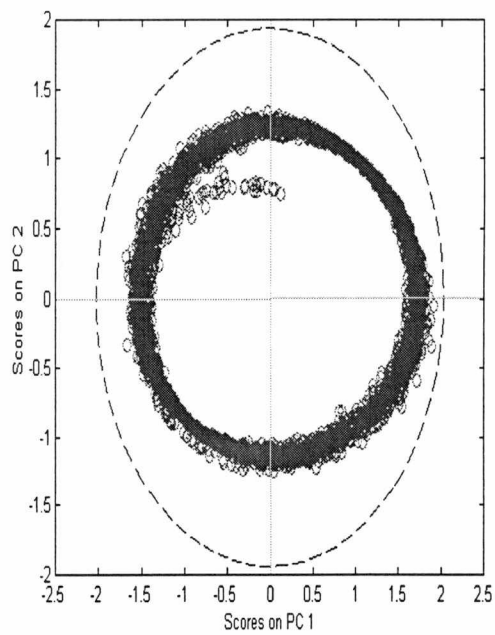
(a)



(b)



(c)



(d)

Figure D.8. Scores plots for disturbance data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for DIST1S, (b) scores for DIST2S, (c) scores for DIST3S, (d) scores for DIST4S.

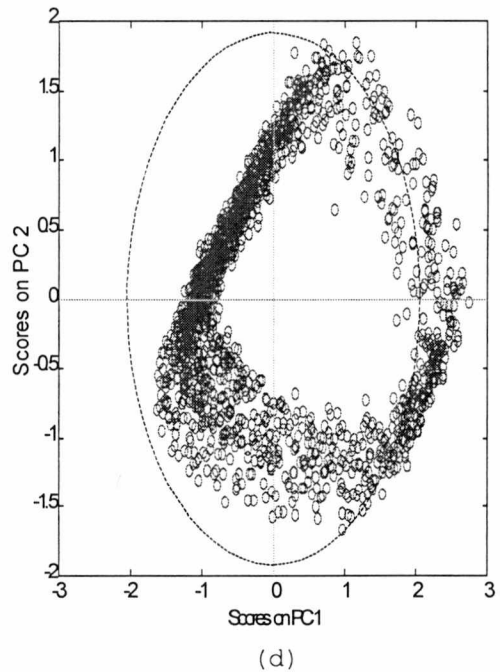
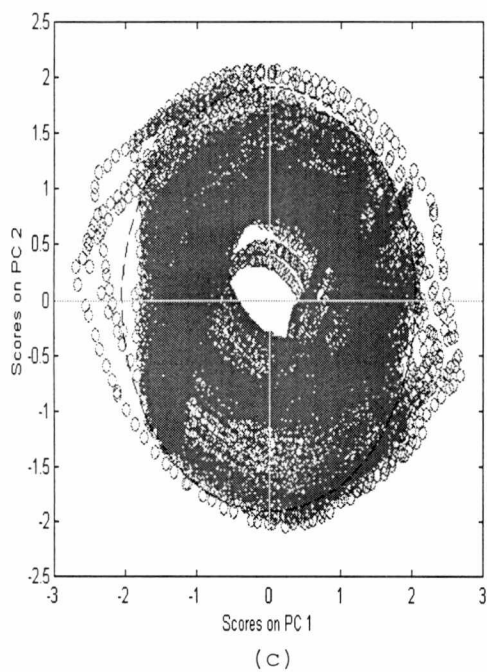
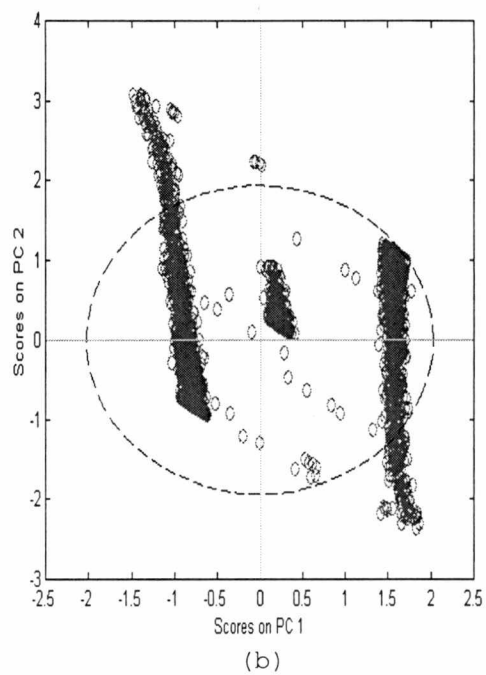
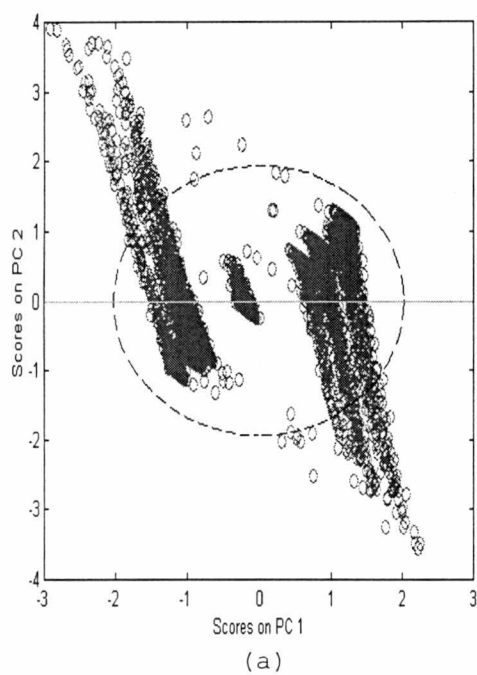


Figure D.9. Scores plots for disturbance data projected onto PCA models for good performance, with 95% confidence limits: (a) scores for DIST5S, (b) scores for DIST6S, (c) scores for DIST1R, (d) scores for DIST2R.

VITA

John Wesley Cox was born in Bristol, Tennessee on April 30, 1969. He attended elementary and middle schools in Bluff City, Tennessee and graduated from Sullivan East High School in June 1987. In September 1987, he entered Tennessee Technological University, and in May 1992 he received a Bachelor of Science degree in Chemical Engineering.

Upon graduation from Tennessee Technological University, he began work as a chemical engineer in the Advanced Controls Technology Group of Eastman Chemical's Engineering and Construction Division located in Kingsport, Tennessee. In 1993, he entered the University of Tennessee Graduate School to pursue a graduate degree in Chemical Engineering; he received the Master of Science degree in Chemical Engineering in December 1998.