MODELING THE DYNAMIC RESPONSE OF PRESSURES IN A DISTRIBUTED HELIUM REFRIGERATION SYSTEM

BY

JOHN CARL BRUBAKER

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering
in the Graduate College of the
Illinois Institute of Technology

Approved _____
Adviser

TABLE OF CONTENTS

	Page
LIST OF TABLES	v
LIST OF FIGURES	vi
LIST OF SYMBOLS	xiv
ABSTRACT	xix
CHAPTER	
I. INTRODUCTION	1
1.1 The Refrigeration System	2
1.2 Goals of the Investigation	8
II. DESIGN AND PERFORMING OF EXPERIMENTS	13
2.1 Observations on the System Dynamics	13
2.2 Suitability of a Linear Model	14
2.3 Choice of Recorded Variables	17
2.4 Choice of Experimental Parameters	18
2.5 Choice of Sample Time and Test Duration	23
2.6 Performing the Experiment	25
III. PROCEDURES FOR DATA ANALYSIS AND SYSTEM MODELING	28
3.1 Linear Time Invariant Parametric Models	29
3.2 Black Box Modeling	34
3.3 Black Box Model for Compressor Flow Inputs	37
3.4 Black Box Model for Inventory Flow Inputs	53
3.5 White Box Model for Compressor Flow Inputs	65
IV. MODELING RESULTS AND DISCUSSION	80
4.1 Black Box Results for Compressor Flow Inputs	80
4.2 Black Box Results for Inventory Flow Inputs	108
4.3 Averaged and Validated Black Box System Model	111
4.4 Detailing System Feedback using White Box Model	148
4.5 Disturbance Characterizations	157
4.6 Recommendations for Control Improvements	164

Page

CHAPTER

V. SUMMARY AND CONCLUSION 168

5.1 Summary of Investigation Goals and Techniques 168
5.2 Responses to Controllable Inputs 169
5.3 Uses for the Modeling Results 171

APPENDIX

A. PROGRAMS FOR PRODUCING MODELS AND FIGURES 174

A.1 Matlab Program ModF2.m 175
A.2 Matlab Program ModFC.m 189
A.3 Matlab Program SimF2.m 199
A.4 Matlab Program SimFC.m 209
A.5 Matlab Program Micro1mod.m 217
A.6 Matlab Program Micro2mod.m 235

BIBLIOGRAPHY 253

LIST OF TABLES

Table	Page
1. Three Data Sets of Compressor Flow vs. Loader Valve Position	22
2. Modeling Input Parameters and Sum of Squared Errors Values	81
3. Location Averaged Model Coefficients	112
4. P1 and P2 Model Coefficients for Secondary Flow Input	152
5. P1 and P2 Model Coefficients for Primary Flow Input	153
6. P1 Model Coefficients for A3 F3 Flow Input	160
7. P1 and P2 Model Coefficients for Ambient Temp. Input . . .	163

LIST OF FIGURES

Figure	Page
1. Tevatron Refrigeration System	3
2. Compressor Flow vs. Loader Valve Position	22
3. SISO System Block Diagram	30
4. B0 F2 Flow and FS Flow from Data Set 3	38
5. Sum of B0 F2 Flow and FS Flow from Data Set 3	38
6. A0 P1 Data, Filtered Data, and Model for Step in B0 F2	39
7. A0 P2 Data, Filtered Data, and Model for Step in B0 F2	39
8. B0 P1 Data, Filtered Data, and Model for Step in B0 F2	40
9. B0 P2 Data, Filtered Data, and Model for Step in B0 F2	40
10. BA P1 Data, Filtered Data, and Model for Step in B0 F2	41
11. BA P2 Data, Filtered Data, and Model for Step in B0 F2	41
12. C0 P1 Data, Filtered Data, and Model for Step in B0 F2	42
13. C0 P2 Data, Filtered Data, and Model for Step in B0 F2	42
14. D0 P1 Data, Filtered Data, and Model for Step in B0 F2	43
15. D0 P2 Data, Filtered Data, and Model for Step in B0 F2	43
16. E0 P1 Data, Filtered Data, and Model for Step in B0 F2	44
17. E0 P2 Data, Filtered Data, and Model for Step in B0 F2	44
18. EA P1 Data, Filtered Data, and Model for Step in B0 F2	45
19. EA P2 Data, Filtered Data, and Model for Step in B0 F2	45
20. F0 P1 Data, Filtered Data, and Model for Step in B0 F2	46
21. F0 P2 Data, Filtered Data, and Model for Step in B0 F2	46

Figure Page

22. SY P1 Data, Filtered Data, and Model for Step in B0 F2 47

23. SY P2 Data, Filtered Data, and Model for Step in B0 F2 47

24. FC Inventory Flow from Data Set 22 54

25. FD Inventory Flow from Data Set 22 54

26. Sum of F2 Flows from Data Set 22 55

27. Net Flow Change into Discharge Header from Data Set 22 . . 55

28. A0 P1 Data for Step in FC 56

29. A0 P2 Data, Filtered Data, and Model for Step in FC 56

30. B0 P1 Data for Step in FC 57

31. B0 P2 Data, Filtered Data, and Model for Step in FC 57

32. BA P1 Data for Step in FC 58

33. BA P2 Data, Filtered Data, and Model for Step in FC 58

34. C0 P1 Data for Step in FC 59

35. C0 P2 Data, Filtered Data, and Model for Step in FC 59

36. D0 P1 Data for Step in FC 60

37. D0 P2 Data, Filtered Data, and Model for Step in FC 60

38. E0 P1 Data for Step in FC 61

39. E0 P2 Data, Filtered Data, and Model for Step in FC 61

40. EA P1 Data for Step in FC 62

41. EA P2 Data, Filtered Data, and Model for Step in FC 62

42. F0 P1 Data for Step in FC 63

43. F0 P2 Data, Filtered Data, and Model for Step in FC 63

Figure Page

44. SY P1 Data for Step in FC 64

45. SY P2 Data, Filtered Data, and Model for Step in FC 64

46. Primary Flow and Secondary Flow from Data Set 5 69

47. Sum of Primary Flow and Secondary Flow from Data Set 5 . . 69

48. A0 P1 Data and Combined Flow Model from Data Set 5 70

49. A0 P1 Data and Separate Flow Model from Data Set 5 70

50. A0 P2 Data and Combined Flow Model from Data Set 5 73

51. A0 P2 Data and Separate Flow Model from Data Set 5 73

52. A0 P1 Combined Flow Model Components from Data Set 5 . . 74

53. A0 P1 Separate Flow Model Components from Data Set 5 . . 74

54. A0 P2 Combined Flow Model Components from Data Set 3 . . 76

55. A0 P2 Separate Flow Model Components from Data Set 3 . . 76

56. P1 Primary Components for Combined Model from Data Set 5 77

57. P1 Secondary Components for Combined Model from Data Set 5 77

58. P1 Primary Components for Separate Model from Data Set 5 . 78

59. P1 Secondary Components for Separate Model from Data Set 5 78

60. P2 Primary Components for Separate Model from Data Set 3 . 79

61. P2 Secondary Components for Separate Model from Data Set 3 79

62. P1 Model for Unit Step in A0 F2 Based on Data Set 1 89

63. P2 Model for Unit Step in A0 F2 Based on Data Set 1 89

64. P1 Model for Unit Step in A0 F2 Based on Data Set 2 90

65. P2 Model for Unit Step in A0 F2 Based on Data Set 2 90

Figure Page

66. P1 Model for Unit Step in B0 F2 Based on Data Set 3 91

67. P2 Model for Unit Step in B0 F2 Based on Data Set 3 91

68. P1 Model for Unit Step in B0 F2 Based on Data Set 4 92

69. P2 Model for Unit Step in B0 F2 Based on Data Set 4 92

70. P1 Model for Unit Step in B0 F2 Based on Data Set 5 93

71. P2 Model for Unit Step in B0 F2 Based on Data Set 5 93

72. P1 Model for Unit Step in B0 F2 Based on Data Set 6 94

73. P2 Model for Unit Step in B0 F2 Based on Data Set 6 94

74. P1 Model for Unit Step in C0 F2 Based on Data Set 7 95

75. P2 Model for Unit Step in C0 F2 Based on Data Set 7 95

76. P1 Model for Unit Step in C0 F2 Based on Data Set 8 96

77. P2 Model for Unit Step in C0 F2 Based on Data Set 8 96

78. P1 Model for Unit Step in D0 F2 Based on Data Set 9 97

79. P2 Model for Unit Step in D0 F2 Based on Data Set 9 97

80. P1 Model for Unit Step in D0 F2 Based on Data Set 10 98

81. P2 Model for Unit Step in D0 F2 Based on Data Set 10 98

82. P1 Model for Unit Step in D0 F2 Based on Data Set 11 99

83. P2 Model for Unit Step in D0 F2 Based on Data Set 11 99

84. P1 Model for Unit Step in D0 F2 Based on Data Set 12 100

85. P2 Model for Unit Step in D0 F2 Based on Data Set 12 100

86. P1 Model for Unit Step in E0 F2 Based on Data Set 13 101

87. P2 Model for Unit Step in E0 F2 Based on Data Set 13 101

Figure Page

88. P1 Model for Unit Step in E0 F2 Based on Data Set 14 102

89. P2 Model for Unit Step in E0 F2 Based on Data Set 14 102

90. P1 Model for Unit Step in F0 F2 Based on Data Set 15 103

91. P2 Model for Unit Step in F0 F2 Based on Data Set 15 103

92. P1 Model for Unit Step in F0 F2 Based on Data Set 16 104

93. P2 Model for Unit Step in F0 F2 Based on Data Set 16 104

94. P1 Model for Unit Step in F0 F2 Based on Data Set 17 105

95. P2 Model for Unit Step in F0 F2 Based on Data Set 17 105

96. P1 Model for Unit Step in F0 F2 Based on Data Set 18 106

97. P2 Model for Unit Step in F0 F2 Based on Data Set 18 106

98. P2 Model for Unit Step in FC Based on Data Set 19 109

99. P2 Model for Unit Step in FC Based on Data Set 20 109

100. P2 Model for Unit Step in FC Based on Data Set 21 110

101. P2 Model for Unit Step in FC Based on Data Set 22 110

102. Averaged P1 Model for Unit Step in A0 F2 116

103. Averaged P2 Model for Unit Step in A0 F2 116

104. Averaged P1 Model for Unit Step in B0 F2 117

105. Averaged P2 Model for Unit Step in B0 F2 117

106. Averaged P1 Model for Unit Step in C0 F2 118

107. Averaged P2 Model for Unit Step in C0 F2 118

108. Averaged P1 Model for Unit Step in D0 F2 119

109. Averaged P2 Model for Unit Step in D0 F2 119

Figure Page

110. Averaged P1 Model for Unit Step in E0 F2 120

111. Averaged P2 Model for Unit Step in E0 F2 120

112. Averaged P1 Model for Unit Step in F0 F2 121

113. Averaged P2 Model for Unit Step in F0 F2 121

114. Averaged P2 Model for Unit Step in Inventory Flow 122

115. B0 F2 Flow and Secondary Flow from Data Set 23 124

116. Sum of B0 F2 Flow and Secondary Flow from Data Set 23 . . . 124

117. B0 P1 Data and Simulation for Step in B0 F2 125

118. B0 P2 Data and Simulation for Step in B0 F2 125

119. E0 P1 Data and Simulation for Step in B0 F2 126

120. E0 P2 Data and Simulation for Step in B0 F2 126

121. E0 F2 Flow and Secondary Flow from Data Set 24 127

122. Sum of E0 F2 Flow and Secondary Flow from Data Set 24 . . . 127

123. B0 P1 Data and Simulation for Step in E0 F2 128

124. B0 P2 Data and Simulation for Step in E0 F2 128

125. E0 P1 Data and Simulation for Step in E0 F2 129

126. E0 P2 Data and Simulation for Step in E0 F2 129

127. F0 F2 Flow and Secondary Flow from Data Set 25 131

128. Sum of F0 F2 Flow and Secondary Flow from Data Set 25 . . . 131

129. C0 P1 Data and Simulation for Step in F0 F2 132

130. C0 P2 Data and Simulation for Step in F0 F2 132

131. F0 P1 Data and Simulation for Step in F0 F2 133

Figure Page

132. F0 P2 Data and Simulation for Step in F0 F2 133

133. F0 F2 Flow and Secondary Flow from Data Set 26 134

134. Sum of F0 F2 Flow and Secondary Flow from Data Set 26 . . . 134

135. C0 P1 Data and Simulation for Step in F0 F2 135

136. C0 P2 Data and Simulation for Step in F0 F2 135

137. F0 P1 Data and Simulation for Step in F0 F2 136

138. F0 P2 Data and Simulation for Step in F0 F2 136

139. FC Inventory Flow from Data Set 27 138

140. FD Inventory Flow from Data Set 27 138

141. Sum of F2 Flows from Data Set 27 139

142. Net Flow Change into Discharge Header from Data Set 27 . . 139

143. B0 P1 Data for Step in FC 140

144. B0 P2 Data and Simulation for Step in FC 140

145. E0 P1 Data for Step in FC 141

146. E0 P2 Data and Simulation for Step in FC 141

147. FC Inventory Flow from Data Set 28 143

148. FD Inventory Flow from Data Set 28 143

149. Sum of F2 Flows from Data Set 28 144

150. Net Flow Change into Discharge Header from Data Set 28 . . 144

151. B0 P1 Data for Step in FC 145

152. B0 P2 Data and Simulation for Step in FC 145

153. E0 P1 Data for Step in FC 146

Figure Page

154. E0 P2 Data and Simulation for Step in FC 146

155. Black Box System Block Diagram for Compressor Flow Inputs 147

156. P1 Components for Common Secondary Model from Data Set 5 150

157. P2 Components for Common Secondary Model from Data Set 3 151

158. White Box System Block Diagram for Compressor Flow Inputs 155

159. A3 F3 Flow Caused by Heater Step to 400 Watts 159

160. A0 P1 Data and Model for A3 F3 Flow Increase 159

161. P1 Models for A3 F3 Flow Increase 160

162. Ambient Temperature Change 162

163. P1 Data and Model for Ambient Temperature Change Input 162

164. P2 Data and Model for Ambient Temperature Change Input 163

LIST OF SYMBOLS

Symbol	Definition
<u>System Locations</u>	
A1 through F4	refrigerator locations in accelerator ring
RB, LB	refrigerator locations distant from ring locations
B0 through F0	compressor locations in accelerator ring
BA, EA	compressor locations adjacent to B0 and E0
SY	compressor location distant from ring locations
CHL	central helium liquifier
IN	inventory control location, physically at B0
<u>Valves</u>	
CL	inventory valve between ring and tank farm, or position of CL valve [% open]
DL	inventory valve between ring and CHL, or position of DL valve, [% open]
LS	compressor low stage loader valve, or position of LS valve, [% open]
VPi	initial valve position [% open], refers to either compressor or inventory valves
VPf	final valve position [% open], refers to either compressor or inventory valves
<u>Flows</u>	
F2	deviation flow through compressor, [g/s]
F3	main deviation flow out of refrigerator, [g/s]
F4	main deviation flow into refrigerator, [g/s]
FC	deviation flow through valve CL to tank farm, [g/s]
FD	deviation flow through valve DL to CHL, [g/s]

Symbol	Definition
Flows (continued)	
FL	power lead deviation flow out of refrigerator, [g/s]
FT	transfer line deviation flow into refrigerator, [g/s]
FP	primary deviation flow through a compressor, caused by movement of valve LS, [g/s]
FS	secondary deviation flow through compressors and inventory valves caused by changing P1, [g/s]
f{LS}	functional relation giving absolute value of compressor flow for LS position, [g/s]
Flow Parameters	
A	cross sectional area of pipe, [m^2]
D	pipe diameter, [m]
R	gas constant specific to helium, 2.077 [kJ/kg K]
T	temperature, [K]
V	header volume, [m^3]
c	speed of sound, [m/s]
k	ratio of specific heats for helium, 1.66 [-]
t	time, [s]
v	fluid velocity, [m/s]
x	axial distance along pipe, [m]
ρ	density of helium gas, [kg/m^3]
Pressures	
P1	deviation pressure of suction header, [psi]
P1a	absolute pressure in suction header, [psia]
P1o	absolute pressure in suction header at time prior to step in compressor LS position, [psia]

Symbol	Definition
<u>Pressures (continued)</u>	
P2	deviation pressure of discharge header, [psi]
P2a	absolute pressure in discharge header, [psia]
P2o	absolute pressure in discharge header at time prior to step in compressor LS position, [psia]
Pc	absolute pressure on which equation (2.6) is based, having a constant value of 15.9, [psia]
Patm	atmospheric pressure, [psia]
<u>Modeling Terms</u>	
ARX	auto-regressive with exogeneous input model form
BJ	Box-Jenkins model form
OE	output-error model form
A(q) through F(q)	polynomials in the time delay variable q^{-1}
G(q)	discrete time domain transfer function of the plant or of a forward path, a ratio of polynomials in q^{-1}
H(q)	discrete time domain transfer function of the noise, a ratio of polynomials in variable q^{-1}
R(q)	discrete time domain transfer function of a feedback path in the system, a ratio of polynomials in q^{-1}
g(k)	discrete time impulse response of the plant
h(k)	discrete time impulse response of the noise
e(t)	discrete time stochastic noise signal
u(t)	discrete time input signal
w(t)	discrete time undisturbed output signal, used with OE model
y(t)	discrete time output signal

Symbol	Definition
Modeling Terms (continued)	
$\hat{y}(t \mid t-1)$	output prediction at discrete time t given past outputs and inputs
$\hat{y}(t \mid \theta)$	output prediction at discrete time t given past inputs and outputs and model parameters
a through f	coefficients in polynomials A(q) through F(q), e.g. a11 and f11
hs	location number designation for compressor house
ko	sample number at start of valve step
kn	final sample number
ki1	index number of initial sample on which P1 data extension is based
kf1	index number of final sample on which P1 data extension is based
ki2	index number of initial sample on which P2 data extension is based
kf2	index number of final sample on which P2 data extension is based
n_k	number of delay periods of dead time
n_a	number of model parameters in A(q)
n_b	number of model parameters in B(q)
q^{-1}	time delay operator for one sample time delay
q^{-k}	time delay operator for k sample time delays
sse	sum of the squared errors
step	size of flow input step, [g/s]
$V_N(\theta)$	sample variance of prediction error

Symbol	Definition
Modeling Terms (continued)	
θ	model parameter vector
θ^T	transpose of model parameter vector
φ	regression vector of measured data
φ^T	transpose of regression vector
ω_s	sample frequency, [rad/s]
ω_n	Nyquist frequency, [rad/s]

ABSTRACT

A mathematical model is created of the dynamic response of pressures caused by flow inputs to an existing distributed helium refrigeration system. The dynamic system studied consists of the suction and discharge pressure headers and compressor portions of the refrigeration system used to cool the superconducting magnets of the Tevatron accelerator at the Fermi National Accelerator Laboratory. The modeling method involves identifying the system from data recorded during a series of controlled test, with effort made to detect locational differences in pressure response around the four mile accelerator circumference.

A review of the fluid mechanics associated with the system indicates linear time invariant models are suitable for the identification, particularly since the governing equations of one dimensional fluid flow are approximated by linear differential equations. An outline of the experimental design and the data acquisition system are given, followed by a detailed description of the modeling, which utilized the Matlab programming language and associated System Identification Toolbox.

Two representations of the system are presented. One, a black box model, provides a multi-input, multi-output description assembled from the results of single input step function testing. This description indicates definite variation in pressure response with distance from the flow input location, and also suggests subtle differences in response with the input location itself. A second system representation is proposed which details the relation

between continuous flow changes and pressure response, and provides explanation of a previously unappreciated pressure feedback internal to the system.

Data from additional tests provide characterization of typical disturbances to the system. The affects of perturbations in flow exiting the refrigerator's magnet heat load, and of changes in ambient temperature surrounding the pressure headers, are modeled for direct addition to the plant model outputs. Suggestions are presented for the use of the plant and disturbance models in the simulation and improvement of system control.

CHAPTER I

INTRODUCTION

Fermi National Accelerator Laboratory is operated by Universities Research Association, Inc. for the United States Department of Energy. Its primary function is to support high energy physics experimentation, theory, and education using an extensive physical plant and a support staff of technicians, engineers, and scientists. During the experimental work, protons and anti-protons can be delivered to experiment stations at energies of one trillion electron volts through a complex of five particle accelerators. The two particles types are either collided together within a particle detector or guided to fixed targets made up of other materials. The final accelerator in the series of five, the Tevatron, incorporates nearly 1000 niobium-titanium superconducting magnets to guide particle beams in a 4.0 mile circumferential path inside an evacuated tube, and on to their targets [Col79]*.

Cooling for these magnets comes from a large helium refrigeration system, operating at a temperature of 4.8 Kelvin [Rod83]. Since the system is distributed, helium is transferred over large distances in conduits having significant volume as well as resistance to flow. It is the dynamic response of pressures within the refrigeration system to changes in helium compressor flows and inventory flow that is addressed in this study. By using an analysis based on first principles of fluid mechanics, the influence of flow on pressure is verified, and the potential of employing linear dynamic models is realized.

* Corresponding to references in the Bibliography

Due to the complexity of the system, however, first principles are used only in a supporting role to model the refrigeration plant. As the plant and its control system physically exist, experimental methods can be used to obtain the dynamic response to a variety of inputs. The inputs include flow changes having magnitude and direction typical of past operation, as well as disturbances to the system. From the data collected, an attempt is made to identify the system using commercially available modeling software. The models produced suggest alternative schemes of controlling the plant could be investigated to improve system regulation.

1.1 The Refrigeration System

The refrigeration system for the Tevatron consists of 24 distributed satellite refrigerators supplied with high pressure gas by nine distributed compressor stations. A Central Helium Liquifier (CHL) and its transfer line supplement the supply of liquid helium produced by the refrigerators. Figure 1 shows the system schematically as it was configured during this investigation.

1.1.1 Refrigerators. The satellite refrigerators contain all the components of a Claude vapor compression refrigerator [Bar85], except for the compression stage, and are referred to simply as "refrigerators". For identification, refrigerators are grouped into six sectors, A through F, and within sectors are numbered one through four. Nearby the Tevatron ring are refrigerators LB (left bend) and RB (right bend) which support fixed target operation. During the experimentation of this study, LB and RB were not operating, but did receive a small gas purge from the Tevatron system.

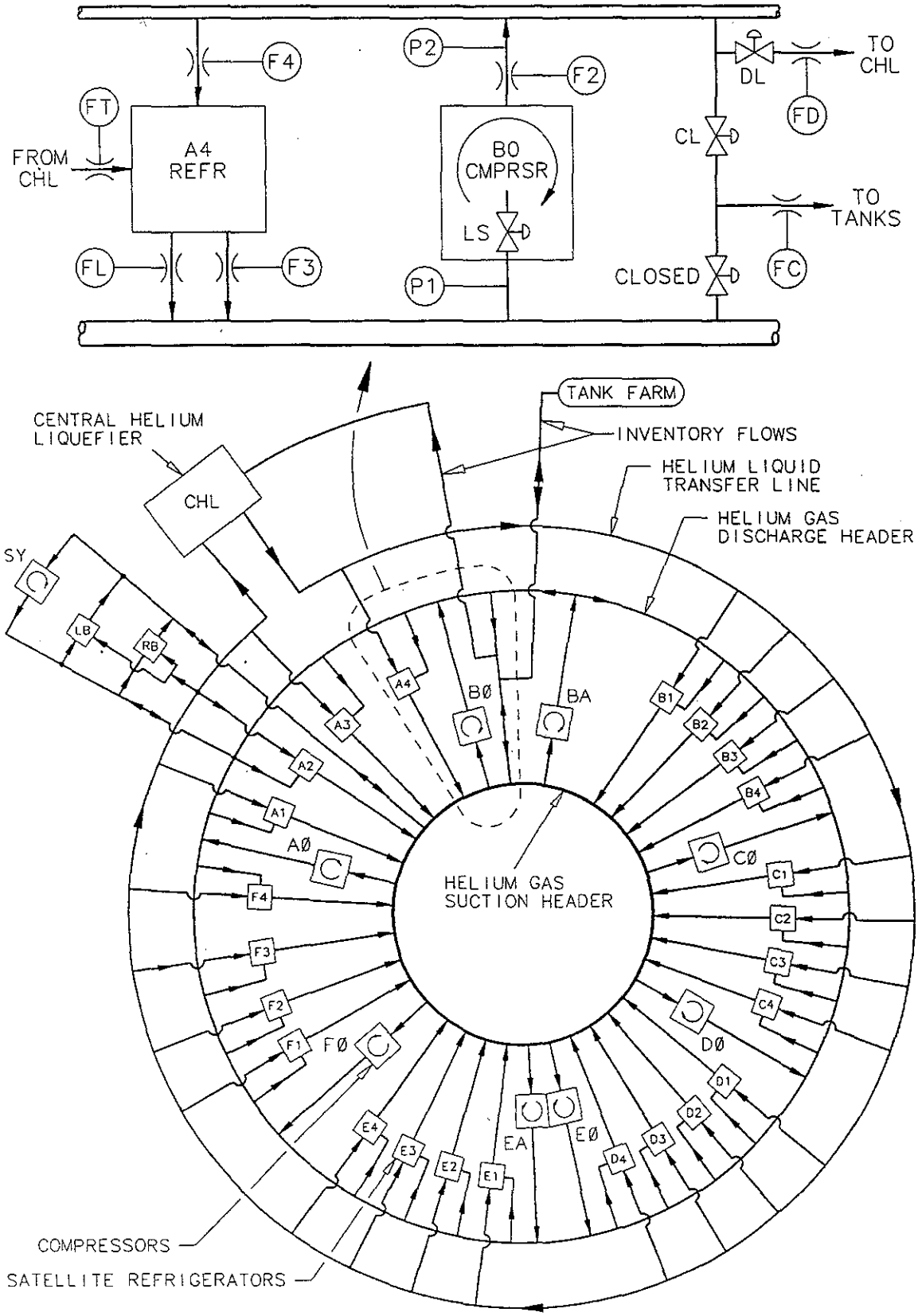

Figure 1. Tevatron Refrigeration System

With the exception of RB and LB, all refrigerators are operated identically, and all have essentially the same characteristic behavior, although helium flow rates through individual locations do vary. Each refrigerator is capable of providing 966 Watts of refrigeration at 4.8 Kelvin when supplemented by liquid helium from the CHL.

1.1.2 Compressors. Compressor stations around the ring consist of four compressors each, and are identified by the letter of the sector in which they are positioned followed by a 0 (zero) or an A. The zero and A stations in B sector and E sector are physically situated side by side. Two compressors sit in the SY (switchyard) station along with the LB and RB refrigerators. These two compressors are connected with the Tevatron system and are used for added capacity when LB and RB are not operating.

All compressors are two stage oil lubricated screw type machines which are followed by oil removal systems. Individual compressors are powered by either a 350 or a 400 horsepower motor, and flow rate through each may be varied. Control is accomplished by changing the position of the low stage inlet valve, labeled LS in Figure 1. Throughout the discussion, each compressor station will be referred to as a "compressor" for simplicity, since the flows from all four machines enter the system through the same pipe and are recorded as one flow.

1.1.3 Flow Balance. Gas piping headers run the entire circumference of the ring, connecting all refrigerators to all compressors. A three inch diameter discharge header supplies 290 psig gas from compressors to the refrigerators. Under conditions that existed during this testing, compressors (locations) are

capable of up to 220 g/s maximum flow, and individual refrigerators require nominally 45 g/s each. In addition to the high pressure gas flow, each refrigerator uses a nominal (but highly variable) 5 g/s supplemental flow of liquid helium supplied by the Central Helium Liquifier (CHL) through the transfer line located near the discharge header. The transfer line also supplies liquid nitrogen required in the refrigerator, but normally this flow has no effect on the dynamics of the helium system.

An eight inch diameter suction header collects all helium flow that exits the refrigerators (nominally 50 g/s each) and returns it to the inlet side of the compressors. Suction pressure is around 1.5 psig. This header has the same circumference as the discharge header and transfer line, although that is not apparent in the schematic of Figure 1. The suction header path is different from the other two conduits in that it resides largely underground in a tunnel housing the superconducting magnets and particle beam line. The transfer line and discharge header are placed outdoors above the tunnel.

An important feature of the system is the gas inventory exchange that takes place between the ring and the Central Helium Liquifier. The CHL receives high pressure gas back from the ring at a flow rate somewhat less than it supplied to all the refrigerators through the transfer line. This flow moves through the valve labeled DL in the dirty line, so-called for its alternative use in purifying helium gas prior to Tevatron operation. Flow also leaves the ring through the valve labeled CL in the clean line to clean storage tanks maintained at a pressure of 150 psig by the CHL facility. Flow in

the dirty line and clean line are typically around 40 g/s each. The CHL also takes back any liquid in the transfer line that is not used by the refrigerators.

1.1.4 Instrumentation. The entire refrigeration system is well instrumented, and locations for those devices most important to the investigation are shown in the enlarged portion of Figure 1. In this discussion, nominal measurement values for devices are given in parenthesis following their instrument name. The full name of a measurement will include the location name and instrument name, for example, B0 P1. All pressures are measured using commercial transducers which utilize a flexing diaphragm that forms a variable capacitor. Mass flows are calculated from various types of volume flow sensors along with pressure and temperatures associated with the location. Flow values will always be given as mass flow in g/s.

The measurements labeled P1 (1.5 psig) and P2 (290 psig) represent the pressures of the suction header and discharge headers, respectively, at compressor locations. Mass flow F2 (up to 220 g/s) is the throughput of all the compressors at a compressor location, and produces the pressure difference between suction and discharge headers. F2 is measured by an orifice plate flow meter.

Four flows stream in or out of refrigerators. F4 (45 g/s) is the stream of high pressure gas required to produce liquid helium, FT (5 g/s) is the liquid helium delivered through the transfer line, and F3 (50 g/s) and FP (up to 1.0 g/s) are the low pressure gas streams exiting the refrigerator to the suction header. F4 is measured by an orifice plate meter, F3 by a vortex shedding

meter, and FT by a venturi meter. FL, which serves to cool power leads of the superconducting magnets, is a small flow through a calibrated restriction communicating between two nearly constant pressures. When the superconducting magnets are not powered, as was the case in all testing of this investigation, the FL flow is only about 0.5 g/s. FL is read on a rotometer and not recorded electronically.

1.1.5 System Control. The instrumentation readback, recording, and system control is performed by a distributed control system (DCS) designed by the accelerator Controls Department [Nor94]. The DCS is organized into sectors, similar to the cryogenic system itself. Within each refrigerator or compressor location resides an interface crate to which instrument and control devices are wired. Through Arcnet serial communication technology, all interface crates in a sector communicate with one Multibus II crate for the sector. The Multibus II crate contains one single board Intel 80386 microprocessor based computer for each refrigerator or compressor location. A given 80386 computer handles all finite state machine and control loop calculations for its respective location. It also passes on readbacks from all instruments and control elements to a Token Ring, which in turns passes this information on to the accelerator's central ACNET (Accelerator Control Network) system.

Using the ACNET, the refrigerator system operation is easily viewed or manipulated from a console. Application programs allow for storing and plotting of readings, as well as downloading changes to finite state machines and control loops. Control loops may also be disabled so control elements can

be positioned manually from the console. A valve, such as that labeled CL in Figure 1 for instance, may be set to positions between 0 and 100 % of its travel. Recorded data may be electronically mailed to personal computers for analysis using a wide variety of numerical and visualization software.

1.2 Goals of the Investigation

The technical motivation to study pressure dynamics resulting from changes in flows is driven by the prospect of improving system performance and ease of operation. By identifying a mathematical model of the system, better qualitative and quantitative understanding of the system is achieved. Though the investigation concentrates on this identification and understanding, it also lays the groundwork for further study in the area of system control. Some of the control issues are reviewed but not addressed in detail.

1.2.1 Basic Understanding of Dynamics. Since start of Tevatron operation in 1984, no detailed studies of the dynamics of the suction and discharge gas header had been performed. This was due to more pressing problems with the system, severe lack of Tevatron operational time dedicated to such studies, and the non-existence of convenient computer software and hardware to make use of any data resulting from such studies. Neither, apparently, had an attempt had been made to operate the system with all feedback control disabled with an intent of determining how the system parameters drift.

Qualitative observations during operation indicated that there were differences in behavior dependent on location around the ring. For instance,

with equal number of individual compressors running at all compressor locations, E0 P1 tended to assume a lower pressure than B0 P1. By strict definition, steady state was never really achieved during those observations since an oscillation of pressures nearly always existed, but rather the time averaged E0 P1 was lower than P1 at other locations. Differences such as this were also suspected with regard to dynamic effects of changing compressor flows at different locations. The B0 system inventory control or the SY compressor location logically could create the locational differences since both of these features interrupt the system's physical symmetry, but these effects were speculative.

The plant in a typical process control system is often assumed to be first order in nature, however, for this refrigeration system this had not been verified and time constants had not been determined. It also was generally assumed that the suction header is somewhat undersized for its flow compared to the discharge header, and that P1 would therefore take longer to settle after a disturbance than would P2. Performing the many steps of this investigation would shed light on these issues.

1.2.2 Modeling to Quantify Dynamics. The process of developing a mathematical model for the refrigeration system is a means of quantifying the importance of suspected influences (inputs) on the system. It allows for the classification of inputs into controllable (valve positions or flows) and uncontrollable (disturbances) and identifies which are important enough to warrant further study. Arguments are presented for restricting this effort to linear time invariant modeling of the system using commercially available

software. Issues of the suitability of linear models, such as the goodness of fit for both flow increases and decreases, and the verification of time invariance can be addressed using these tools.

Depending on how well the system behaves and how many valid tests can be performed, the subtle differences due to location, an estimate of the applicable operating range of modeled parameters, and at least a qualitative assessment of the model robustness can be made. Disturbances can also be characterized with regard to amplitude and frequency during the modeling process, even if they are not the primary subject of a model detailing the effects of the controllable inputs. The disturbance characterizations can then be used to tune the control system such that the effects of the disturbances are limited.

1.2.3 Provide Basis for Improving Control. As an ultimate goal, the model developed serves as a basis for improving the control of flows within the Tevatron refrigeration system. There are a number of reasons for improving regulation of the pressures in the suction and discharge headers. These are presented here in increasing order of perceived difficulty of software implementation, with some likely requiring extensive changes to measurement, computer, and final control element hardware.

The model could be used to simulate existing and new control schemes off-line for the purposes of improving loop tuning. Currently, even to experienced operators of the system, the difference between the start of a typical disturbance that will dampen out and one that will result in serious operational problem is not immediately apparent. If the control system could

be changed to hold tighter tolerance on header pressures, then abnormal behavior could be identified sooner. As in many other process plants, a problem in this system could be identified many minutes earlier if regulation could be significantly improved.

In addition, better regulation of the suction header may improve reliability, particularly with regard to operation of cold compressors that have been added near the outlet of each refrigerator. These centrifugal machines are susceptible to tripping upon rapid fluctuations of their outlet pressure, which is P1. Simulation also may allow for reliable change in loop tuning parameters when the Tevatron configuration is changed between collider mode and fixed target mode. Disturbances at the suction header are thought to differ between the two modes since, during fixed target physics, magnet power is pulsed resulting in oscillating F3 flows.

Most of the time that the Tevatron is in operation, at least one compressor is left "idling" with a low flow, ready for the event that another machine trips off. Fermilab's costs of electricity to run an individual 400 horse power motor is on the order of $250/day [Mar96]. Due to a decreasing power factor with decreasing compressor flow, this cost does not vary much with compressor throughput. The automated turn on of additional compressors instead of simply idling machines has long been discussed. However, there has been no basis for a computed decision on where additional flow would be appropriate, other than to simply count the number of machines running at each location. A model could be used to aid in this calculation.

Currently, there is no provision for compressor locations (or refrigerator locations) to communicate information to one another. Control loops at one location know nothing of the state of another location since the loops run on a very localized level. Since compressors do have an effect on pressures all around the ring, the nine compressors locations may be good candidates for a more global multivariable control scheme. All required information would need to be brought to a central location for calculation of the control outputs, which are then sent back to local 80386 computers. In addition, information such as flows in and out of refrigerators might be used in an advanced scheme to feed forward information to the control about disturbances in the gas headers. The existing control scheme of local control might be used as a backup should the global control lose communication with local computers.

Such a flexible system might be designed with a number of weighted, perhaps competing, goals in mind. For instance, for cost optimization, compressors could be controlled such that individual machines only operate in a certain high power factor range, subject to some maximum acceptable deviation between pressures around the ring. Or, if some need emerges to keep pressures relatively constant but for which the absolute value of the pressure is not very important, some different algorithm may be needed. Thus, there are many potential uses of a model, which are limited by the robustness of the model and by the balance between an application's benefit and time expended to develop it.

CHAPTER II

DESIGN AND PERFORMING OF EXPERIMENTS

System identification can be pursued on data of unknown origin and without physical insight. However, the process is expedited by a good understanding of the physics of the system. To that end, the fluid mechanics of the refrigeration system are reviewed to assess the chances for a successful modeling outcome and to expedite modeling. With that background, variables of primary importance to the pressure and flow dynamics are selected, subject to capacity constraints on the data acquisition system. Magnitudes of input disturbances are selected based on operating experience.

2.1 Observations on the System Dynamics

The subject of dynamics in a system this size could encompasses a broad range of study from the initial cooldown of the refrigerators to disturbances at steady state. This study is, however, limited to regulatory behavior of the system at a nominal operating point. A number of observations allow for simplification in the modeling of perturbations around a set point. For one, cooldown of a refrigerator from room temperature takes days, but once the 4.8 K operating temperature is achieved thermal disturbances have only small effects on the pressures of the system. That is, thermal effects from refrigerators can be ignored to first order. In addition, within the refrigerators, pressure is well regulated by helium gas expansion engines having a fast response. Thus, refrigerators themselves contribute relatively little to the pressure dynamics of the overall system. Any

influence refrigerators do have on pressures in the remainder of the system is accounted for by flows F4, FL, and F3.

Changes in compressor loading (flow rate) occur very quickly with signals from the control system, and contribute little to lags in pressure response at distant locations in the ring. From operational experience that is consistent with theory, the slowest pressure dynamics in the system are observed with the largest volumes of the system, the suction and discharge headers. These headers form flow conduits that are rather unique in manner of operation. Both pipe volumes to some extent act as simple reservoirs, collecting and supplying gas as the flows in and out of the headers change. Both pipes also flow helium gas between compressors and refrigerators, but flow direction and rate in any given section of pipe is unknown because of the absence of measurement devices. Modeling these conduits based on first principles of fluid mechanics would be very difficult due to these uncertainties.

2.2 Suitability of a Linear Model

Attempting to apply first principles of fluid mechanics directly to the gas header dynamics might be futile, but a general review can give insight to other modeling means. This consideration leads to the conclusion that a linear time-invariant model based on experimental data may be sufficient.

2.2.1 Gas Headers as Reservoirs. In the simplest of views, the headers look like reservoirs with gas flowing in and out from many sources (compressors and refrigerators). Conservation of mass, equation (2.1), is sufficient to model this situation [Osi87].

$$\frac{V}{RT}\frac{dP}{dt} = \sum_{in} F - \sum_{out} F \tag{2.1}$$

Here, V is the header volume, T is the temperature, and R is the gas constant specific to helium, all of which are treated as constant. P represents an equal pressure throughout the header, and the F's are mass flow rates into and out of the header. The only dynamic aspect of this simplified view is that pressure is proportional to an integration of mass flow across the header boundary. Resistance to the flow is assumed to be negligible.

2.2.2 Gas Headers as a Flow System. Gas transferred in long pipelines is reliably modeled as one dimensional flow. In general the gas is compressible and may or may not have heat exchange through pipe wall with the outside environment. Friction to flow caused by the pipe walls, or other heating of the pipe, can cause changes in the gas temperature and complicate analysis by requiring that an energy balance be performed on the flow system. If heating is small and heat transfer with the environment allows constant temperature to be maintained, the system is isothermal and the governing equations are simplified. This is assumed to be the case for the uninsulated suction and discharge headers, which carry low speed flow. The gas compresses, but not to the extent that special consideration is needed in calculating friction, and elevation change effects are negligible in this system.

Under the above circumstances, the continuity equation (conservation of mass), equation (2.2), and the momentum equation (Newton's second law), equation (2.3), describe the relation of flow, pressure, time, and distance in a section of pipe.

$$\frac{1}{c^2}\frac{\partial P}{\partial t} = -\frac{\partial(\rho v)}{\partial x} \tag{2.2}$$

$$-\frac{\partial P}{\partial x} - \frac{2f\rho v^2}{D} = \frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho v^2)}{\partial x} \tag{2.3}$$

The terms of equation (2.3) represent; first, the change in pressure, P, with distance along the flow direction, x; second, the pressure drop due to pipe wall friction; third, the mass flux variation, ρv, with time, t; and fourth, the momentum flux change, ρv^2, in the flow direction, x. Within terms of both equations, ρ is fluid density, v is the fluid velocity, f is the friction factor, D is the pipe diameter, and c is the velocity of sound.

For low speed flows, the mass flux variation and momentum flux change (terms on the right hand side of equation (2.3)) represent only a few percent of the pressure drop with distance, and are often ignored. Recognizing also that ρvA is the mass flow, F, in the pipe section for which A is the cross sectional area of the pipe, the above equations can be rewritten.

$$\frac{A}{c^2}\frac{\partial P}{\partial t} = -\frac{\partial F}{\partial x} \tag{2.4}$$

$$\frac{\partial P}{\partial x} = -\frac{2f}{DA} vF \tag{2.5}$$

To numerically analyze a pipeline, a system of differential equation sets similar to (2.4) and (2.5), each set representing a segment of pipe, is solved. Time steps and pipe segment length are chosen such that solution of the system is stable, and boundary conditions such as the pressure upstream and the flow rate downstream are specified.

The insight for this investigation gained by a review of these equations is that, under some conditions, the equations are linear differential equations. The multiplication of F and v in the right hand side of (2.4) causes non-linearity, but when they are not changing significantly, the whole term can be treated as a constant. Thus, before starting the laborious collection of data and modeling, there is good reason to believe that linear modeling of pressure fluctuations in the refrigeration system gas headers will be sufficient for characterizing pressure changes with time. Efficient tools exist for producing this type of model. In addition, the variables of primary interest: pressure, flow, position or location in the ring, and time, are clearly demonstrated by review of the fluid mechanics.

2.3 Choice of Recorded Variables

With the study reduced to consider only the slowest and most pervasive pressure dynamics of the refrigeration system, a useful list of parameters to record is obtained. From the previous discussion, pressure and flows around the gas headers, as depicted in Figure 1, are certainly required. Other recorded pressures include those just inside the refrigerators, near their inlet and outlet, which serve as backup to suction and discharge pressure values. These have somewhat different magnitudes but follow the same trends, and also may help identify the source of disturbances entering the headers. Flows internal to the refrigerators serve as a secondary check of the F3 and FT flows, and flows through pressure controlled relief valves at compressor locations indicate whether any flow is bypassed directly from discharge header to suction header.

Operating experience indicates that it also is important to observe not only the F2 flows, but also the changes in the LS valve positions that can cause F2 to change. Due to wear in the hydraulic positioning mechanism of these valves, they are known to move on their own. If they are under loop position control, the LS valves will be repositioned after the next computer sample time. Still, the resulting flow change can be significant and might explain some disturbances. Compressor motor horsepower, which changes with the F2 flow, is also recorded as a means of deciphering unintentional flow change, and also indicates an individual compressor is running.

To help verify the assumptions that refrigerators add little to the system's pressure dynamics, 25 pressure, temperature, and flow parameters from one individual refrigerator location are monitored. Also, pressures in the helium inventory lines, and backup measurement of inventory line flows located within the CHL, are collected for modeling changes caused by movement in valve B0 CL. A few miscellaneous devices are recorded, including; atmospheric pressure, ambient temperature, magnet current, and values associated with the CHL liquid helium supply. In all, 360 instruments are recorded, which is the maximum that will fit in the data acquisition system dedicated to this purpose.

2.4 Choice of Experimental Parameters

In addition to the choice of instruments to record, the profile and magnitude of inputs, location of input signal, sample time, and duration of the testing must be determined. Severe constraints on the available study time window require that most of these decisions be made before the testing

is performed, and be based on past operating experience. Also, the large number of recorded parameters means that during the testing, only a handful of the parameters can be monitored by the operator. The combination of insufficient time to develop a model between tests and of not being able to monitor recorded parameters in real time, makes prior selection of testing parameters an important task.

2.4.1 Test Inputs. Ljung [Lju87] presents arguments for designing the inputs to system identification experiments. Considering single input single output systems without feedback present, the applicable results of these arguments can be summarized with the following. Theoretically, any n different sinusoidal inputs, or an input containing any n sinusoids is enough to identify a system whose true representation is an nth order differential equation. Identification is improved if the input frequency spectrum is contained in the system's frequency range of interest, and if an approximate model is known beforehand, an optimal input can be developed. Both this range and an approximate model, however, often are not known prior to the experiment.

If feedback is present during the experiment, for instance because the system is unstable without active control or specific experimental time is not available, the above summary also holds if certain conditions hold with respect to the feedback controller algorithm and the input signal. The presence of feedback also tends to reduce the sensitivity of the output to the input, making it more difficult to separate the effects of noise and disturbances on the system.

2.4.2 Compressor Valves. Positioning of the compressor low stage loader valves (LS valves) is the primary means of purposely affecting both P1 and P2. During operation, these valves are generally allowed to move between 80% and 100% under closed loop control regulating P2. Loop calculated changes in valve position are thus something less than 20%, with poor, but stable, control resulting. Flow through the valves is a nonlinear function of the valve position, and since it is flow that relates to pressure through pipeline flow theory, a correlation must be made between valve position and flow. It is assumed that, at least for a short duration, the refrigeration system can be run in an open loop mode, but that if required, the additional conditions required to produced a model from closed loop operation can be satisfied.

As previously mentioned, reliable control of the LS valves is not always achieved. Due to the difficulty of producing even relatively simple accurate step functions with these valves, no attempt is made to perform a detailed analysis of the optimal excitation input. Step functions can be achieved reasonably well on selected compressors, though, and also satisfy the spectral criteria for an input. That is, a step function contains an infinite number of sinusoids sufficient to excite mechanical systems typically having low order dynamics, and steps signals are what have always been used to control the system in the past. Position drift and overshoot are reduced by running the valves between 100% and some other value so that, at least at one end of the travel, a solid mechanical stop is encountered. Specific step sizes of 2%, 4%, and 8% in both the directions, hopefully would provide an

indication of any differences between the effects of flow increases and flow decreases.

A correlation between compressor loader valve position and flow rate is obtained using data obtained by Martinez [Mar96] for three different compressors. The data is reproduced in Table 1, and shows variation between the data sets, due in part to differences is inlet pressure, P1, during the data collection. Placing these mass flows on an equal inlet pressure basis and averaging the three sets provides for the normalized final column of Table 1, and for the correlation

$$f\{LS\} = 0.001593(LS)^3 - 0.4947(LS)^2 + 34.59(LS) - 959.0 \text{ for } 80\% < LS \leq 98\%$$

$$f\{LS\} = 5.372(LS) - 482.4 \text{ for } 98\% < LS \leq 100\%. \qquad (2.6)$$

This correlation is plotted in Figure 2, along with the normalized average data on which it is based. Note that the correlation applies to an inlet pressure of 15.9 psia, but that multiplying the result by the actual absolute value of inlet pressure and dividing by the 15.9 value allows use of the equation under any inlet pressure condition.

2.4.3 Inventory Valve. The near symmetry of the circular gas headers suggests that the response to changes in compressor flow at different locations will have similar shape, but that the inventory control flow may cause some magnitude variation. Operationally, B0 is the most important location because of the presence of the CL and DL valves, and therefore is a priority for completion of compressor valve response testing. To directly gauge locational differences, the compressor valve changes outlined in Section 2.4.2 are also

Table 1. Three Data Sets of Compressor Flow vs. Loader Valve Position

Valve Position (%)	Flow Rate (g/s) at Test Inlet Pressure			Normalized Average Flow (g/s) at 15.9 psia
	15.6 psia	15.8 psia	15.9 psia	
100	54.0	57.1	54.6	54.77
99	48.6	48.4	52.6	49.46
98	44.3	43.3	45.6	44.03
97	44.3	42.1	43.4	42.90
96	42.5	41.8	42.4	41.88
95	41.8	41.5	40.9	41.05
94	39.9	40.9	39.8	39.86
93	40.6	39.9	39.2	39.56
92	38.6	39.6	38.9	38.71
91	38.1	39.2	38.8	38.41
90	38.1	39.0	38.1	38.08
89	37.5	38.6	37.2	37.45
88	37.4	38.2	37.2	37.28
87	36.7	37.5	36.5	36.59
86	36.5	37.6	36.1	36.42
85	36.0	36.9	35.5	35.83
84	35.6	36.6	35.5	35.60
83	35.0	36.6	34.9	35.20
82	34.8	36.6	34.4	34.97
81	34.5	36.1	34.2	34.64
80	34.3	35.9	33.6	34.31

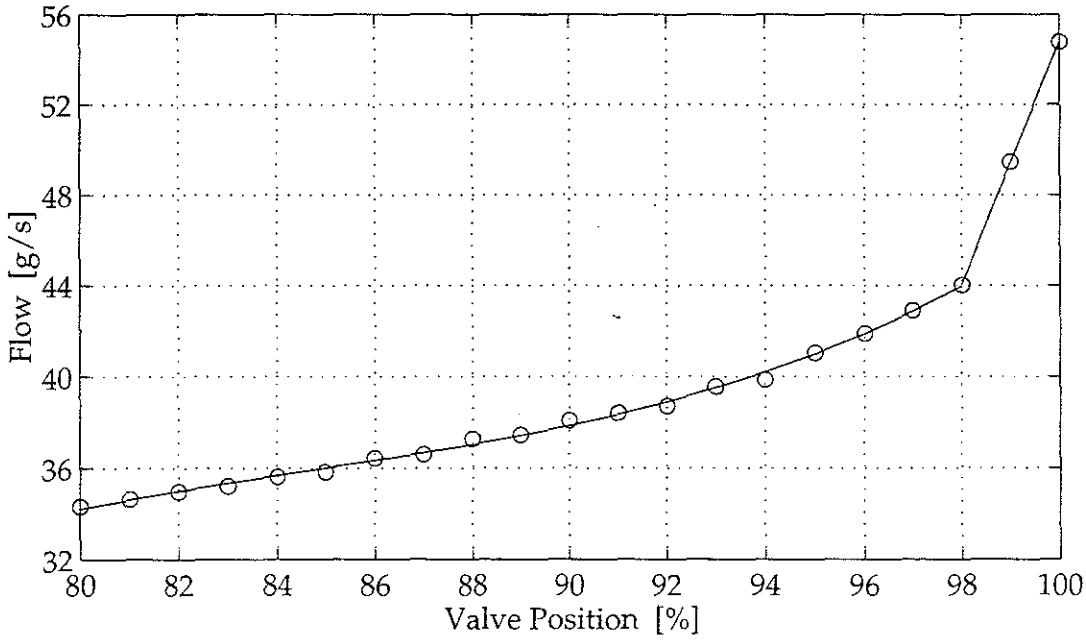

Figure 2. Compressor Flow vs. Loader Valve Position

performed at D0 and F0. Fewer tests, typically only one having increasing and one having decreasing flow, are run at C0, E0, and A0, as time permits. It is assumed that, because the inventory valves are essentially co-located, flow through either B0 CL and B0 DL would have the same effect on the header system pressures. Experience with the inventory controls provides less guidance in choosing these test inputs, and changes on the order of 5% around a nominal 50% position are a starting point. This flow and that of all compressors are viewed as the controllable inputs under normal operating conditions.

2.5 Choice of Sample Time and Test Duration

2.5.1 Sample Time. Fluid mechanics, design of dynamic experiments, and features of the data acquisition system all play a role in selecting the data sample time. From the equation for the speed of sound, c, in an ideal gas

$$c = \sqrt{kRT} \tag{2.7}$$

where k is the ratio of specific heats and R is the gas constant (both of helium), the time for a pressure disturbance to propagate is found as a function of temperature, T. For k=1.66, R=2.077 J/kgK, and T=273 degrees K, c is 970 m/s. For a disturbance to reach the furthest point, that is half way around the ring 2.0 miles (3200 m) away, takes about 3.3 s. Compared to the time constants of the system this is an insignificant amount of time, but it does give a lower limit to sample time for a model that does not need to include transport delays.

Sampling theory [Zie93] indicates that in order to capture the effect of inputs having frequency components of ω_n and lower, the sample frequency,

ω_s, must be at least $2\omega_n$. In the limiting case of $\omega_s=2\omega_n$, should the system respond to input frequencies higher than ω_n, that response would be attributed to lower frequency inputs. As a rule of thumb ω_n, known as the Nyquist frequency, should be defined somewhere above the frequency at which the system is well damped, and the sampling time should be several times faster than that. Sampling faster, while not as detrimental as sampling slower, has some problems of its own. In general, a faster sample frequency carried through to the modeling stage results in poles clustered around the point 1 on the unit circle. This can result in computational problems during numerically sensitive procedures. For this investigation, sampling faster also decreases the total amount of study time since there is a fixed buffer space in which to store data.

Since prior to analyzing the data the Nyquist frequency is not well defined, the goal becomes to sample as fast as possible given the other constraints. The data acquisition system is capable of holding 60 million data points for up to 360 different instruments. Having previously identified 360 relevant instruments, and performing the required arithmetic, this can equate to sampling each device every five seconds for 9.6 days. After that, either the data acquisition can be halted, or the first data becomes overwritten. The five second sample time is near the minimum to avoid including dead time in the model, and given the discussion to come in Section 2.6, the allotment of over nine days is sufficient. After the testing is complete, the data may be re-sampled at a slower rate if required to ease computational problems.

2.5.2 Test Duration. Design calculations for the gas headers and the presence of severe flow restriction in the suction header, have led operators of the system to believe that dynamics in the suction header are much slower than those is the discharge header. Besides the known localized restrictions, pipe friction terms for the two header also suggest the suction header will come to equilibrium more slowly. Operating experience indicates this equilibrium time for a change in compressor flow is on the order of 20 minutes. To provide margin for error, initially tests should continue for up to 50 minutes, with some reduction in duration possible later once confidence is established, and in order to make room for additional tests.

2.6 Performing the Experiments

The first extended site closure in the history of the Fermilab coincided with all of the testing reported here. During these rare conditions, a skeleton staff oversaw the idle operation of the entire facility, and no physics experiments were planned. Though it is usually warmed up to save operating costs for shutdown periods of greater than a few weeks, the Tevatron accelerator remained cold for this ten day period. Thus all planned testing was completed in the fashion described previously, without affecting laboratory business. An effort was made to adjust flows so that suction pressures around the ring were equalized, in preference to achieving equal discharge pressure. Sometimes this meant operating a few tenths of a psi away from the usual suction pressure value of about 1.2 psig. Resulting variations in discharge pressure were small, and since the system was assumed to have only small

non-linearity, operating slightly off of the nominal conditions would have little adverse effect on the results.

Pressure response tests utilized three primary input types: compressor flow changes, inventory flow changes, and disturbances. Each planned compressor flow input test represents the step change of a control valve (with direction and magnitude as previously indicated in Section 2.4.2) at a single compressor location. Fortuitous events, such as a compressor turning off during a long stretch between tests and the turning on of a compressor to prepare for the next day's testing, added to the data obtained. One of the data sets resulting from these tests is used in Section 3.3 to describe the modeling process for compressor flow inputs. The results of model identification for eighteen such tests are presented in Section 4.1, and an averaged model validation using four additional tests is discussed in Section 4.3.3. A method of directly relating continuously measured compressor flow (rather than initial flow step) to system pressures is given in Section 4.4.

Each inventory flow input test represents a step change in the valve B0 CL, and data from one of these tests is used in Section 3.4 to detail the pressure response modeling performed for this input. Section 4.2 presents the results of model identification for four inventory flow input tests, and Section 4.3.4 discusses an averaged inventory flow input model validation using two additional tests.

Noteworthy is the fact that the system behaved stably, even predictably, when compressor and inventory loop control was disabled, and that it was not necessary to resort to closed loop testing. Under these circumstances,

disturbances could be made and/or observed within the system. Flow rate from a refrigerator, F3, is an uncontrolled input under normal operating condition. It was believed to cause a sinusoidal influence on suction pressure, proposed to result from regulation internal to the refrigerator or from magnet powering. It is possible, however, to purposely vary this flow in several ways. One method which does not significantly affect the flow into the refrigerator, F4, is to use a heater to boil off liquid helium stored in a dewar that is part the refrigerator circuit. This mimics an increase in the cooling load a refrigerator must deal with, which might occur, for instance, when there is a problem with a thermal insulation vacuum. One such experiment is analyzed in Section 4.5.

During some of the testing, monatomic drifts in suction pressure were observed which could not be accounted for by changes in compressor flow. Therefore for one period of over 24 hours, the compressor and inventory system was allowed to operate unregulated in order to gauge this drift. An initial qualitative correlation of suction header pressure to ambient temperature was thus recognized before data analysis began. A model for this disturbance is presented in Section 4.5.

CHAPTER III

PROCEDURES FOR DATA ANALYSIS AND SYSTEM MODELING

Based on the nature of the process and the intended use of the model, a non-linear analysis does not initially appear required. Non-parametric methods that identify the frequency response of the system, to which parameters can later be fit, are most successful only when used in conjunction with sinusoidal input testing. There is reason, then, to concentrate on developing linear time invariant parametric models, for which many software tools are available. If, later, the system appears more complicated, these tools may still be employed to achieve an approximate model or be combined with other techniques.

The dynamics of the gas header system are analyzed on two levels: one as a black box with little need for an understanding of the plant required, and a second using the considerable physical insight gained from the black box model results. Both types of models are developed on a Macintosh Quadra 800 using the System Identification Toolbox [Lju95] authored by Lennart Ljung and published by The MathWorks, Inc. This toolbox is built upon the instruction set of the matrix based numerical computation software MATLAB, which also supports widely used control system design tools. The Toolbox, as it will be referred to, allows for a wide variety of model forms including state-space, multiple input-single output, and single input-single output. It is up to the user through educated guess or experience to choose the most appropriate form.

3.1 Linear Time Invariant Parametric Models

Linear models can be described in either the frequency or time domain. For discrete time signals and plant models, the z-transform is used in the frequency domain description, and discrete convolution is used in the time domain description. Unlike the case of transforms and convolution in continuous systems, easy conversion is made between the z-transform and discrete convolution notations for discrete systems. The time domain notation is primarily used here, with the operator q^{-1} representing a backward shift or delay in time. Thus $q^{-1}u(t)=u(t-1)$, indicates the value of signal u at time t-1, the immediately previous discrete sample time. The time domain transfer operator of the plant, G(q), is a function of the system's impulse response, g(k), and the delay operator.

$$G(q) = \sum_{k=1}^{\infty} g(k)q^{-k} \tag{3.1}$$

Note that the index k starts with one instead of the more standard zero value. Ljung has formulated the notation in this way to emphasis the fact that predictions of outputs are the subject of modeling, and that no controllable input at the time of the prediction has an influence on that prediction.

3.1.1 A Complete Model Description. Generally a system is thought to include the deterministic model of the plant, G(q), as well as a noise or disturbance transfer operator, H(q). A single input-single output (SISO) system is depicted in block form in Figure 3 using this notation. The noise model accounts for signals that can not be explained by the plant model, and is thought of as modifying a (typically) Gaussian noise input e(t). For plant

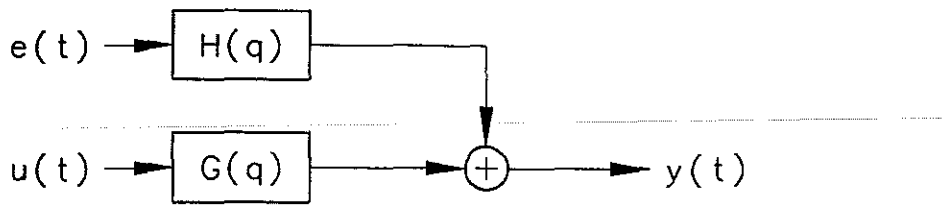

Figure 3. SISO Model Block Diagram

output y(t), plant input u(t), and stochastic noise e(t), such a system model is written

$$y(t) = G(q)u(t) + H(q)e(t). \tag{3.2}$$

In the frequency domain the equation would appear as

$$Y(z) = G(z)U(z) + H(z)E(z). \tag{3.3}$$

Since noise can influence the current output prediction, Ljung chose to denote the noise transfer operator starting with a k index of zero and to define h(k)=1 for k=0. The form of H(q) then becomes

$$H(q) = 1 + \sum_{k=1}^{\infty} h(k)q^{-k} \tag{3.4}$$

This provides for an additive noise term at the time of output prediction that is equal to the noise input e(t).

An example, based on discussion by Ljung*, helps to illustrate the meaning of the notation and to show why it is useful to include a noise model. Consider the prediction of the next output based on past inputs and outputs for the plant. This prediction is denoted $\hat{y}(t \mid t-1)$, and let the plant model be of the form

* see [Lju87] pp. 54-60

$$G(q) = \frac{bq^{-1}}{1 - aq^{-1}} = \sum_{k=1}^{\infty} a^{k-1} q^{-k}. \tag{3.5}$$

From (3.5), two expressions can be given for the output prediction assuming no noise term is present. These are

$$\hat{y}(t \mid t-1) = ay(t-1) + bu(t-1) \tag{3.6}$$

and

$$\hat{y}(t \mid t-1) = b\sum_{k=1}^{\infty} a^{k-1} u(t-k). \tag{3.7}$$

These two equalities emphasize different measurements used to make the prediction. The first is most useful when the immediate past measurement of y is highly accurate, and the second is better when the measurement is poor but a significant number of previous inputs is known. Now, if a noise model of the form

$$H(q) = 1 + cq^{-1} \tag{3.8}$$

is added, the predictor equation is uniquely given by

$$\begin{aligned} \hat{y}(t \mid t-1) &= \frac{c}{1+cq^{-1}} y(t-1) + \frac{bq^{-1}}{1+(c-a)q^{-1} - acq^{-2}} u(t) \\ &= (a-c)y(t-1) + (ac)y(t-2) + (b)u(t) + (bc)u(t-1). \end{aligned} \tag{3.9}$$

In simplifying (3.9), past predictions such as $\hat{y}(t-1 \mid t-2)$ are replaced by quantities known at time t, such as y(t-1). Including a noise model adds parameters that allow a weighting to be given to past output measurements and input values. Given an appropriate noise model, values for the plant model may be more accurately determined from experimental data.

3.1.2 Model Forms. For linear models, the G(q) and H(q) of Figure 3 are both ratios of polynomials in q^{-1}, which may or may not have terms in common. Most generally, an equation for the system takes the form

$$y(t) = \frac{B(q)}{A(q)F(q)} u(t - n_k) + \frac{C(q)}{A(q)D(q)} e(t) \tag{3.10}$$

where n_k is the number of delay periods before an input has an effect at the output, and where B(q) is given by

$$B(q) = b1 + b2q^{-1} + b3q^{-2} + \ldots + bn_b q^{-n_b+1}. \tag{3.11}$$

All other polynomials are given similarly to that for A(q) as in

$$A(q) = 1 + a1q^{-1} + a2q^{-2} + \ldots + an_a q^{-n_a}. \tag{3.12}$$

Depending on the model complexity, any of the polynomials may be missing, and in a search for a black box model, any combination of terms may be used. The A(q) term is unique in that it describes the result of random inputs that enter the system and are affected by the dynamics of the plant model, that is random plant input noise. C(q) and D(q) polynomials describe noise and disturbances occurring at the plant output. Multiple inputs may be present, in which case additional terms of ratio of polynomials are added to (3.10).

The reduction of (3.10) to include only A(q) and B(q) polynomials is referred to as the Auto-Regressive with exogeneous input (ARX) model form. Expanded, this model appears as

$$y(t) + a1y(t-1) + \ldots + an_a y(t-n_a) = b1u(t-1) + \ldots + bn_b u(t-n_b) + e(t). \tag{3.13}$$

Defining the measured data vector, φ, and model parameter vector, θ, as

$$\varphi(t) = [-y(t-1)\ \ -y(t-2) \ldots -y(t-n_a)\ \ u(t-1)\ \ u(t-2) \ldots u(t-n_b)]^T, \tag{3.14}$$

$$\theta = [a1\ \ a2 \ldots an_a\ \ b1\ \ b2 \ldots bn_b]^T, \tag{3.15}$$

the predicted output values at time t given past data φ and model parameters θ can be written as the vector

$$\hat{y}(t \mid \theta) = \theta^T \varphi(t) = \varphi^T(t)\,\theta. \tag{3.16}$$

Minimizing the difference between the predicted and measured values of the output at time t by selecting appropriate model parameter becomes a numerical task performed by computer. In the case of an ARX model with the prediction error characterized by the sample variance

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}[y(t) - \varphi^T(t)\,\theta]^2 \tag{3.17}$$

the calculation is a straight forward linear regression using the least squares criterion. The analytical solution for the relatively simple ARX case is

$$\hat{\theta}_N = \left[\frac{1}{N}\sum_{t=1}^{N}\varphi(t)\,\varphi^T(t)\right]^{-1}\left[\frac{1}{N}\sum_{t=1}^{N}\varphi(t)\,y(t)\right]. \tag{3.18}$$

Methods to determine model parameters for other common variations of (3.10) are based on the form of a linear regression, but require an iterative solution. One variation, the Output Error (OE) form, uses only polynomials B(q) and F(q) and when expanded appears as two equations written in terms of the undisturbed output w(t).

$$w(t) + f1\,w(t-1) + \ldots + fn_f\,w(t-n_f) = b1\,u(t-1) + \ldots + bn_b\,u(t-n_b) \tag{3.19}$$

$$y(t) = w(t) + e(t) \tag{3.20}$$

In this case the regression vector, φ, contains data as well as predictions of the undisturbed output and therefore depends on the model parameters, θ.

$$\varphi(t,\theta) = [u(t-1)\ldots u(t-n_b)\ \ -w(t-1,\theta)\ldots -w(t-n_f,\theta)]^T \tag{3.21}$$

$$\theta = [b1\ \ b2\ldots bn_b\ \ f1\ \ f2\ldots fn_f]^T \tag{3.22}$$

The iterative nature of a solution for θ comes from the need to make assumptions for past predictions of the output, prior to obtaining values for the model parameters that are used to make those predictions. These methods are referred to as pseudolinear regression.

3.2 Black Box Modeling

3.2.1 Modeling Method. A mathematical model relating each input to each output can be developed one by one, or the system model may be determined in state-space format. The state-space approach is best applied when there are well understood relationships between inputs and outputs, and when intermediate variables have physical meaning. Such is the case in many motion control systems for which the first principles relation between acceleration, speed and position is known, but where a need exist to find coefficients that complete the known model form. If state space analysis were applied to the refrigeration system considered here, it might be viewed as an input (a compressor flow at B0) affecting an intermediate variable (pressure locally at B0) which in turn effects a final output (pressure at adjacent C0). In a similar manner all other cause-effect combinations in the system would be identified, and then coefficients for the resulting model form could be determined.

For the purpose of improving control of suction and discharge header pressures using the present control scheme, it is enough to know the effect of a controllable input upon an output, for which developing many individual SISO relations is sufficient. The effects of any identified disturbances may be treated in the same manner. With this view, the refrigerator system is seen as

a black box having disturbance and manipulable valve inputs, and pressure outputs. If pressure at one location effects an adjacent pressure, this relation does not need to be identified, or even acknowledged, while the SISO is being developed.

3.2.2 Data Preparation. The numerical techniques included in the Toolbox are most efficiently used if a stream of data is first normalized by subtracting the mean value and dividing by the standard deviation of the values. Subtracting the mean allows the software to concentrate on changes in the variables, without having to adjust to some, perhaps large, offset value. This is most useful when the input is a series of steps or a sine function. Dividing by the standard deviation aids the computations by allowing all manipulation to be done on numbers of the same magnitude.

For this investigation, because the inputs are single step functions, the initial values are subtracted instead of subtracting means. The resulting deviation variable has the desired effect of reducing a large offset between zero and the absolute value of the variable. For instance, P2 nominally has a value of 290 psig, but perturbations are on the order of a few psig. The data is not, however, divided by the standard deviation because changes in all variables are close in numerical magnitude for the physical units chosen. Not dividing by the standard deviation greatly simplifies understanding of modeling results since the coefficients automatically have the proper units and may be applied directly to other data sets for model verification.

Data is visually inspected with plotting routines of the ACNET control system, and evaluated for clarity of trends and absence of overwhelming

disturbances. Since down loading data from the control system to the Macintosh and subsequent manipulation into a usable matrix format is time consuming, data for a single step input test was chosen for model development. That particular data is from one of the longer trials, and has a large compressor valve step at the important B0 location. After progress was made with this data, other data were prepared and modeled in the same manner.

The data from all pressure response testing outlined in Section 2.6 were arranged into numbered data sets. The first 22 sets are given in order of location of the input, starting at A0 and ending with the inventory valve location. Sets 1 through 18 constitute those used to obtain models for compressor flow inputs, and sets 19 through 22 are those from which inventory flow input models are built. Sets 23 through 26 are used to validate an averaged compressor flow input model, and sets 26 and 27 validate the averaged inventory flow model. Though inventory valves are located near the B0 compressors, they are designated to have location IN to distinguish these tests from compressor loader valve tests. This ordering of data sets allows for the most logical comparison of results at a given input location. It has no relation to the order in which the tests were performed or in which the data was analyzed. Data set 29 contains the experimental results for the F3 flow disturbance, and set 30 holds the results of the ambient temperature influence test.

3.3 Black Box Model for Compressor Flow Inputs

In this section the data of set 3 in analyzed to form a model relating compressor flow change, F2, to both suction pressure, P1, and discharge pressure, P2. This provides an example of the procedure generally used to process other similar data sets. Later, in Chapter IV, the SISO models produced in this manner are collected with those for inventory flow input to assemble an overall system model.

3.3.1 Characteristics of Data. The data of set 3, prepared as described above, is shown in Figures 4 though 23. There are a number of striking aspects to the plots that warrant consideration prior to the description of the modeling. The increase in flow B0 F2, resulting from a compressor valve movement from 92.5% to 100%, is given by the upper plot in Figure 4. Note that correlation (2.6), corrected for inlet pressure, predicts a flow step change of 16.0 g/s that is very close to the value plotted. There is some noise superimposed on the signal and a very slight decrease in the magnitude after the step.

Also shown is the sum of flows other than B0 F2, and this will be called the secondary flow, FS. The FS is defined as the sum of the change in all other flows entering the discharge header, and it represents deviations in all the compressor flow except the stepped flow (at B0 in this case), plus the deviation in inventory flows. The decreases in F2 at the non B0 locations are such that alone they could easily be missed due to noise in the signals, but collectively summed in FS they are on the order of the step increase in B0 F2. Figure 5 shows the sum of all F2 and inventory flows as much different from

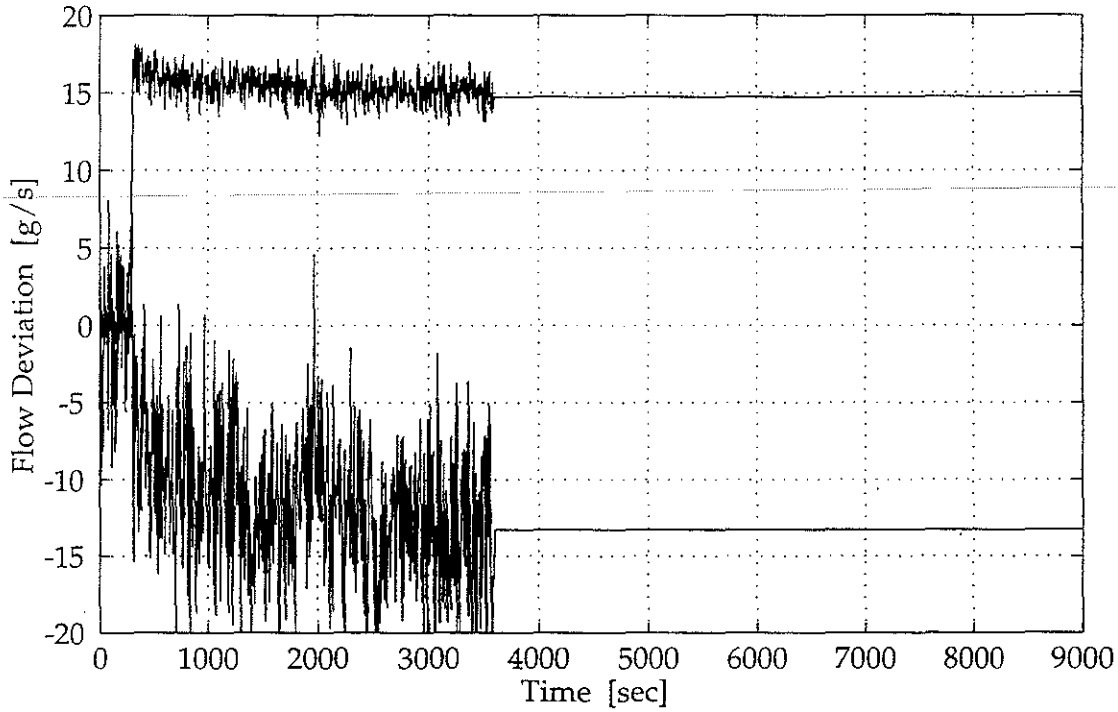

Figure 4. B0 F2 Flow and FS Flow from Data Set 3

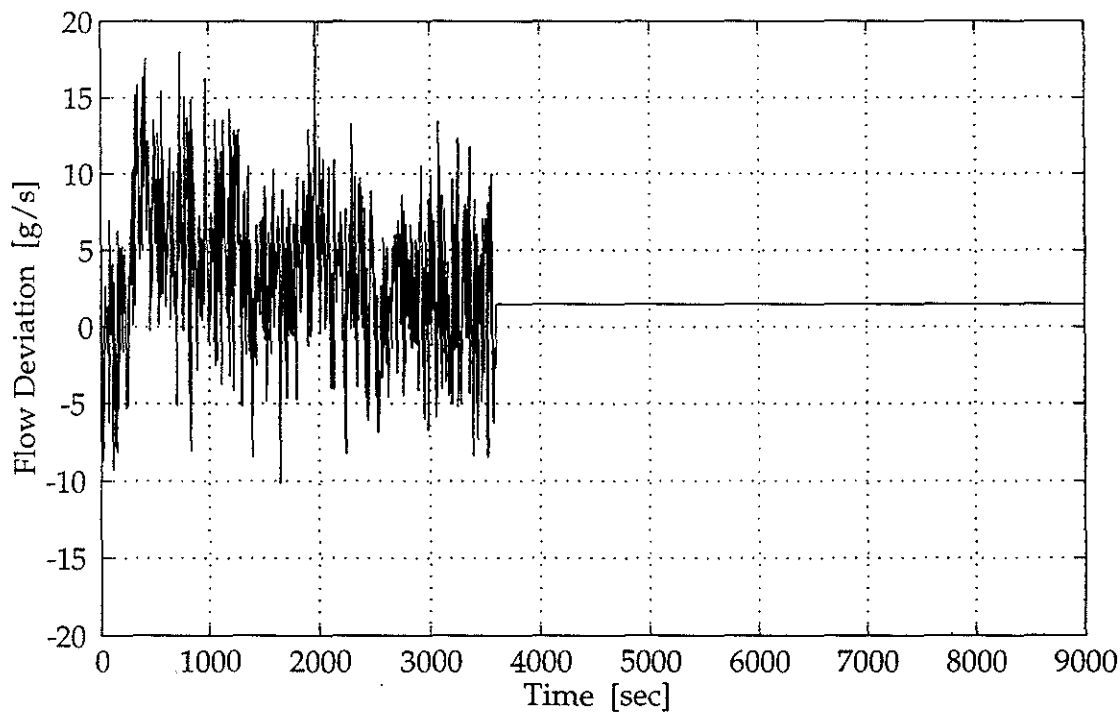

Figure 5. Sum of B0 F2 Flow and FS Flow from Data Set 3

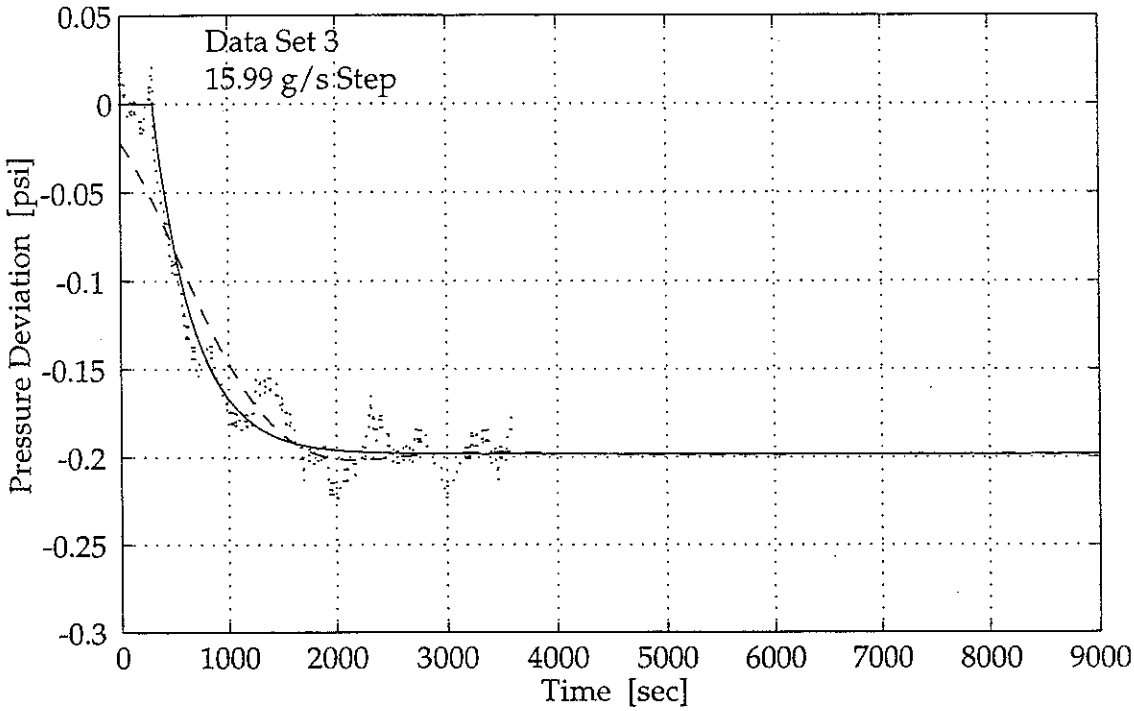

Figure 6. A0 P1 Data, Filtered Data, and Model for Step in B0 F2

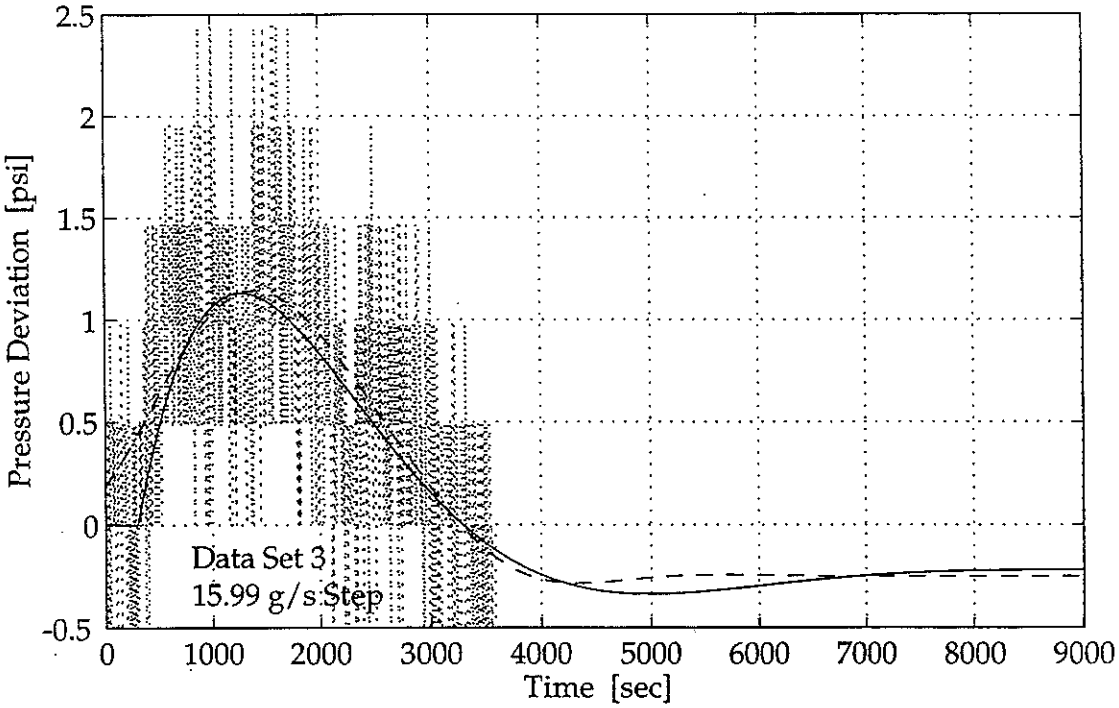

Figure 7. A0 P2 Data, Filtered Data, and Model for Step in B0 F2

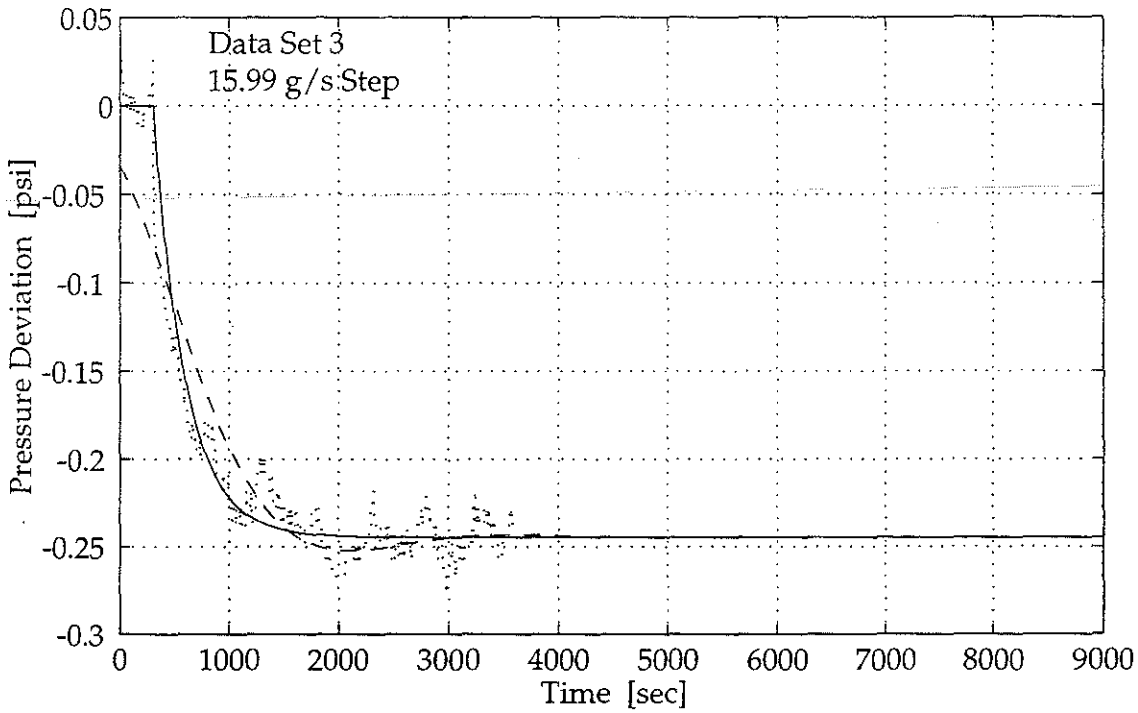

Figure 8. B0 P1 Data, Filtered Data, and Model for Step in B0 F2

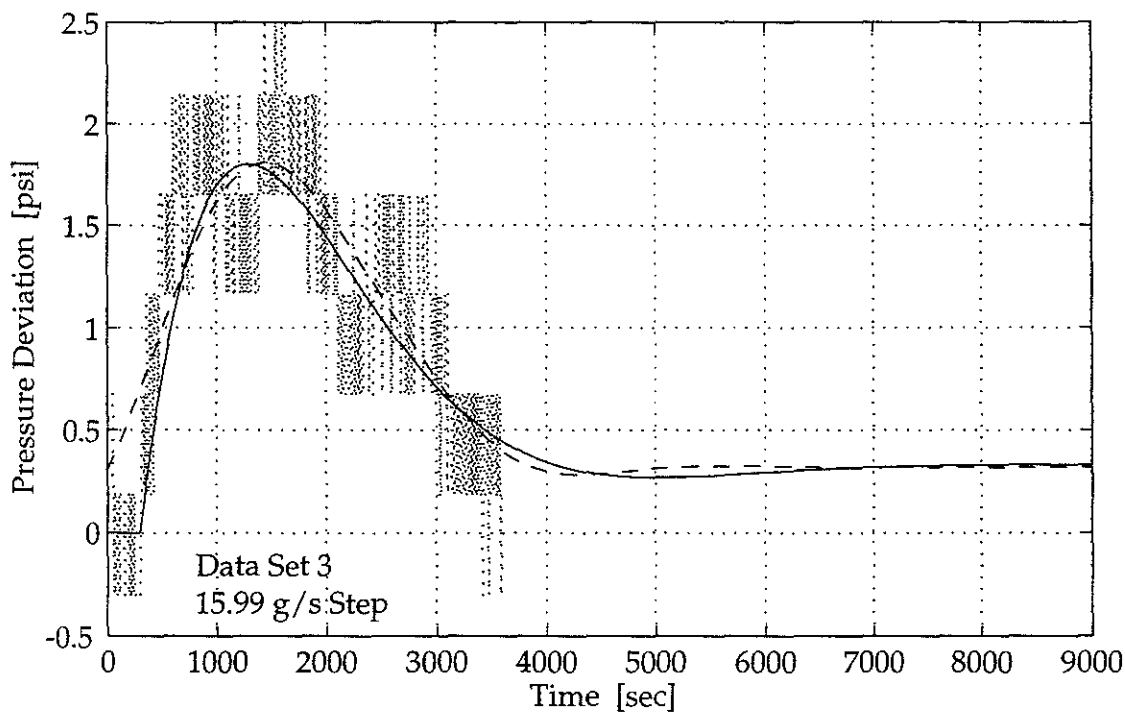

Figure 9. B0 P2 Data, Filtered Data, and Model for Step in B0 F2

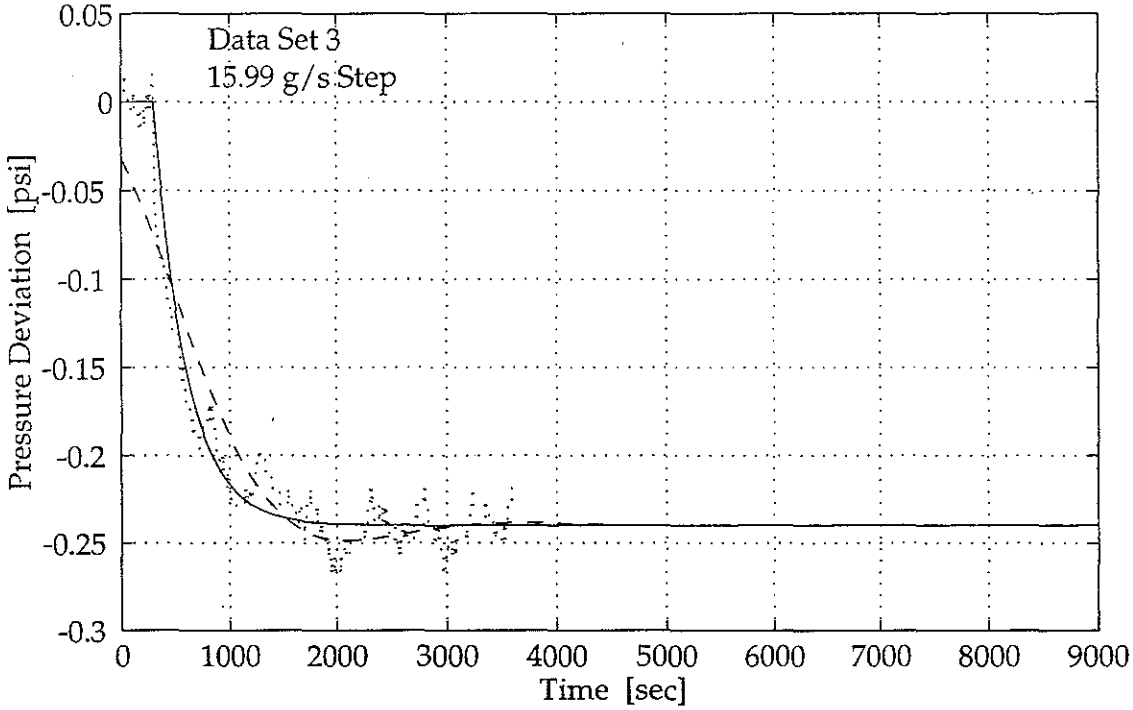

Figure 10. BA P1 Data, Filtered Data, and Model for Step in B0 F2

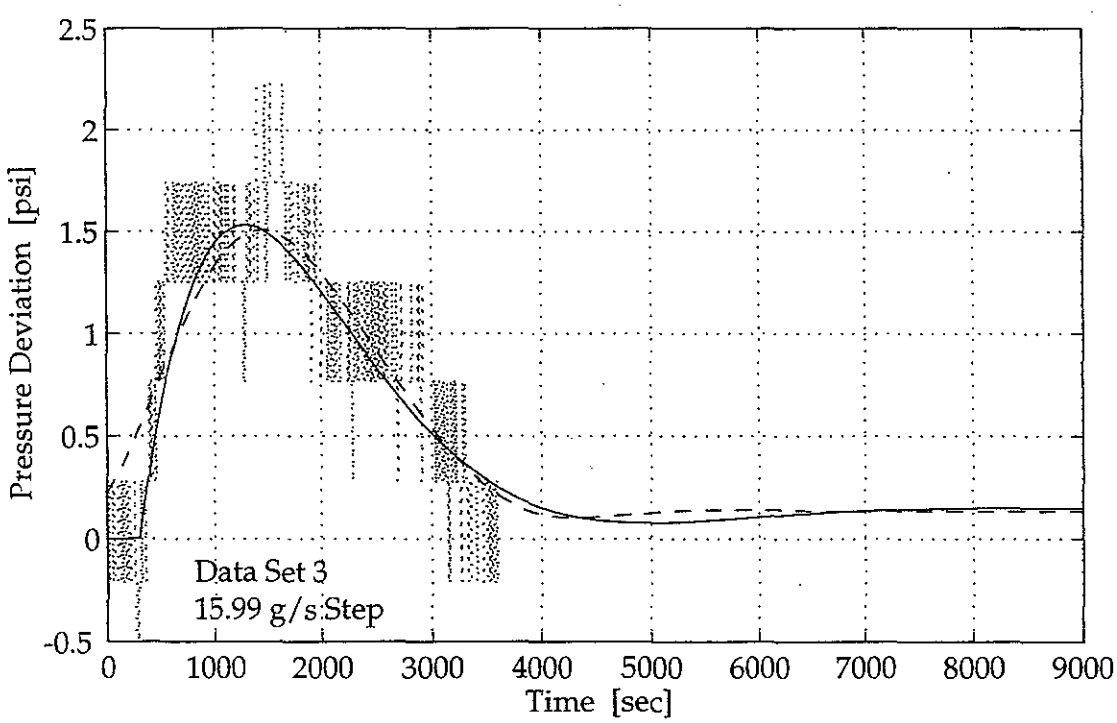

Figure 11. BA P2 Data, Filtered Data, and Model for Step in B0 F2

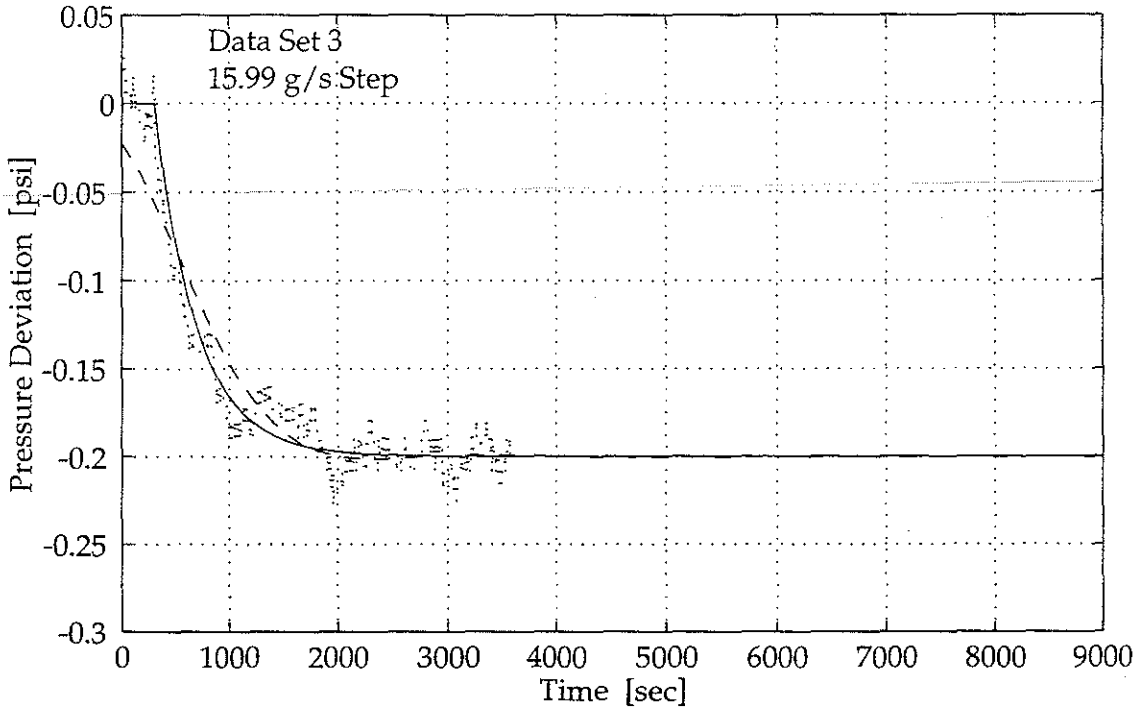

Figure 12. C0 P1 Data, Filtered Data, and Model for Step in B0 F2

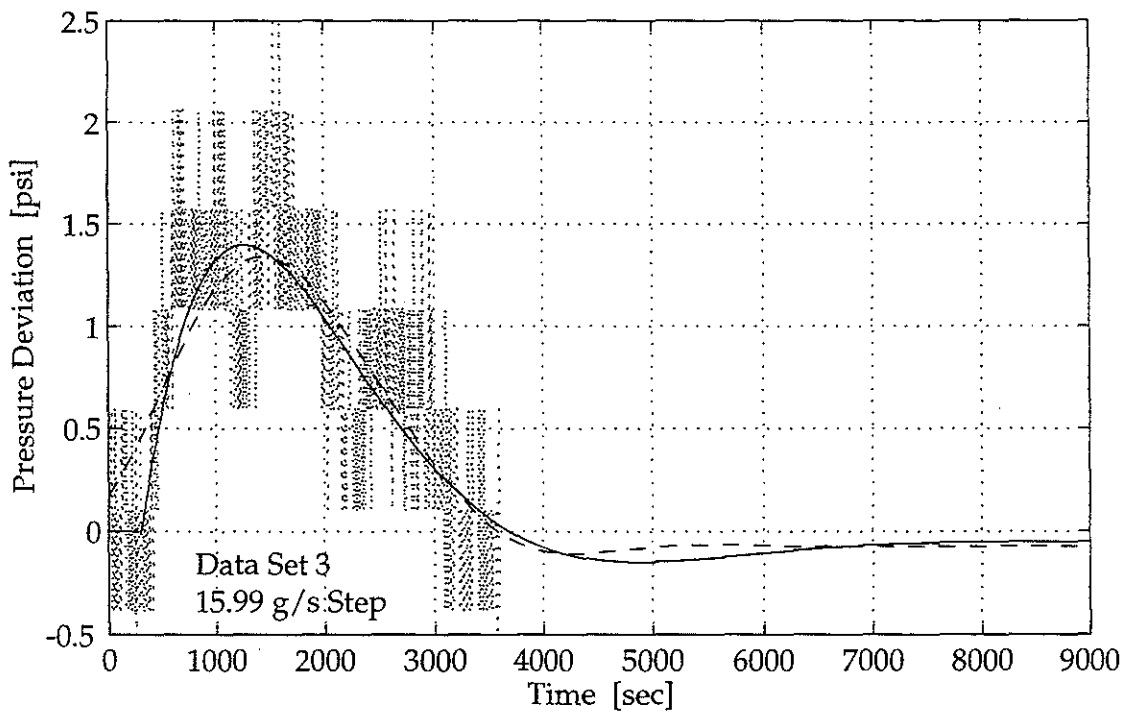

Figure 13. C0 P2 Data, Filtered Data, and Model for Step in B0 F2

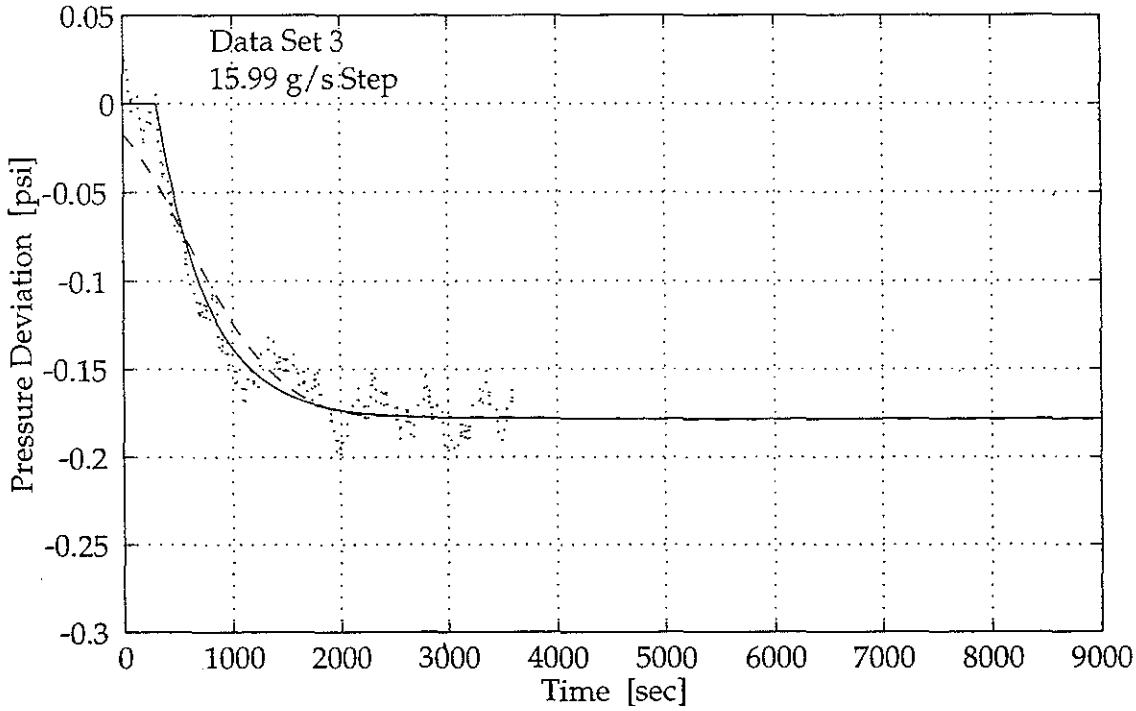

Figure 14. D0 P1 Data, Filtered Data, and Model for Step in B0 F2

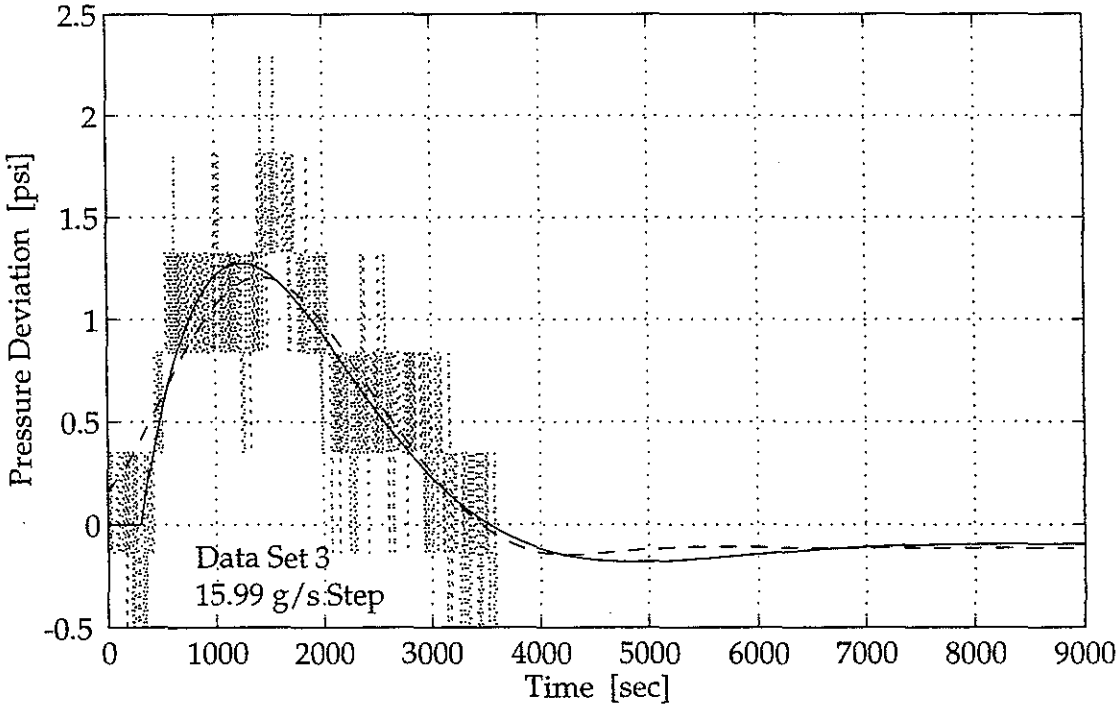

Figure 15. D0 P2 Data, Filtered Data, and Model for Step in B0 F2

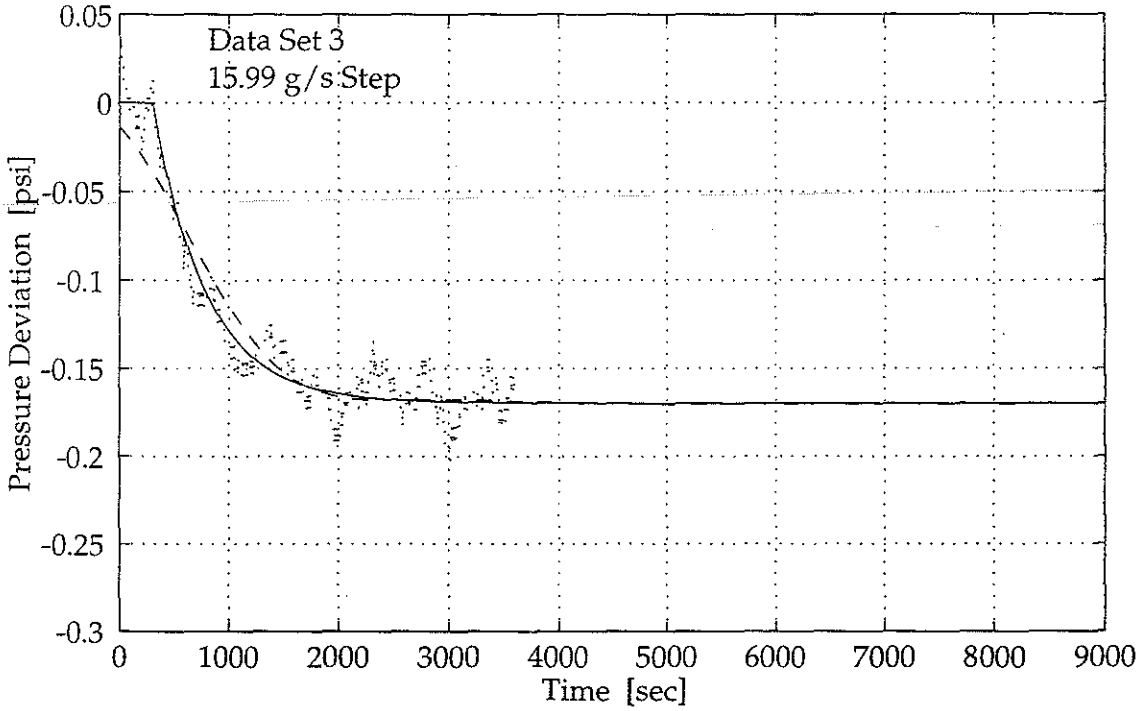

Figure 16. E0 P1 Data, Filtered Data, and Model for Step in B0 F2

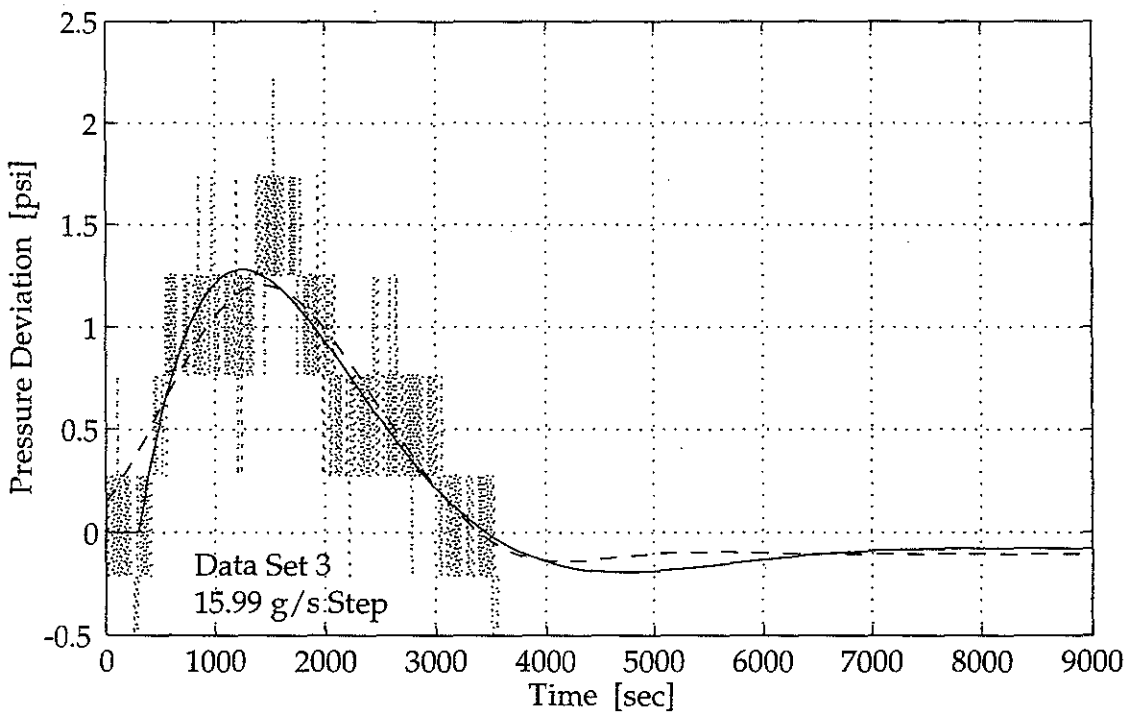

Figure 17. E0 P2 Data, Filtered Data, and Model for Step in B0 F2

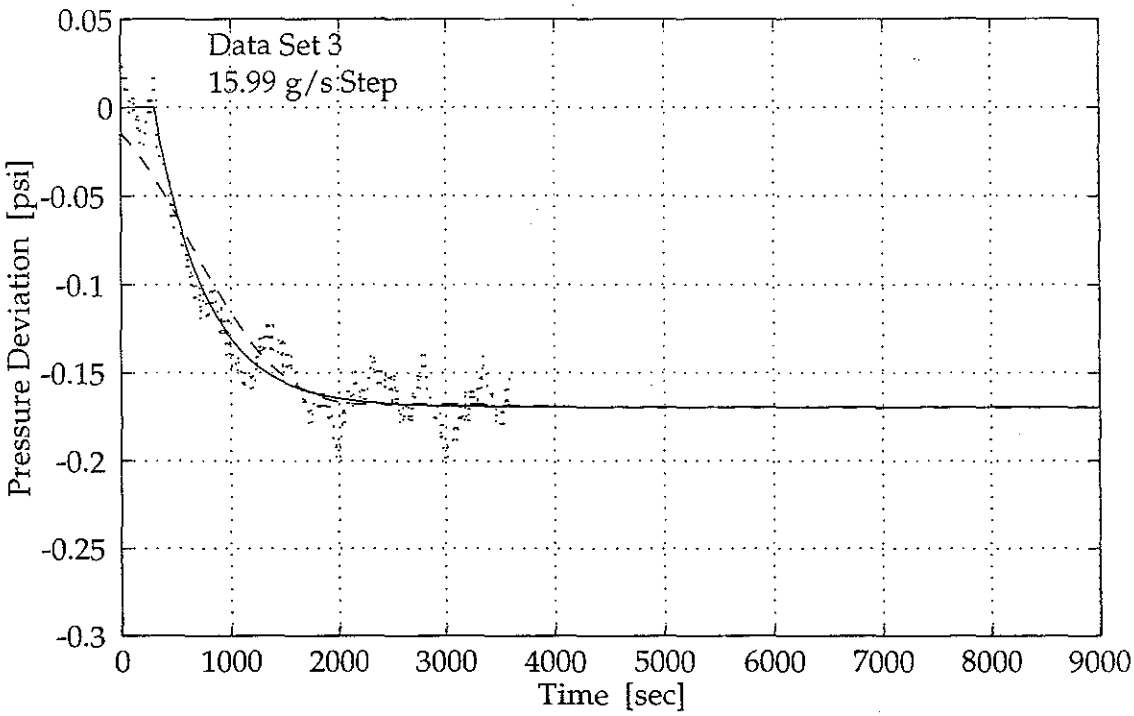

Figure 18. EA P1 Data, Filtered Data, and Model for Step in B0 F2

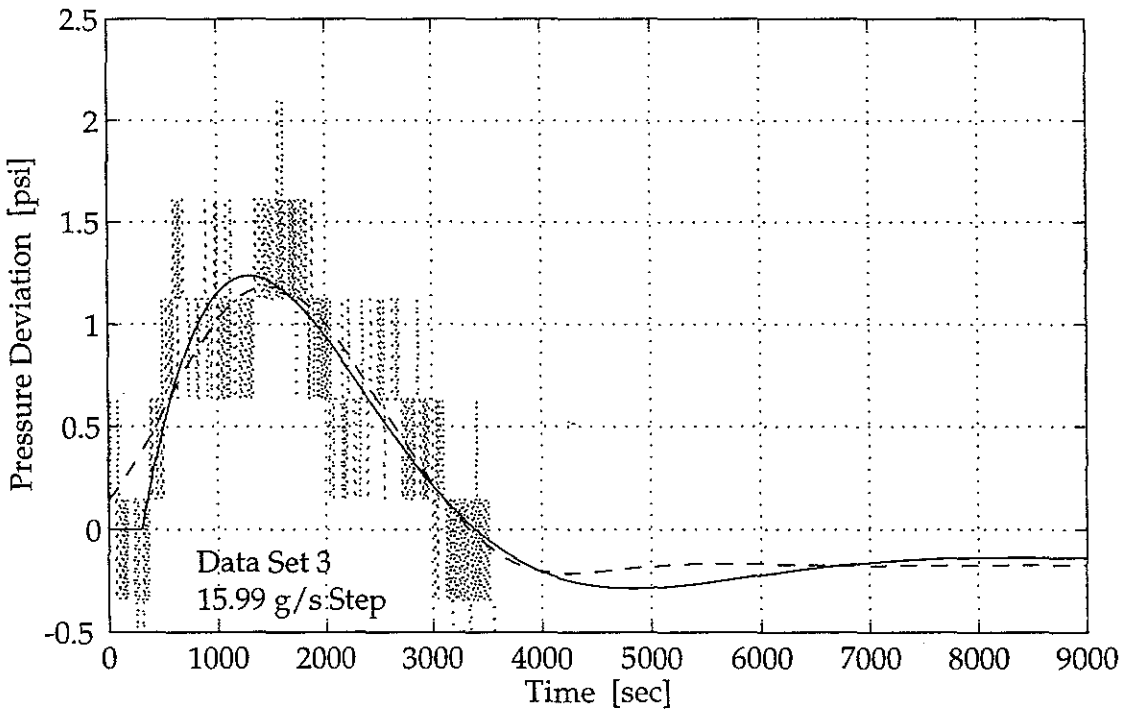

Figure 19. EA P2 Data, Filtered Data, and Model for Step in B0 F2

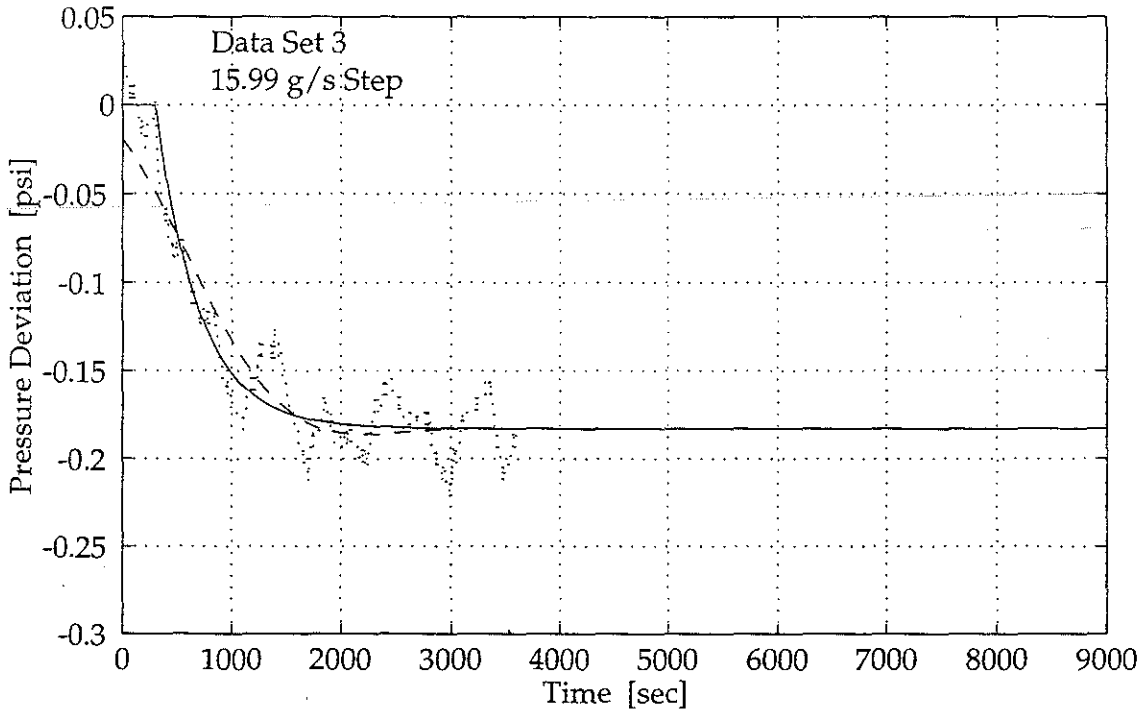

Figure 20. F0 P1 Data, Filtered Data, and Model for Step in B0 F2

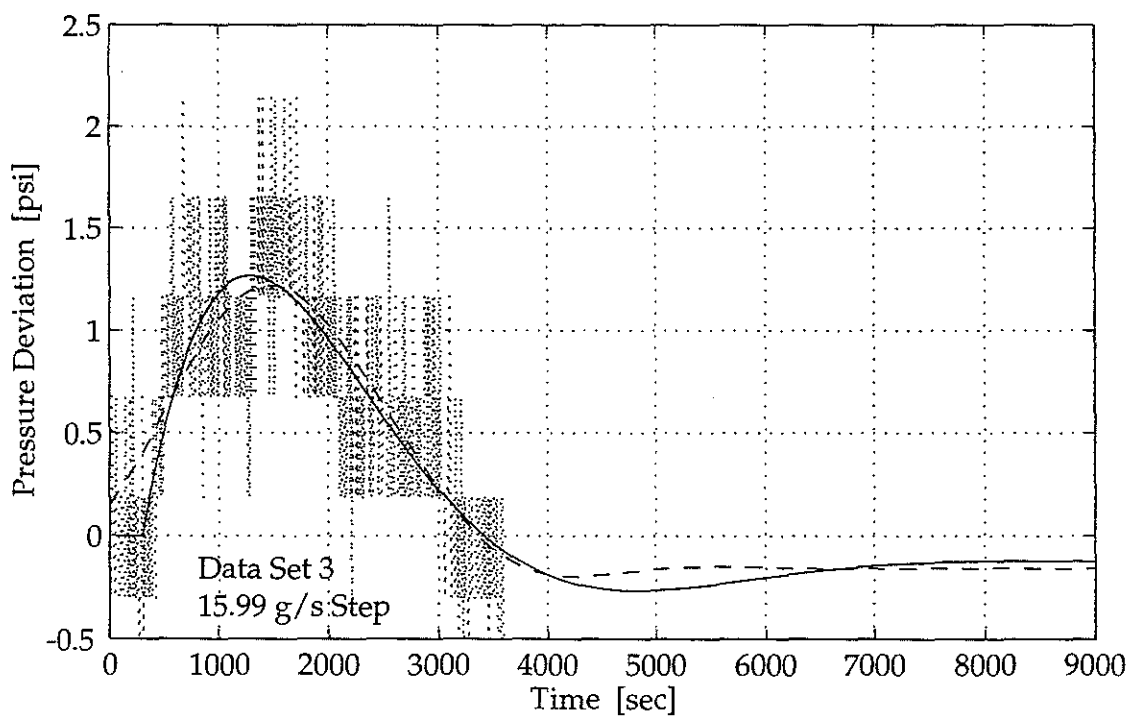

Figure 21. F0 P2 Data, Filtered Data, and Model for Step in B0 F2

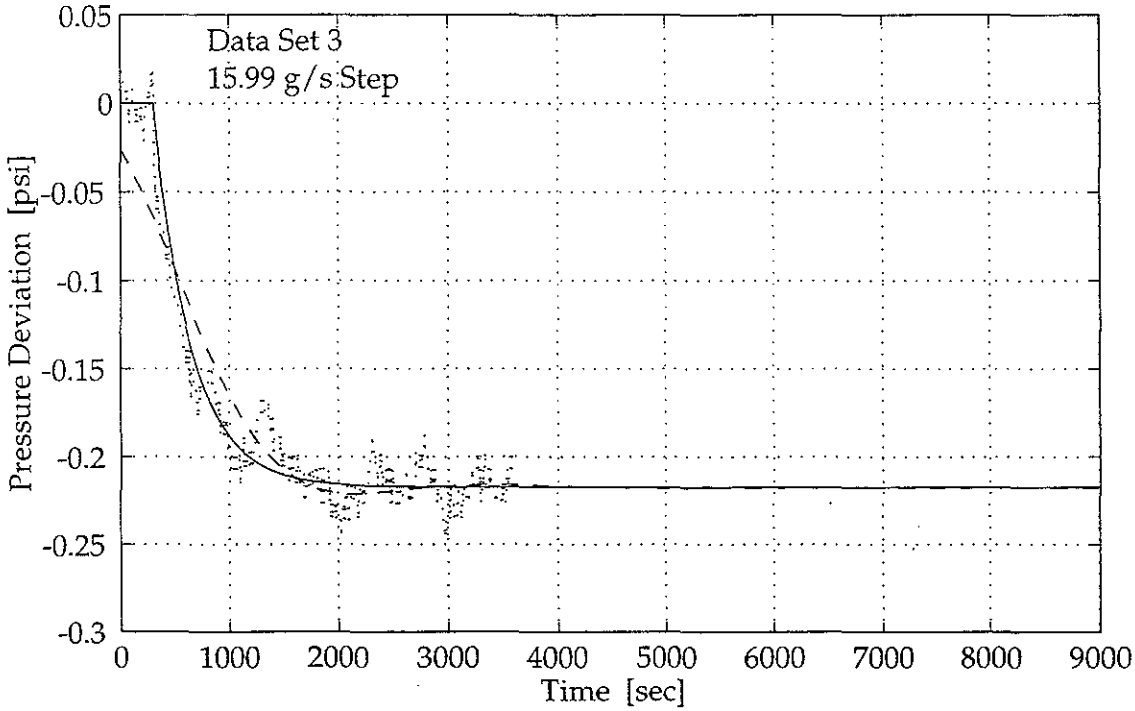

Figure 22. SY P1 Data, Filtered Data, and Model for Step in B0 F2

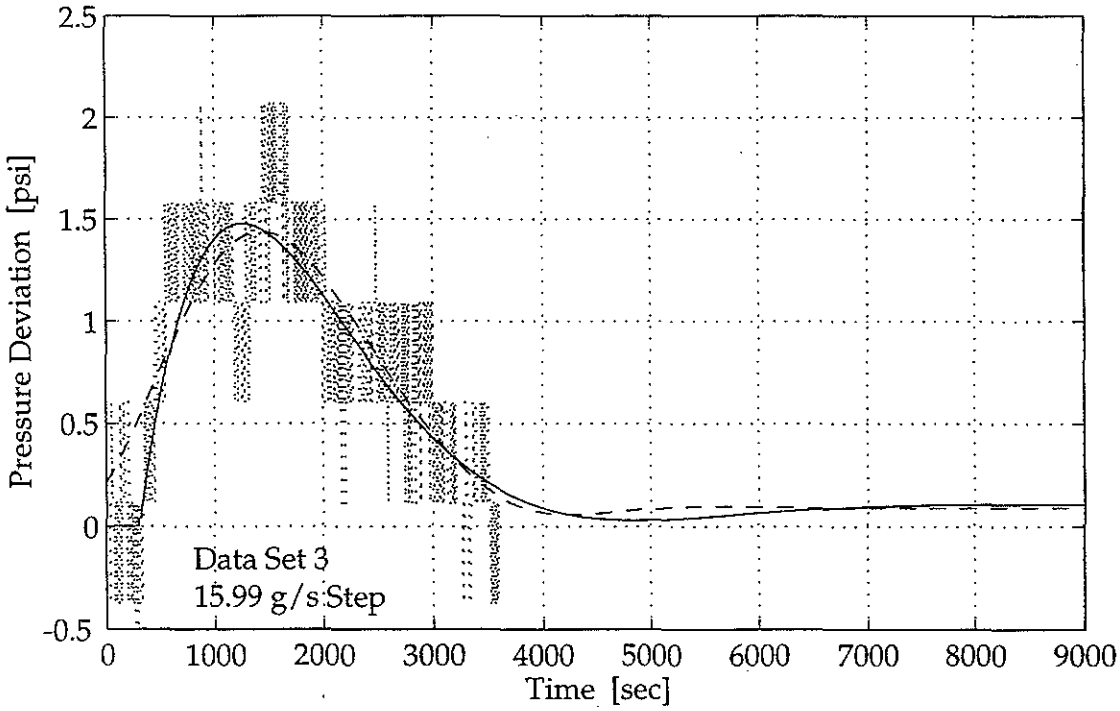

Figure 23. SY P2 Data, Filtered Data, and Model for Step in B0 F2

a step function. All flow plots have been extended with constant values found from the average of some number of final points. This emphasizes that a value other than zero is reached, that is the flows do not return to their original value.

Figure 4 indicates that the effect of suction header pressure on compressor flow rate can not be ignored in a complete explanation of the dynamics of the system. The reduction in suction pressure around the ring due to purposely increasing F2 flow at one location causes reduction in F2 flow at all other locations having constant compressor valve positions. The flow reduction is proportional to the ratio of the absolute values of suction pressure (designated P1a) after the step in manipulated flow to the P1a value prior to the step. Note that P1 can not be applied directly in this ratio since it is only a deviation variable and is not directly related to the helium gas density. This phenomena becomes apparent now because of the subtraction of initial values and summing of small signal changes.

These flow dynamics affect the P1 and P2 pressures in different ways. The pressure data for set 3 for all ring locations is represented by the dotted lines in Figures 6 through 23. The dashed and solid lines are explained later. Here, P1 appears as an exponentially decaying signal with a minor oscillation superimposed. It has no overshoot beyond the final value, contrary to the characteristic seen for the summed flow of Figure 5. Thus P1 reacts to the lower frequencies that make up the step input in compressor valve position, and comes to a final value presumably related to the final value of the sum of all flows.

P2, instead, follows the pattern of the summed F2 flow, indicating a somewhat faster response to flow changes. As a result, there is a large overshoot in P2 pressure before it starts to settle to a final value. At this point a significant problem with the design of the experiment becomes apparent in that the data stream is not sufficiently long to show an ending steady state P2 value. The design assumption that disturbances die out in the discharge header before they do in the suction header proves to be incorrect. Still, useful information can be obtained from the P2 data and it is shown later that the test duration is sufficient to obtain at least a course model. Another characteristic about the P2 data is the digital nature of the signal. P2 is measured with a 0 to 500 psig transducer with 0 to 5 volt output fed into a 12 bit analog to digital converter having an input range of -10 to 10 volts. The resulting resolution of the readback is 0.488 psig, which is clearly seen in Figure 7. Note that, for some unknown reason, the P2 data at the A0 location is the most noisy of all locations. As it turns out, the A0 data is also the most noisy for all other data sets.

Differences in pressure response with location around the ring can be seen in Figures 6 through 23. The response of P1 is progressively lower with distance away from the point of flow change at B0. Despite the high level of noise in the signal, it is seen that the peak value and final values of P2 also generally becomes less with distance away from B0. Whether because of some long term drift or because of the significant F2 flow reduction caused by the lowering P1, this data indicates that the final value of P2 actually falls back to roughly its initial value every where except at the B0 and BA locations.

3.3.2 Modeling the Response to Step Inputs. The data and theory suggest a direct link between continuous flow and the pressure signals over time, but there are advantages to relating pressure response to only the initial compressor loader valve step change. With one modification, the initial step only is what is used for the black box model developed here. Instead of working with the non-linear loader valve position input, the resulting flow step change is used as a model input. This flow step is thought of as constant, though Figure 4 shows some decrease with time of even the B0 F2 flow. Since correlation (2.6) seems to works well, the flow step is calculated directly from the loader valve position change, thus avoiding the need to pick a value out of the noisy flow data. Equation (2.6) is corrected for differing compressor inlet pressure by multiplying by an initial P1a value and dividing by a constant, Pc, on which the correlation is based. The value of Pc is 15.9 psia (see Table 1).

With the variables defined, modeling becomes a process of trying a model form with the Toolbox routines and modifying the form until a suitable one is found. The logical starting point of low order models of both the ARX and OE forms. Recall that the ARX form contains only the A(q) and B(q) polynomials of equation (3.10), so that G(q)=B(q)/A(q) and H(q)=1/A(q) in Figure 3. The OE form contains only B(q) and F(q), thus G(q)=B(q)/F(q) and H(q)=1. When used with the step input and noisy pressure data, however, effort with both these forms proves to be futile. The routines are unable to cope with the noise of either the P1 or P2 signals, and apparently, for numerical reasons beyond the scope of this study, optimizing model coefficients can not be found. Graphically, the results suggest that the software

attempts to do a good job of fitting the first few pressure data points after the step, which results in a highly unstable system. Attempts with selected higher order noise model forms also become unstable. Some other means of modeling the varied noise and disturbances seen in P1 and P2 may be successful, but are not pursued with the black box model.

Rather than continue a potentially endless search for a form that models the signal noise, prefiltering of the data provides a route to the low order plant models that are suggested by the data plots. Low pass non-causal filtering, that is adjusting a data value based on values at both past and future time points, works well if certain issues are addressed. For one, an extension must be made to the data in order to give weight to the final values in the data set, particularly for the P2 data. Otherwise, the filter data stream exhibits ending values higher than expected due to the averaging of past higher values and no future values. Also, in order to maintain the appropriate relation between input and output, both signals must be filtered in the same manner.

Routinely, compressor flow input data set pressures are extended to 2400 points by appending points whose constant value is an average of some number of points at the end of the data set. In the case of data set 3, the last 120 P1 points are averaged, and the last 40 P2 points are averaged to produce extension values. The extended data set is then filtered with a third order Butterworth filter having a cutoff frequency of 0.002 Hz (0.0126 rad/s). The filter order and cutoff frequency are optimized to produce the smooth pressure curve leading to the smallest least squared error between noisy data

and the model output. The filtered data stream is shown as the dashed line in Figures 6 through 23.

With the noise and disturbances gone, the ARX routine of the Toolbox quickly calculates a result. P1 is easily fit to a first order ARX model form

$$P1(t) = \frac{b11}{1 + a11 \cdot q^{-1}} \cdot \frac{P1a(t-1)}{Pc} \cdot [f\{LS(t-1)\} - f\{LS(t-2)\}]. \tag{3.23}$$

where [f{LS(t-1)}-f{LS(t-2)}] is the uncorrected F2 deviation flow rate variable as a function of the compressor valve position given by (2.6). The noise term in the ARX form is not included in (3.23) because it represents only the unuseful difference between the model and the filtered data. P2 is modeled with somewhat more work needed to obtain a proper order, and has the ARX form

$$P2(t) = \frac{b12 + b22 \cdot q^{-1}}{1 + a12 \cdot q^{-1} + a22 \cdot q^{-2}} \cdot \frac{P1a(t-1)}{Pc} \cdot [f\{LS(t-1)\} - f\{LS(t-2)\}]. \tag{3.24}$$

Only variation in the values for the model coefficients is needed to relate P1 and P2 at any location to the flow step in data set 3. The simulation resulting from models (3.23) and (3.24), given an input step size of 16.0 g/s, is shown as the solid line in Figures 6 through 23. These model forms are also successfully applied to all other F2 flow input data sets. The results for these sets are discussed in Section 4.1. The MATLAB program "ModF2.m" written to produce the model coefficients and related figures is given in Appendix A.

3.4 Black Box Model for Inventory Flow Inputs

In this section the data of set 22 is analyzed to form a model relating inventory flow changes, F2, to discharge pressure, P2. This also provides an

example of the procedure generally used to process other similar data sets. Later, in Chapter IV, the SISO models produced in this manner are assembled with those for compressor flow inputs to create an overall system model.

3.4.1 Characteristics of Data. The data of set 22 is prepared as described in Section 3.3.2 and is shown in Figures 24 through 45. The controlled input is valve CL, which steps from 61.2% to 57.1% producing the -8.7 g/s change in flow FC appearing in Figure 24. The flow step in this case is determined directly from flow data since data to produce a valve position-to-flow correlation was not readily available. The flow change causes pressure P2 to increase, which in turn causes flow FD to increase slightly across the fixed position valve DL. This effect is shown in Figure 25. Figure 26 gives the sum of the F2 flows, which shows considerable noise and disturbance but no significant trend. If all the flow out of the discharge header (FC and FD) is subtracted from all flow into the discharge header (the F2s), the result is the net flow change into the discharge header. That result is shown in Figure 27 and is useful to visually relate the net flow effect to the measured P2 values.

The pressure data in Figures 28 through 45 indicate P1 is not affected by changes in inventory flow but that P2 has a first order dependence on FC. This might be anticipated since F2, which is capable of changing P1, exhibits only seemingly random disturbances, perhaps from small movements in compressor valve. The disturbances F2 does experience may even be explained by compressor inlet pressure variation caused by disturbances to P1 originating from refrigerators.

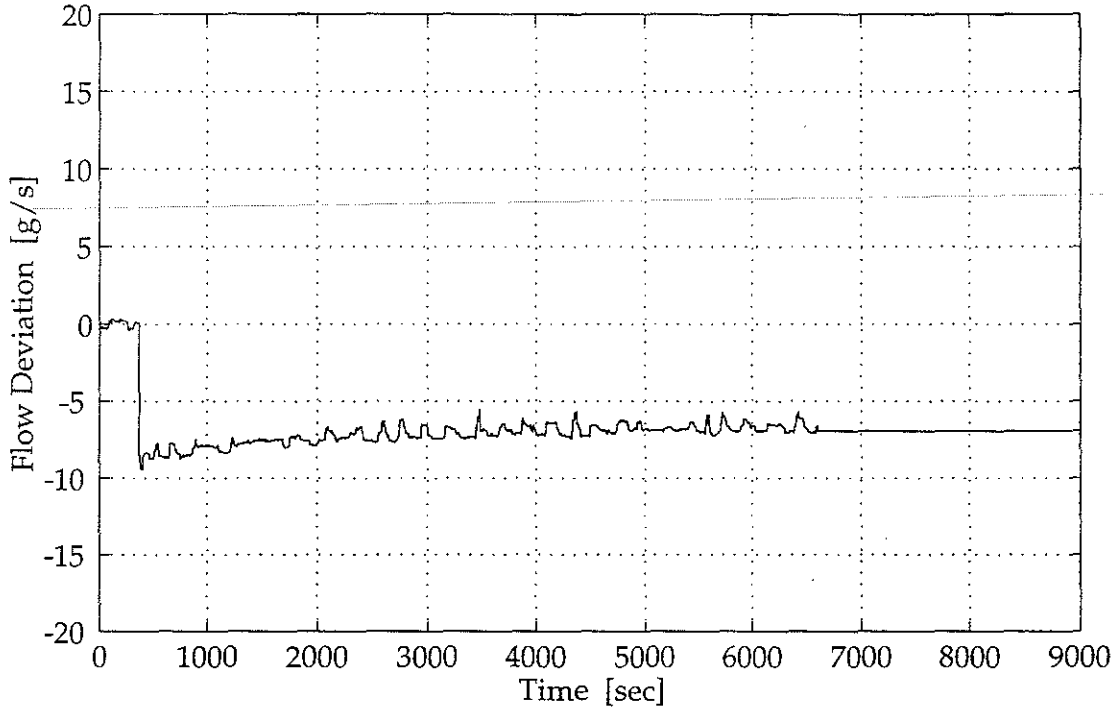

Figure 24. FC Inventory Flow from Data Set 22

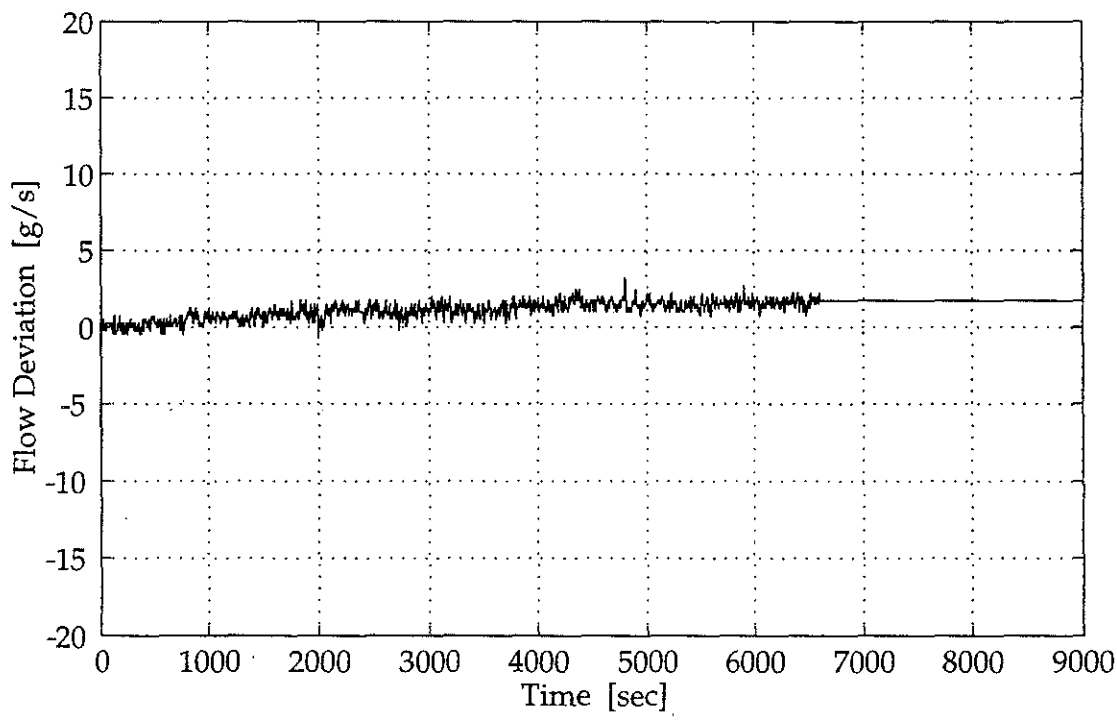

Figure 25. FD Inventory Flow from Data Set 22

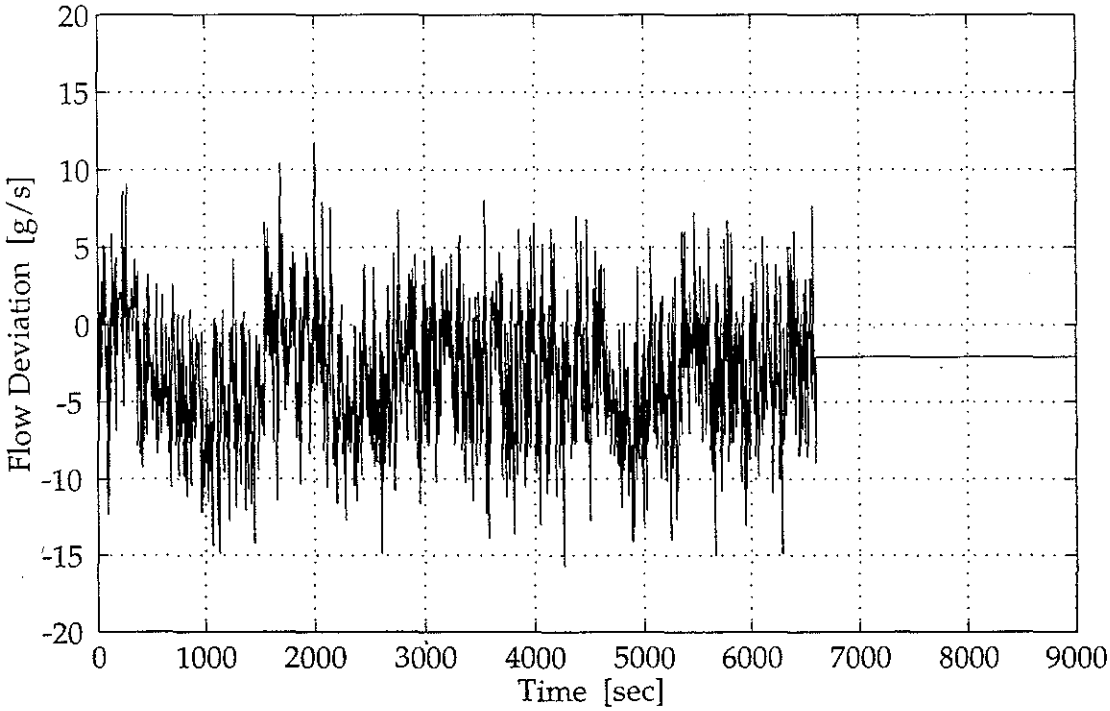

Figure 26. Sum of F2 Flows from Data Set 22

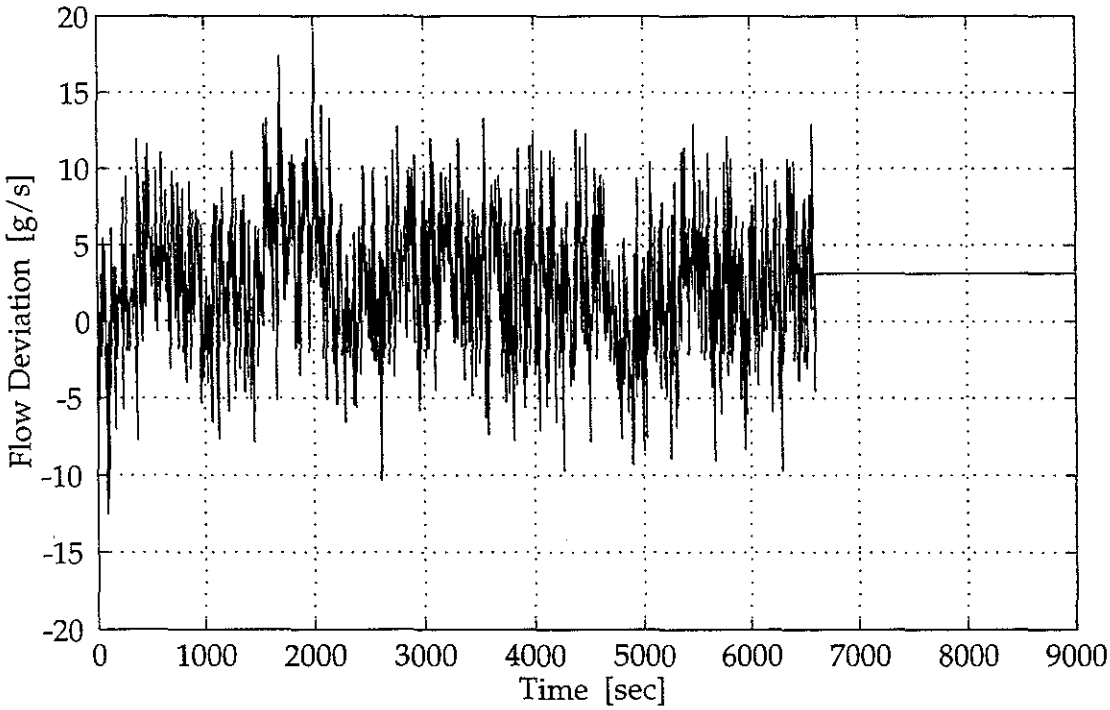

Figure 27. Net Flow Change into Discharge Header from Data Set 22

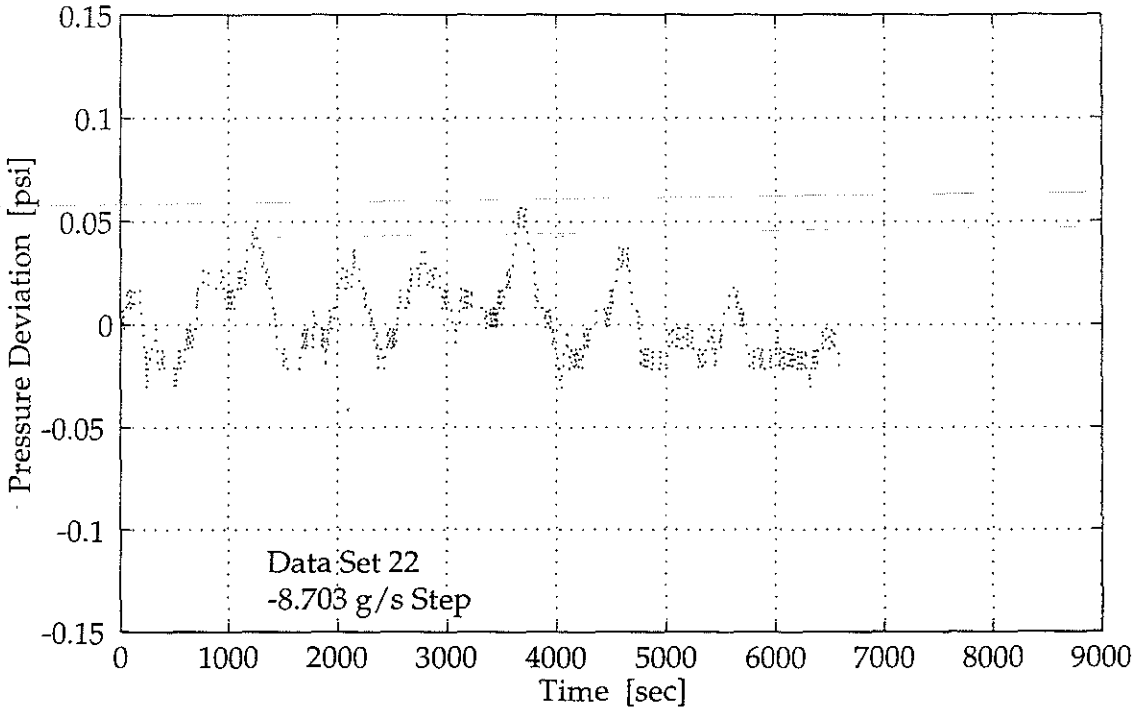

Figure 28. A0 P1 Data for Step in FC

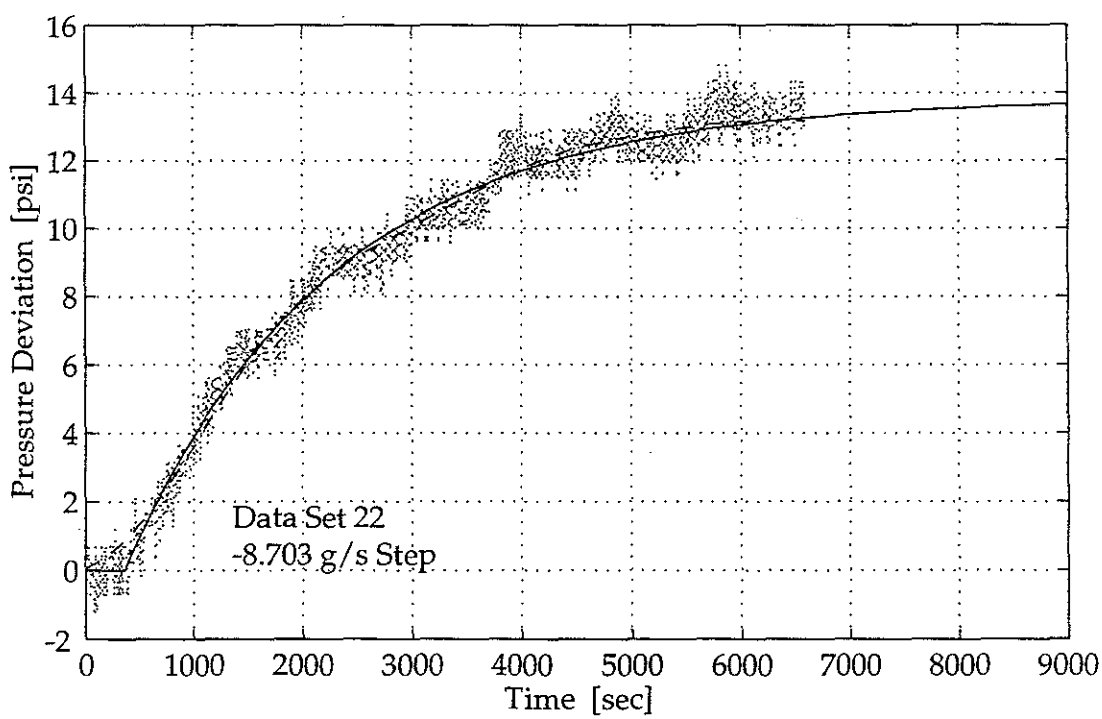

Figure 29. A0 P2 Data, Filtered Data, and Model for Step in FC

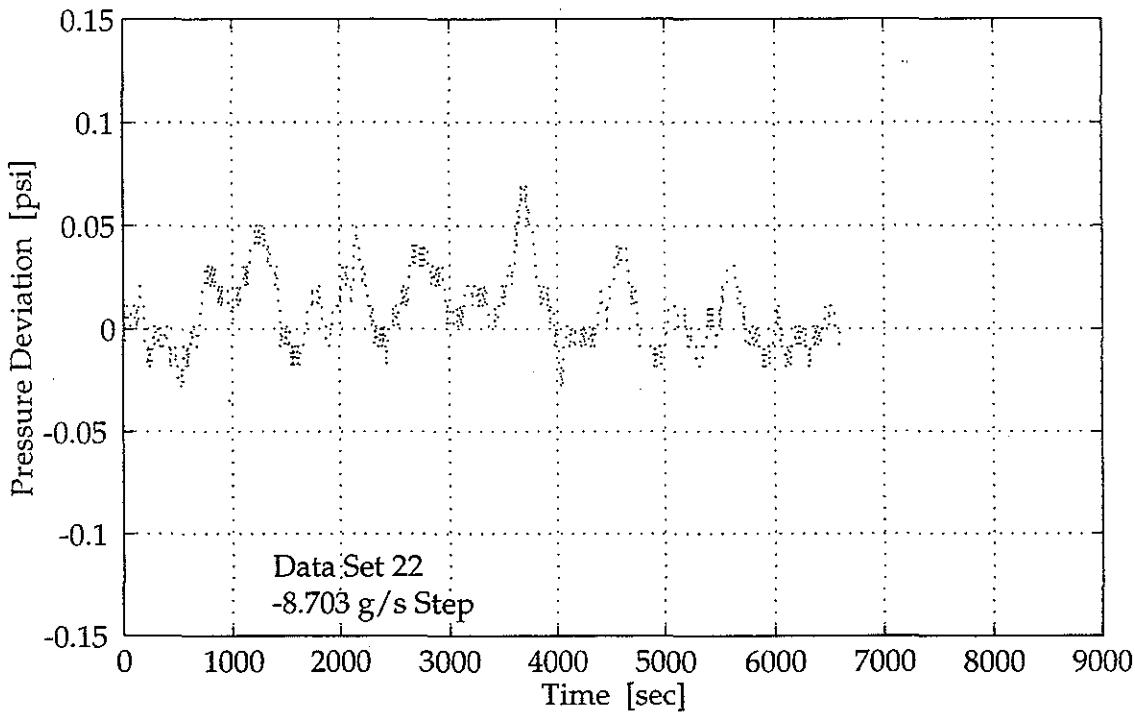

Figure 30. B0 P1 Data for Step in FC

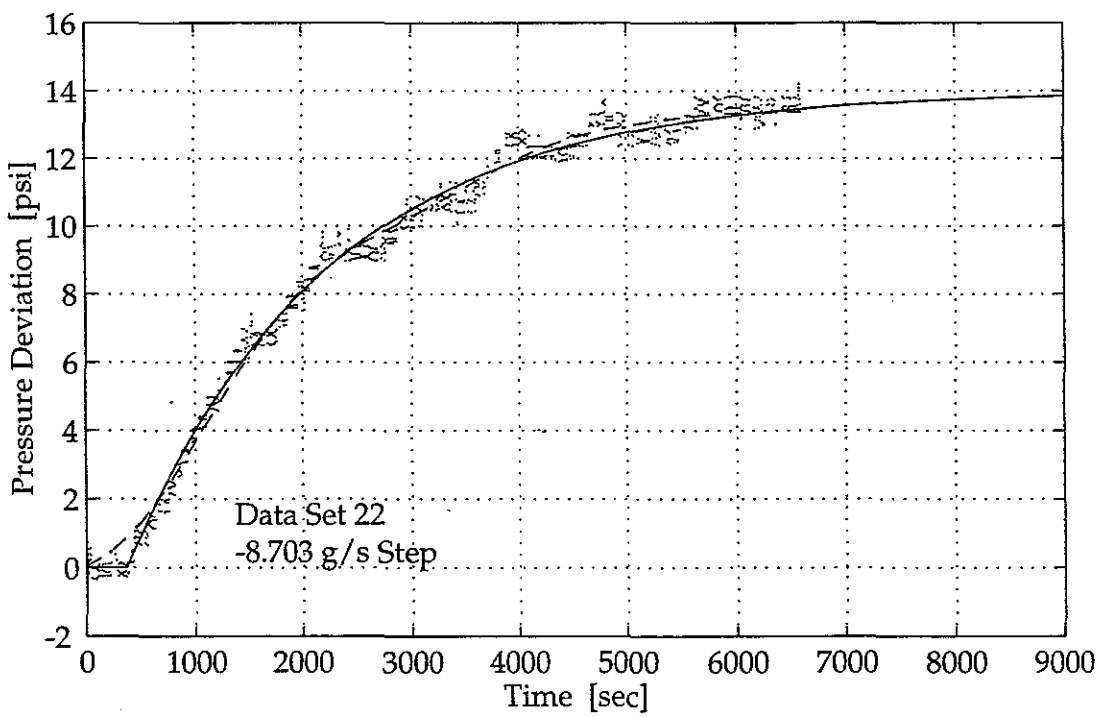

Figure 31. B0 P2 Data, Filtered Data, and Model for Step in FC

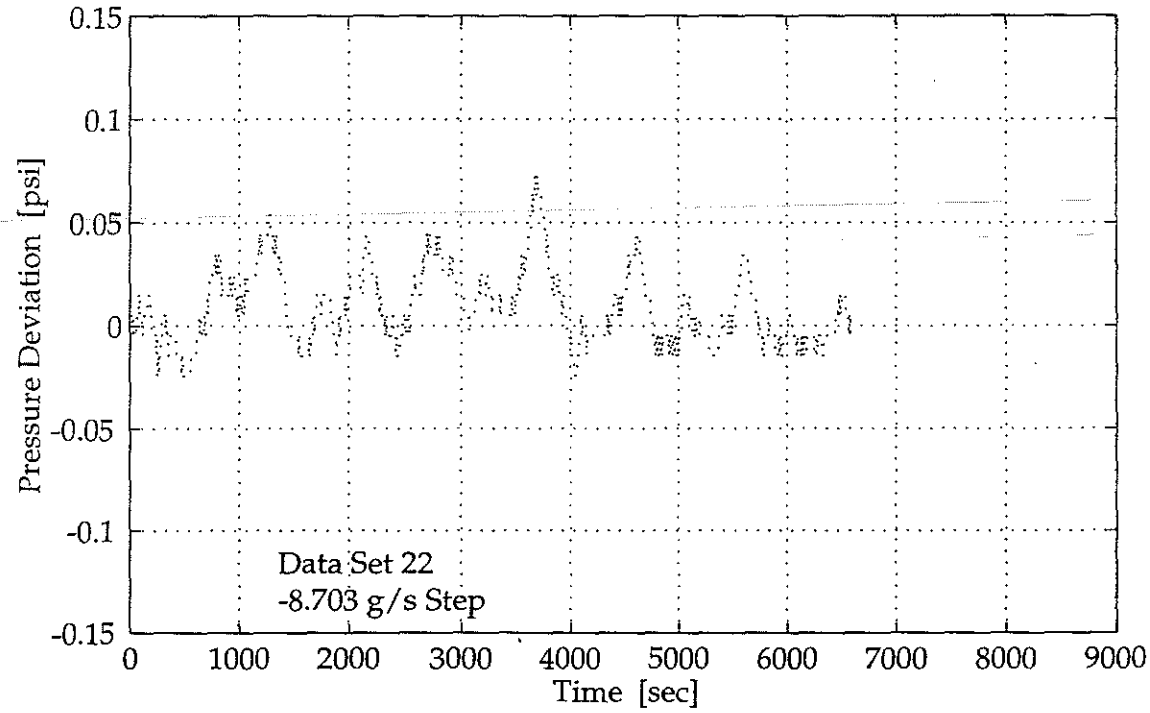

Figure 32. BA P1 Data for Step in FC

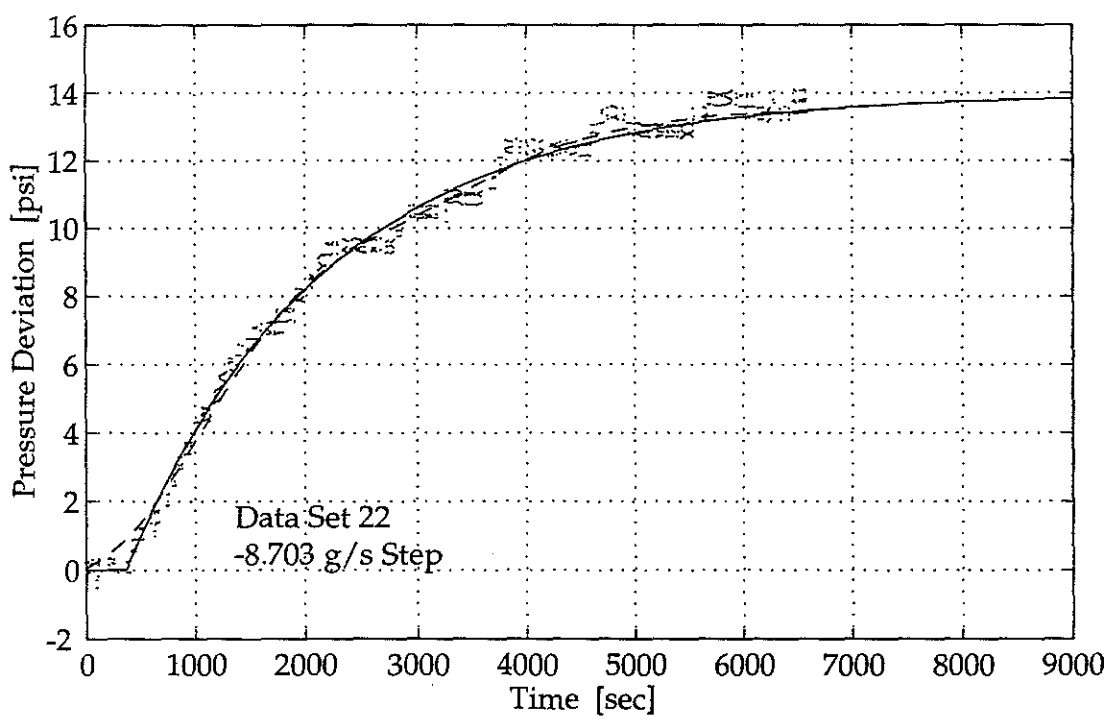

Figure 33. BA P2 Data, Filtered Data, and Model for Step in FC

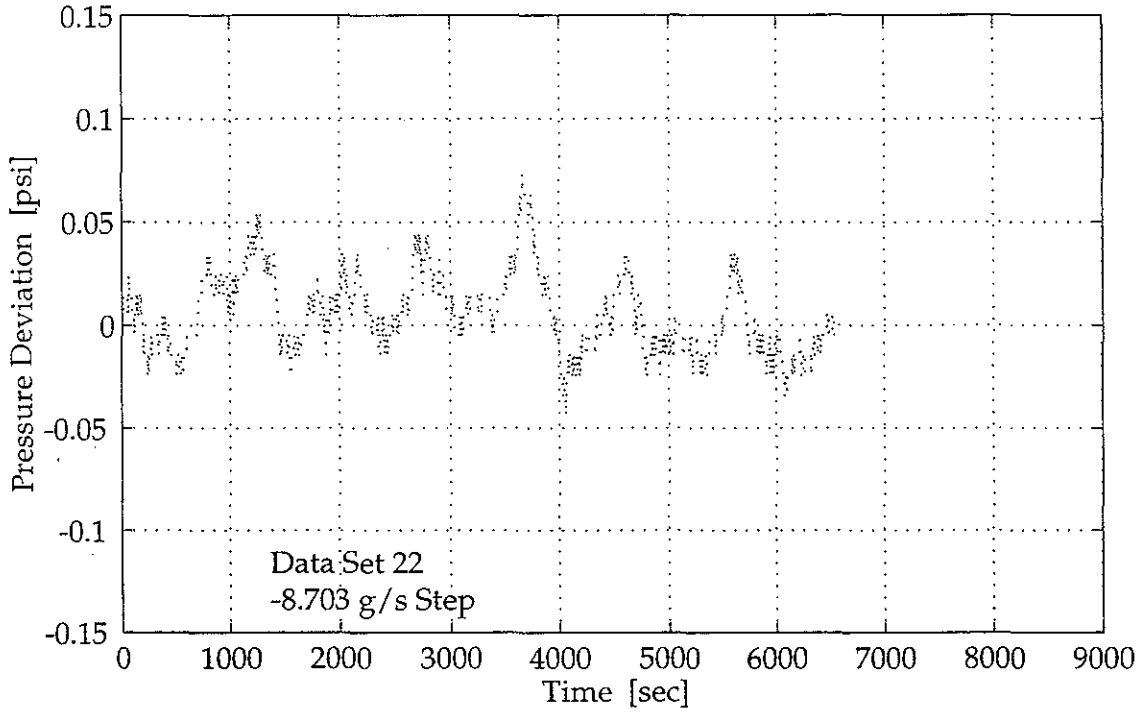

Figure 34. C0 P1 Data for Step in FC

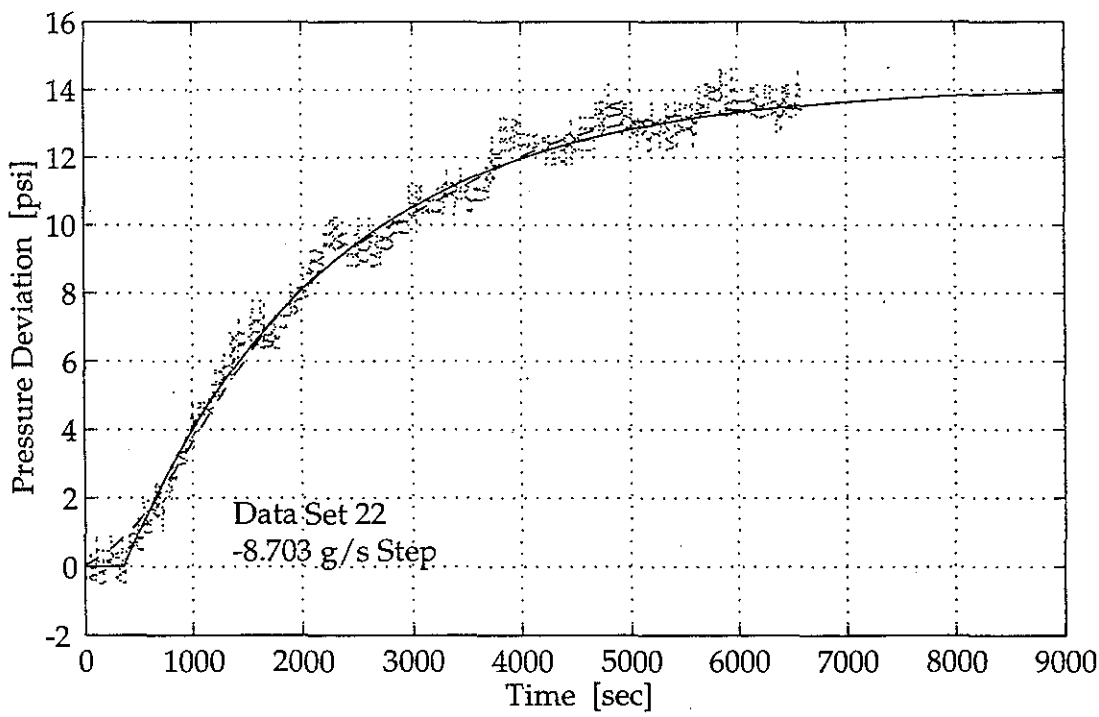

Figure 35. C0 P2 Data, Filtered Data, and Model for Step in FC

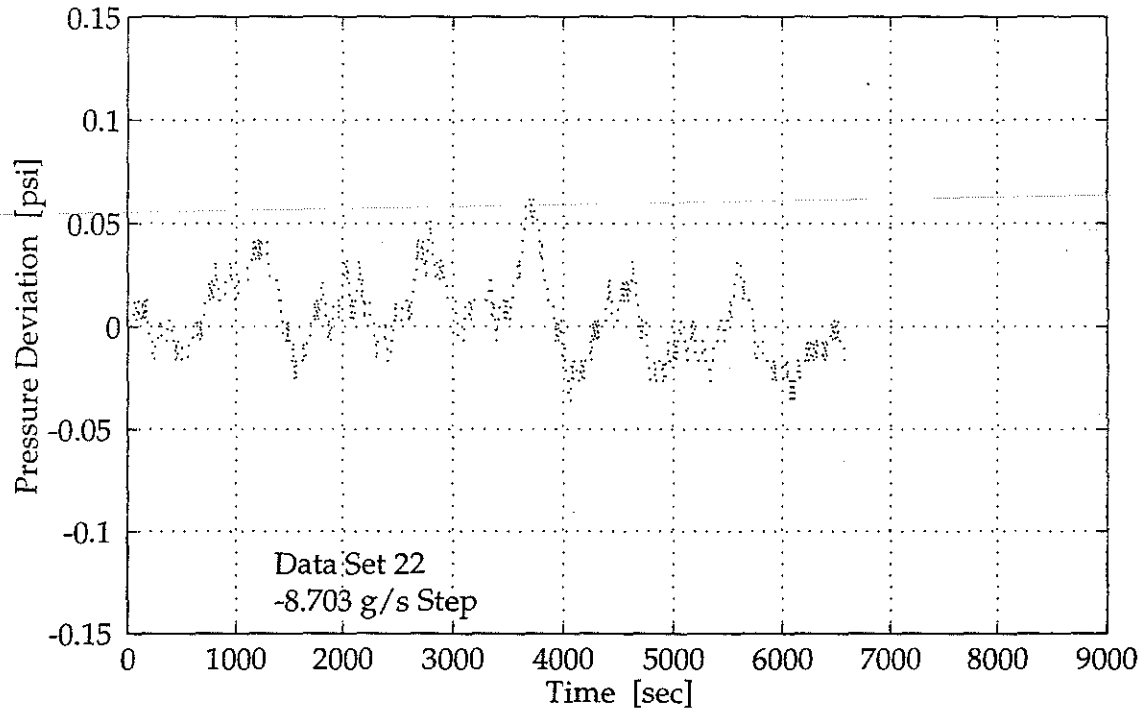

Figure 36. D0 P1 Data for Step in FC

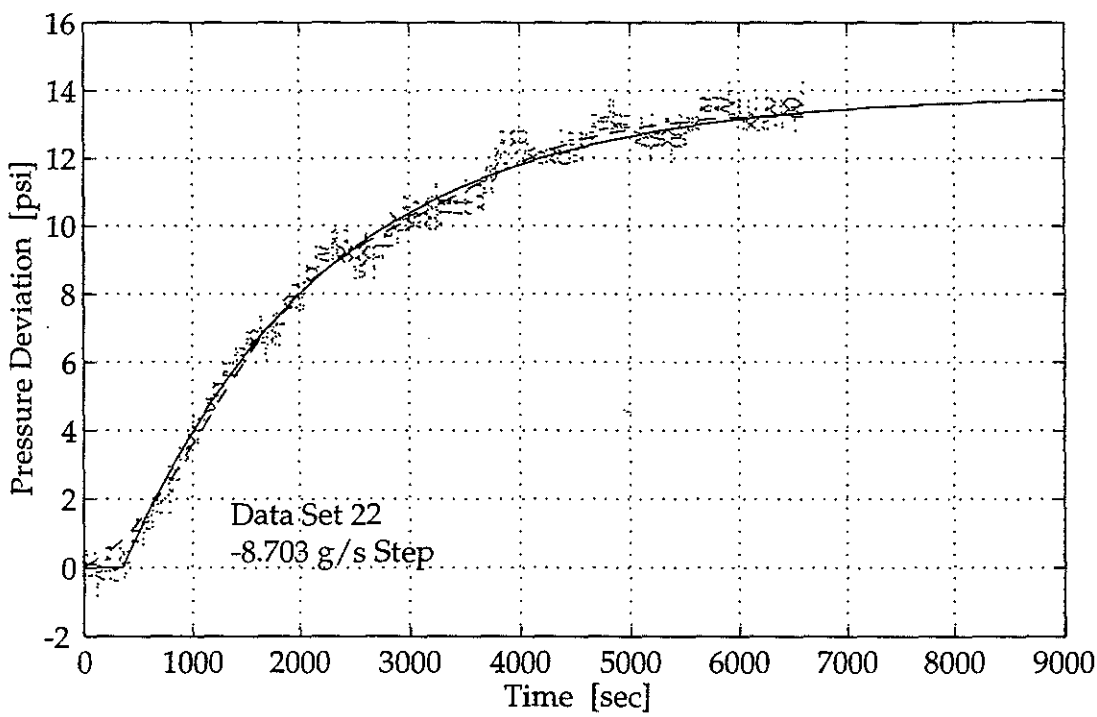

Figure 37. D0 P2 Data, Filtered Data, and Model for Step in FC

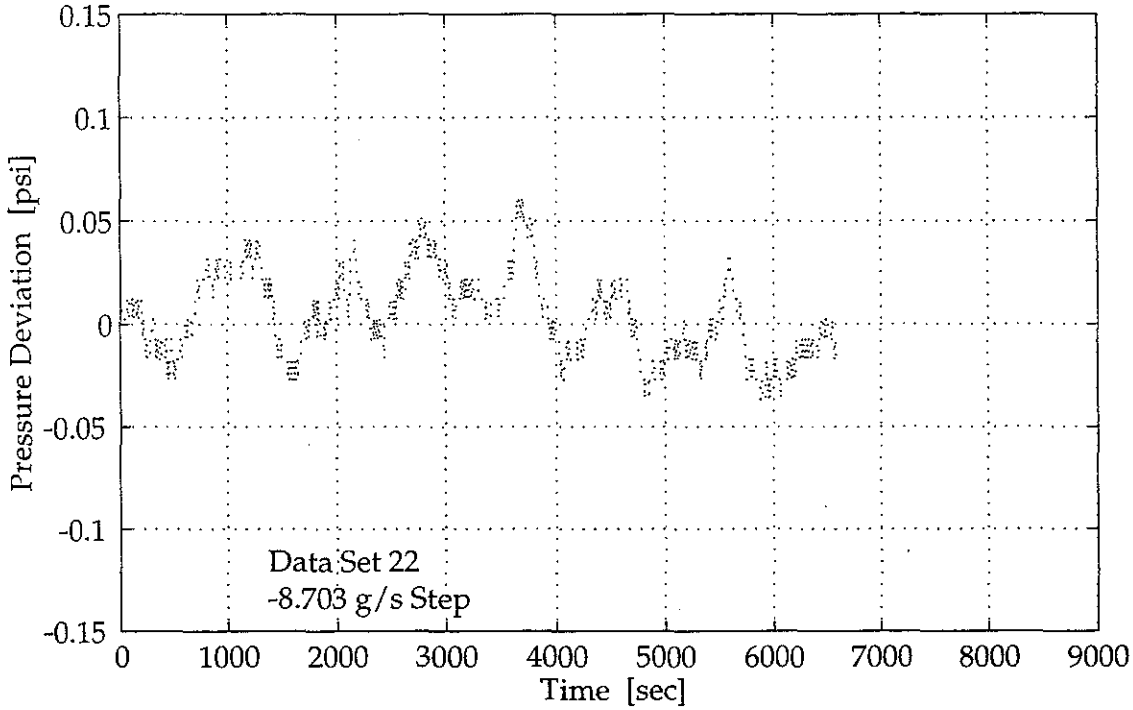

Figure 38. E0 P1 Data for Step in FC

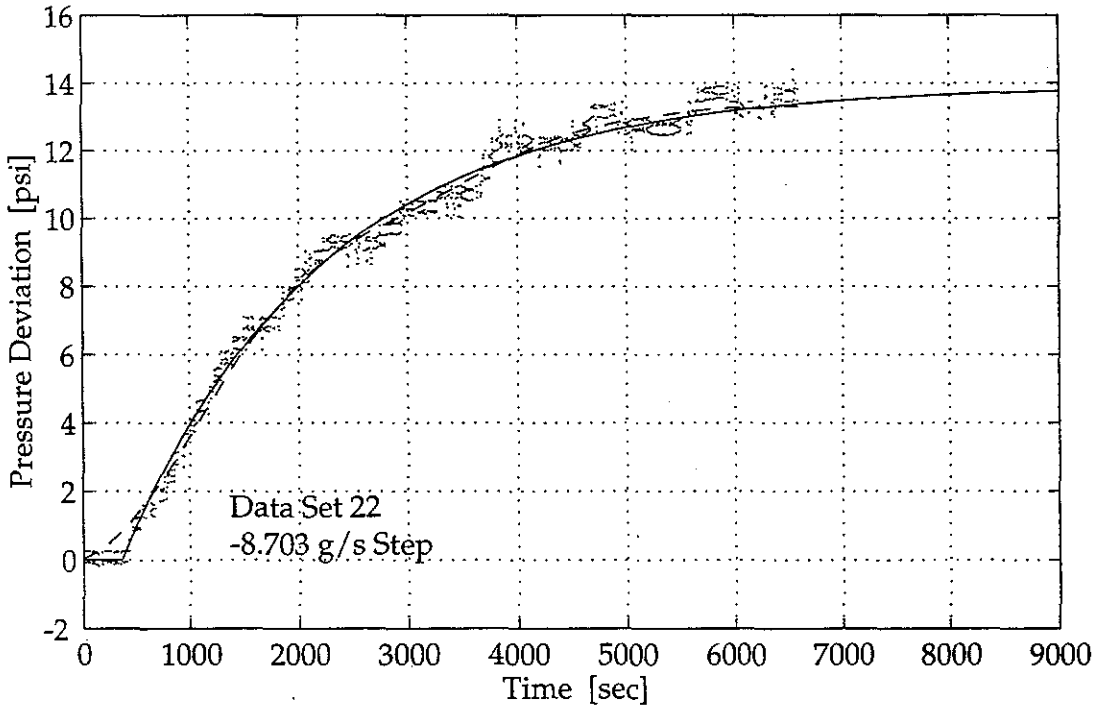

Figure 39. E0 P2 Data, Filtered Data, and Model for Step in FC

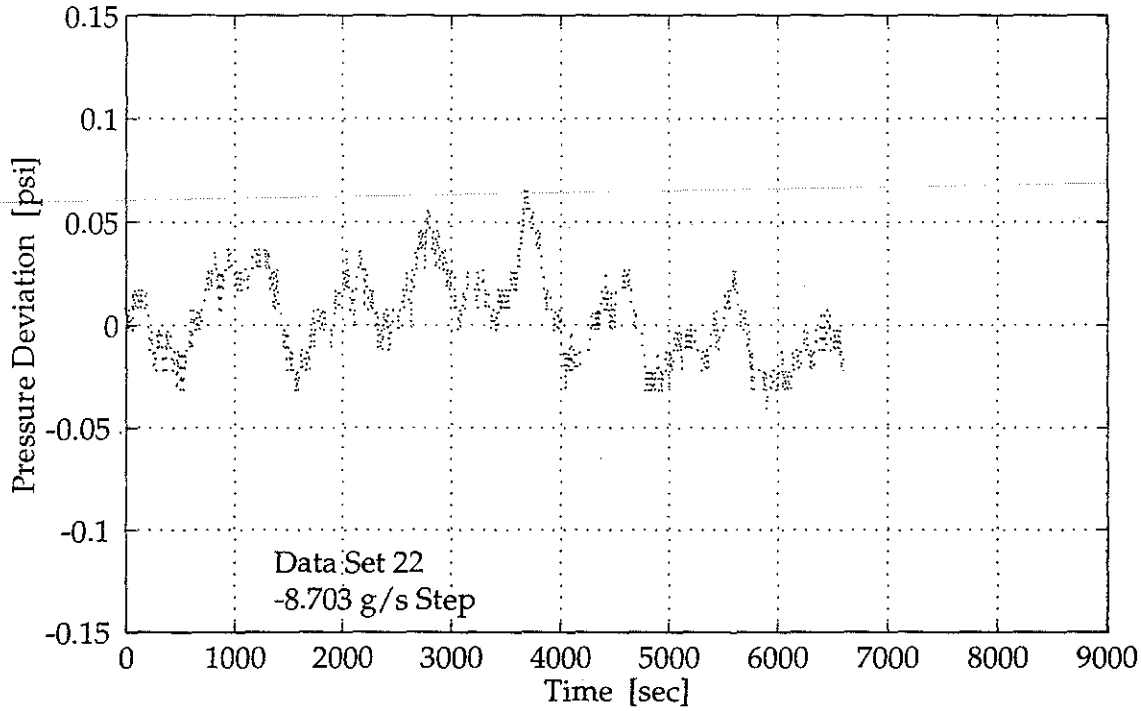

Figure 40. EA P1 Data for Step in FC

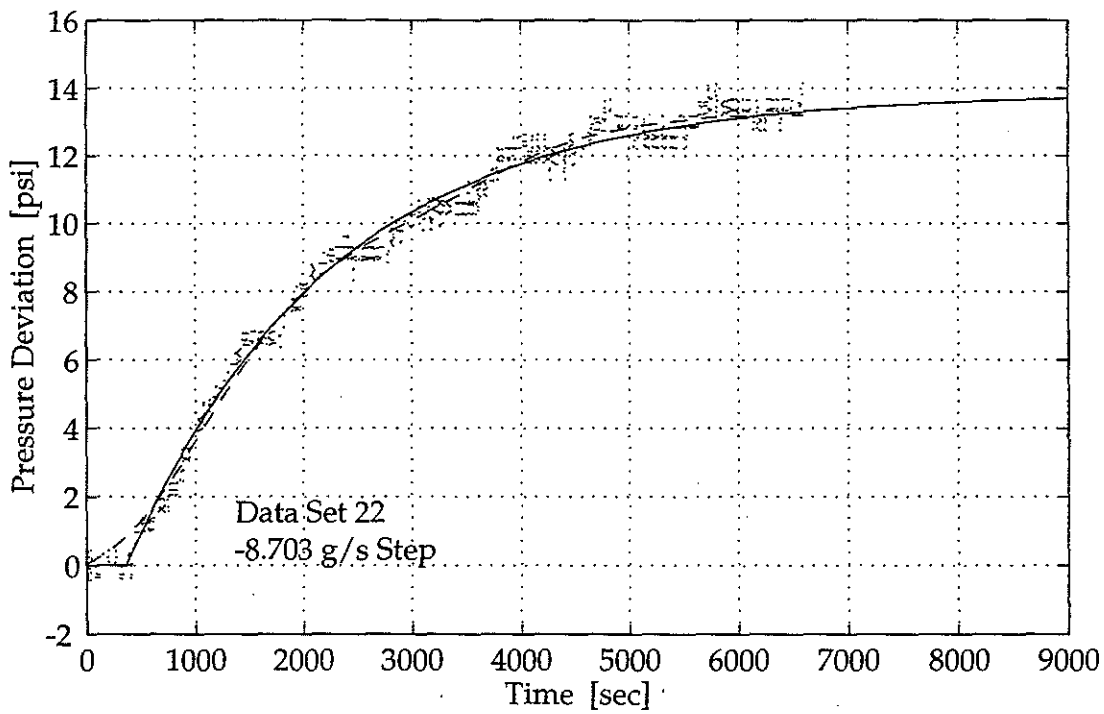

Figure 41. EA P2 Data, Filtered Data, and Model for Step in FC

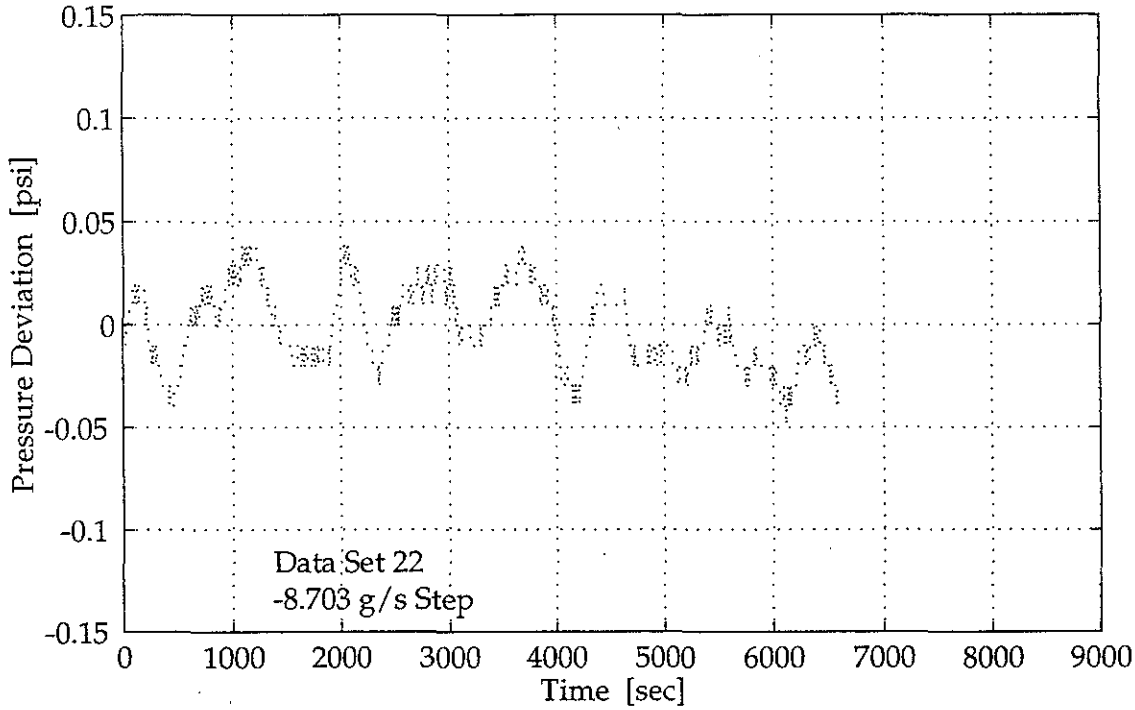

Figure 42. F0 P1 Data for Step in FC

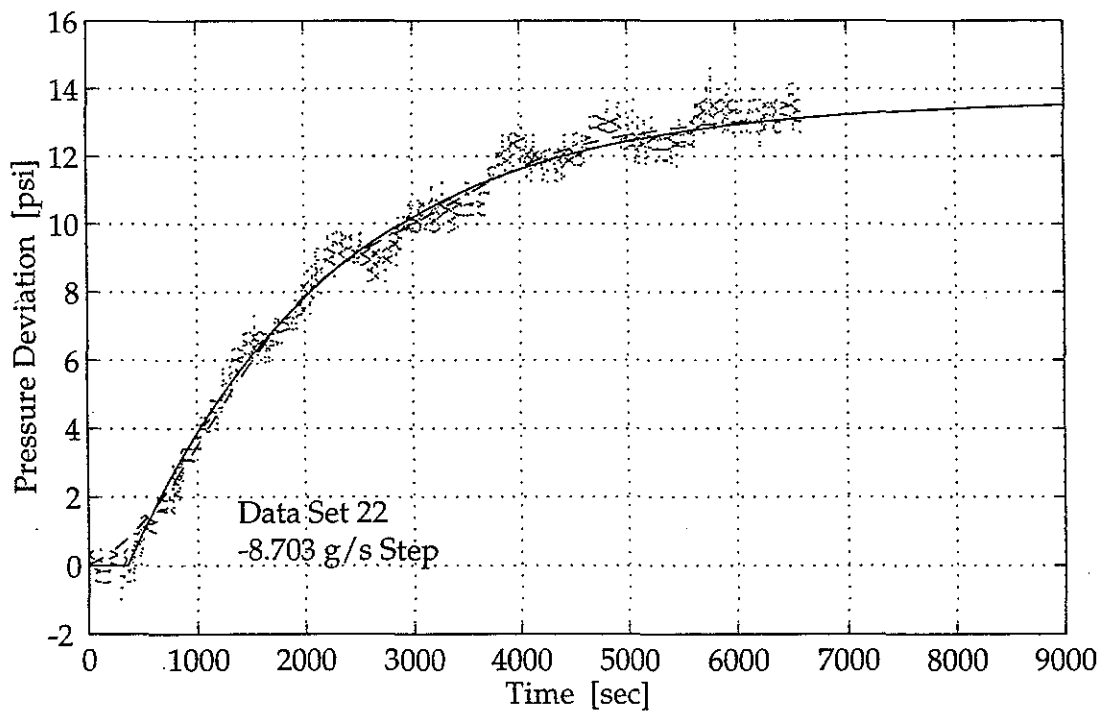

Figure 43. F0 P2 Data, Filtered Data, and Model for Step in FC

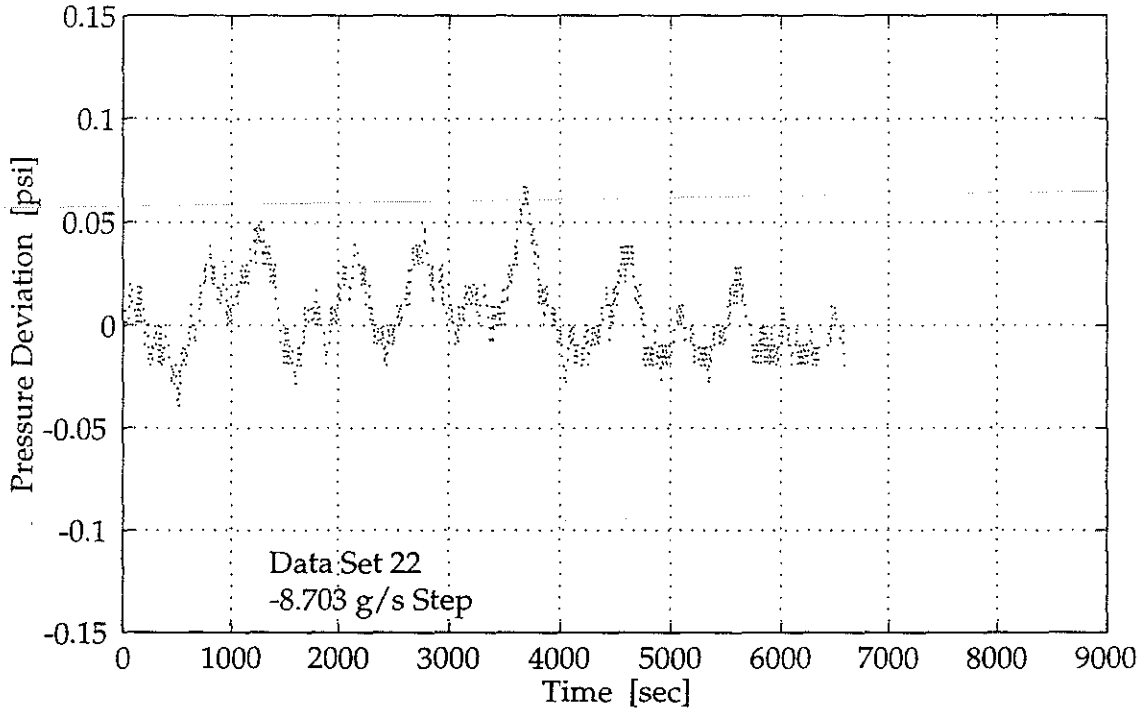

Figure 44. SY P1 Data for Step in FC

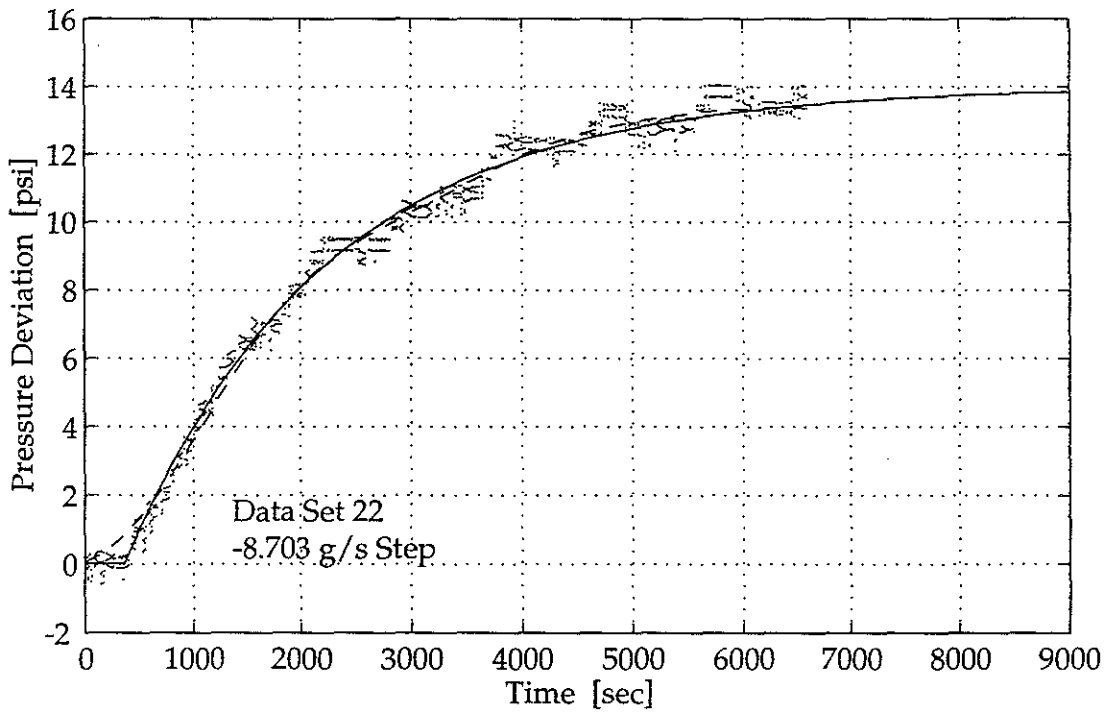

Figure 45. SY P2 Data, Filtered Data, and Model for Step in FC

3.4.2 Modeling the Response to Step Inputs. No model relating P1 to inventory flow changes is attempted, based on the above discussion. Following the modeling performed for compressor flow input, the P2 data for inventory flow input is extended and filtered. A first order ARX form (i.e. equation (3.10) having only A(q) and B(q) polynomials) fits this data well.

$$P2(t) = \frac{b13}{1 + a13 \cdot q^{-1}} \cdot FC(t-1) \tag{3.25}$$

Again, the dashed line in the figures represent the filtered data, and the solid lines are the model output, in this case responding to a -8.7 g/s input. This model form proves satisfactory for all of the inventory flow data sets, without extension of the data. Results for these other sets are presented in Section 4.2. The program "ModFC.m" used to calculate coefficients and produce figures for the inventory flow data and models is given in the Appendix.

3.5 White Box Model for Compressor Flow Inputs

When model identification involves the use of considerable physical insight it is sometimes referred to as a white box modeling, emphasizing that clear theoretical principles of the process help to determine the form of the model. To some extent, knowledge of a linear relation between flow and pressure has already assisted in creating the black box model relating compressor valve position change to pressure fluctuations. Further building on that knowledge, a method is outlined here to directly relate measured flows to pressure changes. The resulting model form provides clues to how the system would behave under some different control schemes to be presented in Chapter IV, Modeling Results and Discussion.

3.5.1 Details Suggested by the Black Box Analysis. Comparing Figures 5 and 7, the relation between the sum of flows into the discharge header and the P2 pressure can be imagined. Though these plots are very "noisy", they both show an overall rising and falling characteristic, with pressure lagging flow as expected. It is unclear if the high frequency variations of both plots represent random measurement noise or whether there is a true relationship between flow and pressure at high frequencies. This same question arises for P1 data. For the black box modeling, noise in the pressure signals was discarded for the purpose of modeling the response to a clean step input. Alternatively, for the purpose of relating flow and pressure directly, different noise models could be attempted in an effort to determine the nature of any high frequency input-output relation.

Another striking feature of the data is the distinct shapes of the primary manipulated flow and the sum of all other flows, and the observation that these two signals change by similar magnitude. The recorded pressure fluctuation is a result of both influences, which may or may not have the same transfer function. The flow-to-pressure model developed below will help separate the effects of the primary flow from those of secondary flows.

3.5.2 Direct Relations Between Instantaneous Flow and Pressure. To investigate the features mentioned above, model forms different from those used previously must be investigated. Including a noise model complicates the identification process in a number of ways. Even assuming reasonably low orders for individual polynomials, the possible combinations of

polynomials forming equation (3.10) produce many model variations. Computation takes longer because there are more parameters to fit to the data and because the solution is iterative, unless the ARX form is used. The ARX form, however, limits the noise model denominator to equal the plant model denominator.

Considerable effort was expended to achieve any result for the white box model. A stable solution to the flow-pressure modeling was first found with the output error (OE) form, for which polynomials A(q), C(q), and D(q) in equation (3.10) equal one. This forms gives G(q)=B(q)/F(q) and H(q)=1 in Figure 3. Noise in this form is essentially modeled as random and occurring directly at the output. Later the Box-Jenkins (BJ) form, in which only A(q) is set to one, was attempted to determine if the noise containing some frequency weighting produces a better model. In this form G(q)=B(q)/F(q) and H(q)=C(q)/D(q). Though there may be other means of arriving at a model, the description that follows is the only procedure that was found to produce stable models for P1 and P2. Analysis of P1 is addressed first.

Data preparation for the white box modeling of P1 was similar to that applied to the pressure data of the black box modeling as described in Section 3.2.2, except that no pre-filtering was performed. Thus, data up to the flow step was averaged to achieve an initial value, an extension was added at the end of the data, and the initial value was subtracted from all data points of the data stream. This procedure was performed for both the input flows and output pressures. The two input flow signals consisted of the manipulated F2 flow (referred to as the "primary" flow and designated FP), and the sum of all

the other F2 flows and the inventory flows (the same "secondary" flow defined during black box modeling and designated FS). Figure 46 shows these flow signals for data set 5, and Figure 47 shows their sum. Data set 5 is one of the longer sets, having a negative F2 flow step at B0 of about 16 g/s.

Two types of flow-to-pressure model are attempted, one using a single input consisting of the sum of FP and FS (designated the "combined" flow model), and a second using an FP input and an FS input (designated the "separate" flow model). To achieve the two input separate flow OE model, it was necessary to provide initial parameter values to the Toolbox routines, otherwise the model result was unstable. This was accomplished by calculating the combined flow OE model, and then retrieving the resulting parameter values and supplying two sets of these values to the separate flow model for use as initial conditions. The only stable separated flow P1 model form was found to be

$$\begin{aligned} P1(t) = & \frac{b14 + b24 \cdot q^{-1}}{1 + f14 \cdot q^{-1} + f24 \cdot q^{-2}} \cdot FP(t-1) \\ & + \frac{b15 + b25 \cdot q^{-1}}{1 + f15 \cdot q^{-1} + f25 \cdot q^{-2}} \cdot FS(t-1) + e(t). \end{aligned} \tag{3.26}$$

The combined flow model form is similar, having the same polynomial orders but where only one input term exists.

Figures 48 and 49 show typical results for the combined flow and separate flow models. Here for the A0 location, the dotted lines represent the data and the solid lines are the model output based on that data. The smooth solid lines in the second half of the figures represent the response of the

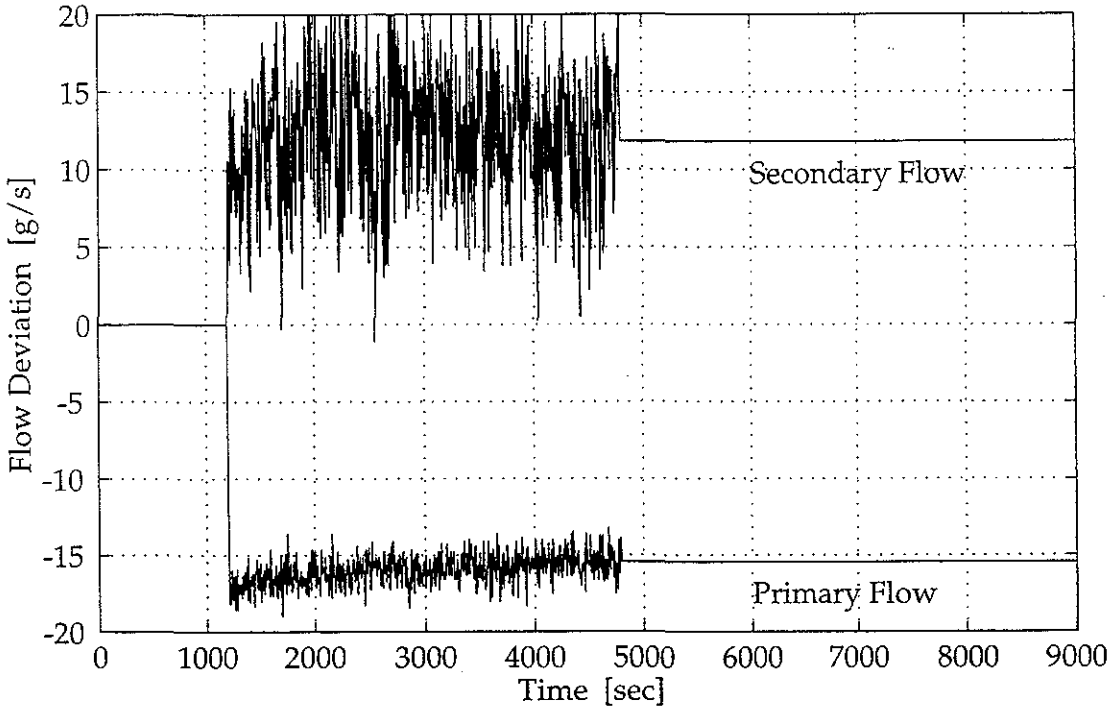

Figure 46. Primary Flow and Secondary Flow from Data Set 5

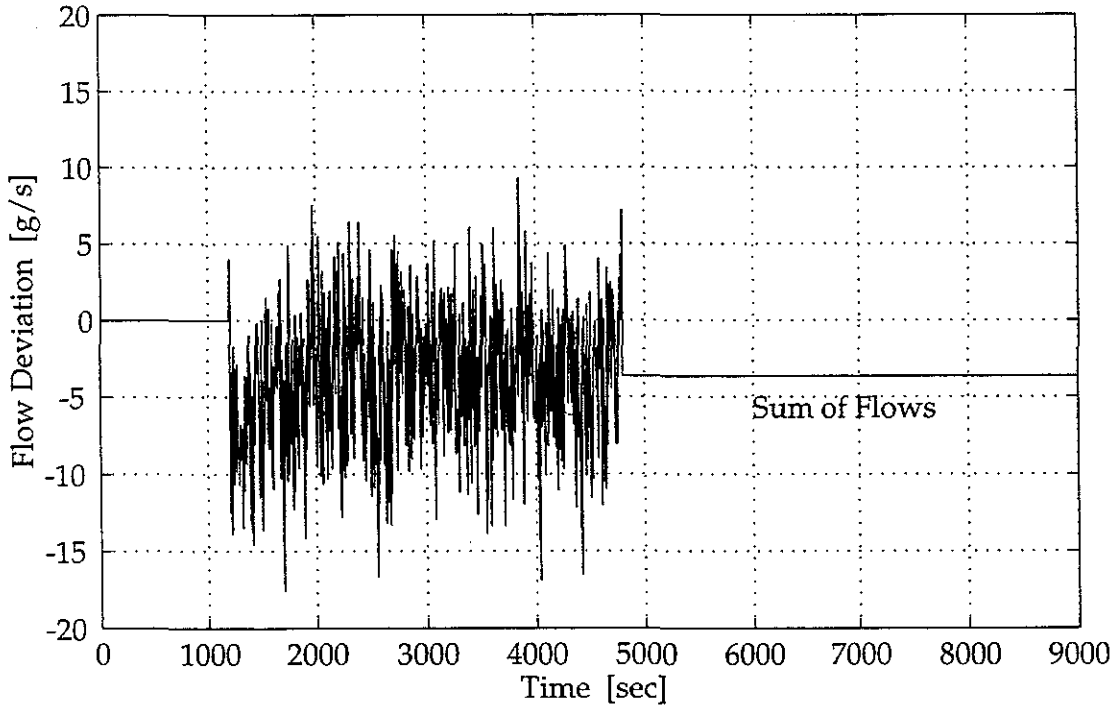

Figure 47. Sum of Primary Flow and Secondary Flow from Data Set 5

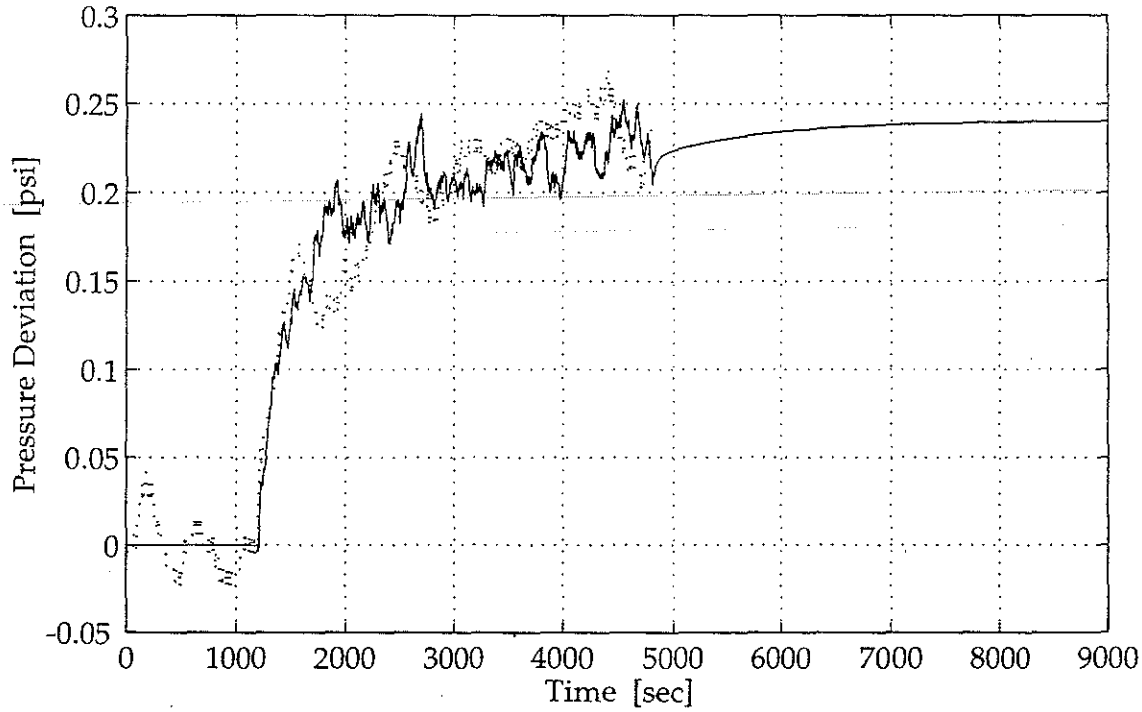

Figure 48. A0 P1 Data and P1 Combined Flow Model from Data Set 5

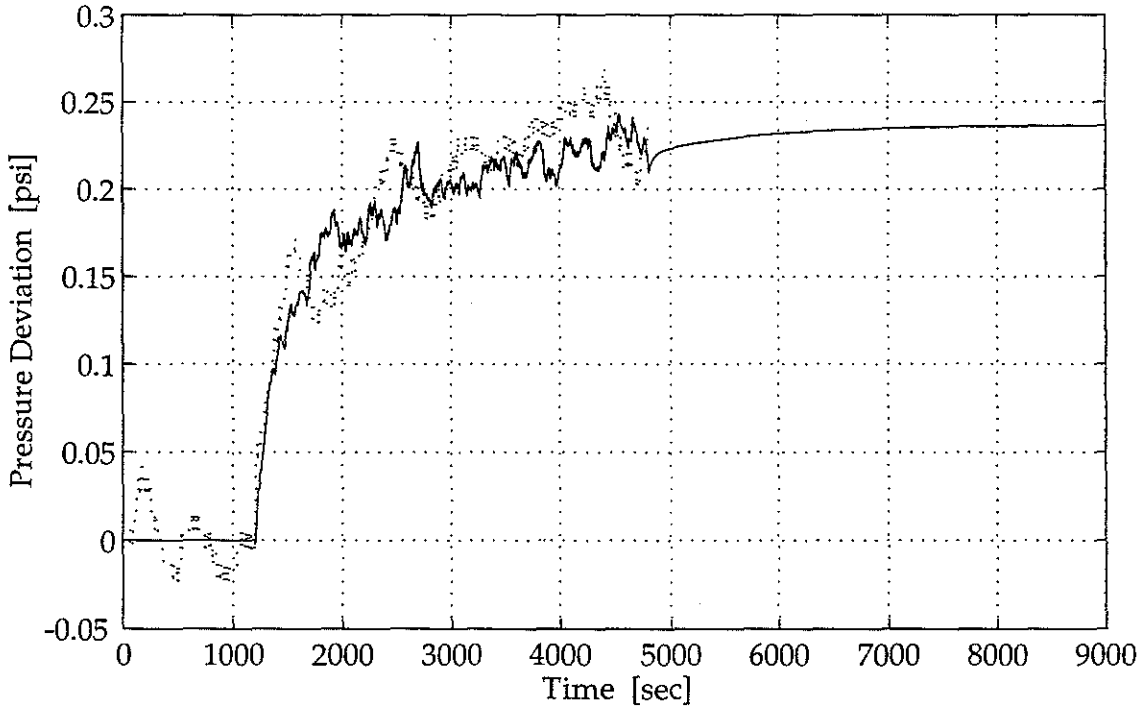

Figure 49. A0 P1 Data and P1 Separate Flow Model from Data Set 5

model to the smooth extended input data. When viewed in this way, the two methods give nearly identical outputs, with both following the course structure of the data, but with neither matching the fine structure of the pressure data very well. Choice of initial conditions is apparently so critical in this application of the Toolbox that the above procedure does not result in a realistic P1 model when working with the data of set 3. In the case of data set 3, the result was stable, but the pressure components tend to diverge well beyond 9000 seconds, and the sum of components does not represent P1 at many of the compressor location very well. The program "Micro1mod.m" included in the Appendix details the modeling steps outlined above. Attempts at a Box-Jenkins model form produced somewhat worse results as judged by the sum of squared error between data and model.

White box OE modeling of P2 occurred in a manner similar to that for P1, with the exception that acceptable initial parameter values were obtained only after calculating a single input ARX model. These initial conditions were, in turn, only suitable for use in the separate flow model. Curiously, the P2 modeling procedure was not successful for the case of data set 5, with stable but very poor model fits resulting. Instead, data set 3 was used because it also consisted of a long data stream with a flow step magnitude similar to data set 5, and it worked well. Input signals for this data set were shown previously in Figures 4 and 5. Details of the parameter initialization and other modeling steps are contained in the program "Micro2mod.m" in the Appendix. The stable separated flow P2 model form was found to be

$$P2(t) = \frac{b16 + b26 \cdot q^{-1} + b36 \cdot q^{-2}}{1 + f16 \cdot q^{-1} + f26 \cdot q^{-2} + f36 \cdot q^{-3}} \cdot FP(t-1) + \frac{b17 + b27 \cdot q^{-1} + b37 \cdot q^{-2}}{1 + f17 \cdot q^{-1} + f27 \cdot q^{-2} + f37 \cdot q^{-3}} \cdot FS(t-1) + e(t). \tag{3.27}$$

Figures 50 and 51 show typical results for the combined flow and separate flow model for P2. The combined model clearly does not produce good results, perhaps because of insufficient initial conditions. Since combined model results for all locations look similar to that for A0 in Figure 50, little further work was performed on this model. The separate flow model however, fits the overall P2 trend well. Just as for the case of P1, the fine details of the P2 data do not appear in the P2 model. Again, Box-Jenkins modeling produced slightly worse models based on the sum of squared errors.

3.5.3 Separating Response due to Two Flow Inputs. The separate effects of the two flow inputs on pressure outputs can be estimated for both combined and separate model types. In the combined flow model case, the primary flow is used as input in a single input combined flow model simulation, and the result plotted. Then, the calculation is performed for the secondary flow as input. For A0 P1 these two results are shown in Figure 52, along with the original combined model P1 response. In the separate flow case, two single input models must be formed, one using the primary flow model parameters, and one using the secondary flow model parameters. These two different models are each used to simulate a portion of the output response using their respective input flow data. The result for this breakout of the separate flow model into two P1 output components is shown in Figure 53 for the A0 location. Comparing Figures 52 and 53, observe that, though the

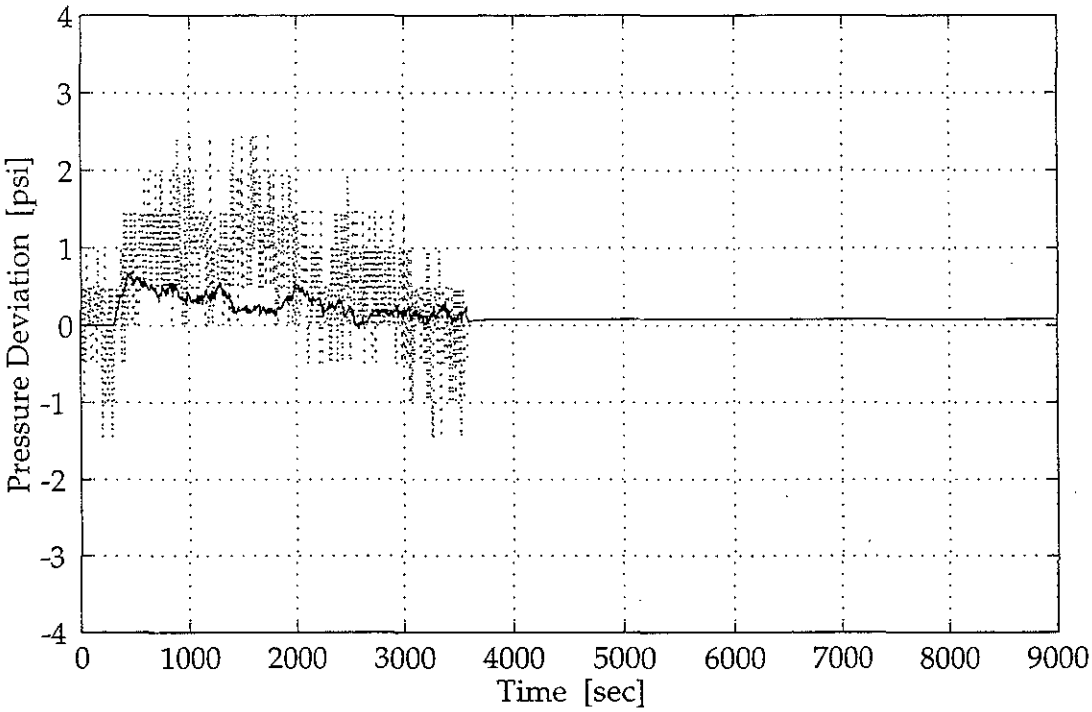

Figure 50. A0 P2 Data and P2 Combined Flow Model from Data Set 3

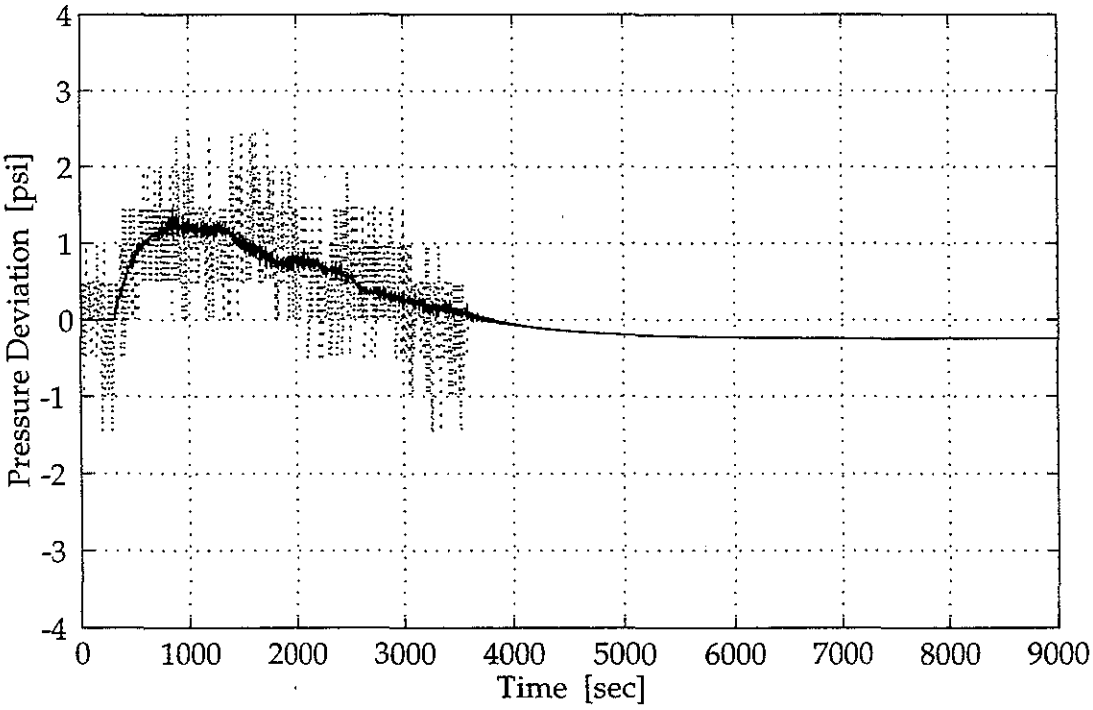

Figure 51. A0 P2 Data and P2 Separate Flow Model from Data Set 3

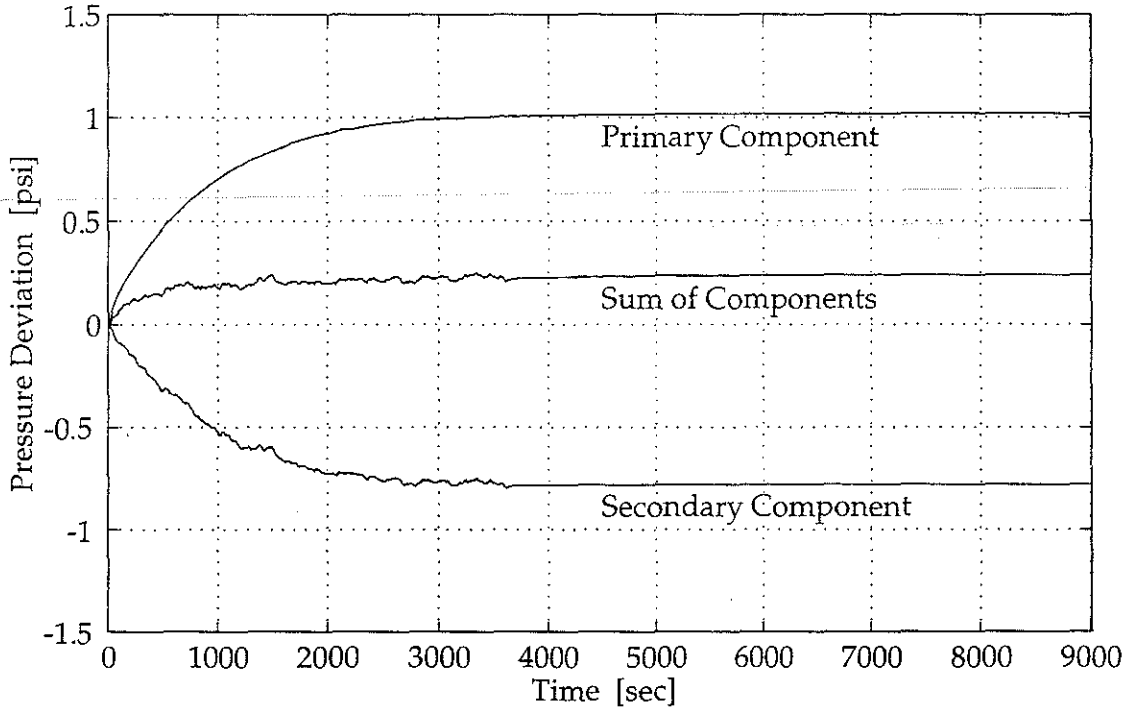

Figure 52. A0 P1 Combined Flow Model Components from Data Set 5

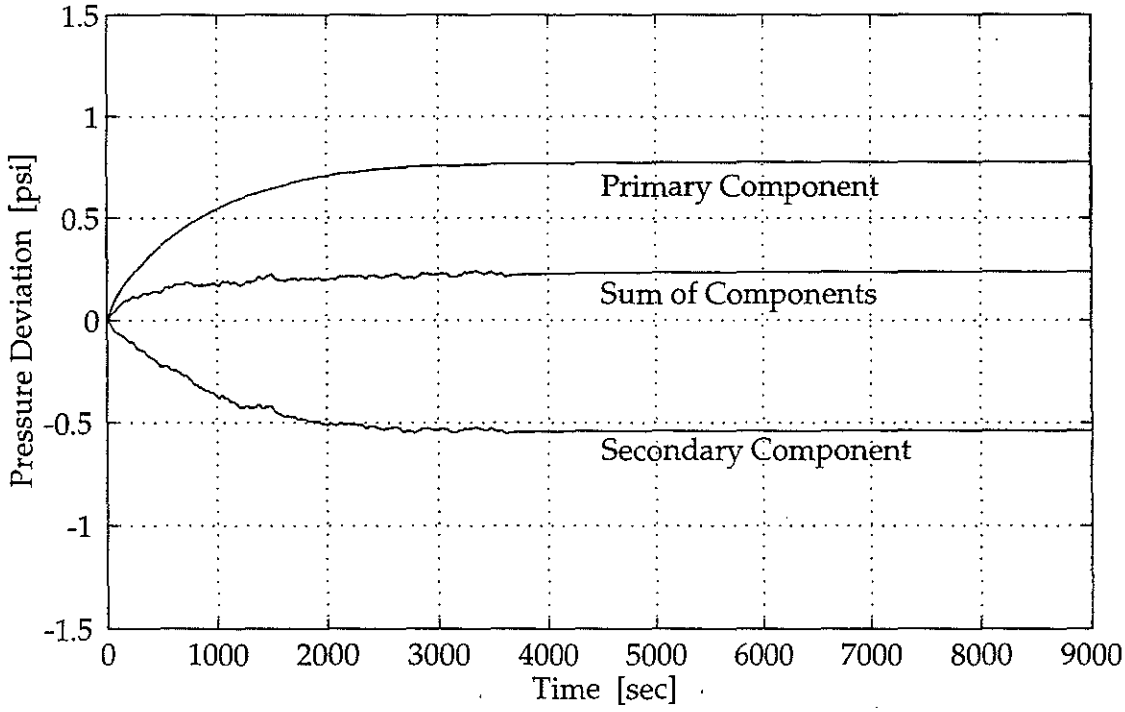

Figure 53. A0 P1 Separate Flow Model Components from Data Set 5

sum of primary and other components are nearly identical for the combined and separate models, like components are somewhat different in magnitude. The combined flow model produces primary and secondary components spaced a greater pressure difference apart than does the separate flow model.

Figures 54 and 55 provide typical results of the breakout into components for the P2 output. Discounting the combined flow models because the sum of components produces a poor fit, the separate flow P2 model output components give convincing evidence at least of the nature of the effect of P1 suction pressure feedback on the P2 discharge pressure. That is, if the primary flow acted alone, no fall in discharge pressure would occur after the initial rise. Figures 56 through 61 summarize the P1 and P2 output component results at all location as obtained for data sets 3 and 5. Note that no plots are given for the unsuccessful P2 combined flow model. No other data sets were analyzed following the white box methods outlined here because of the large amount of time required to do so, and because the results presented are sufficient to explain this previously unrecognized system behavior. Presumably with enough work, models for all data sets could be obtained, though the initial conditions used apparently must be very close to acceptable final model parameter values.

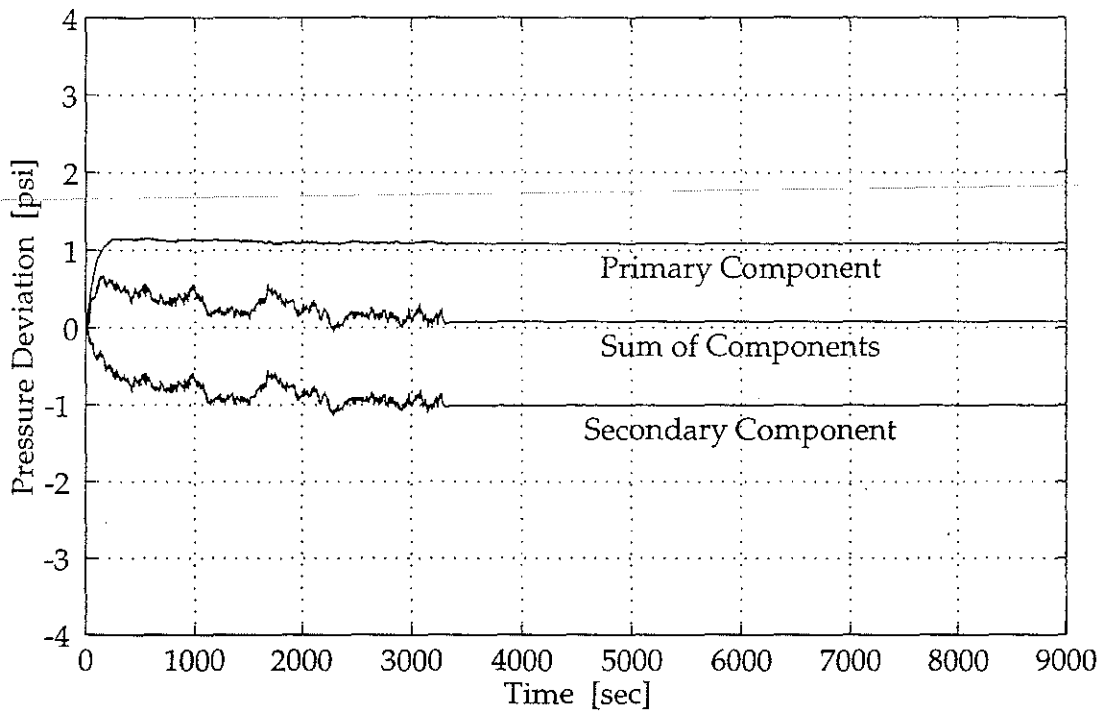

Figure 54. A0 P2 Combined Flow Model Components from Data Set 3

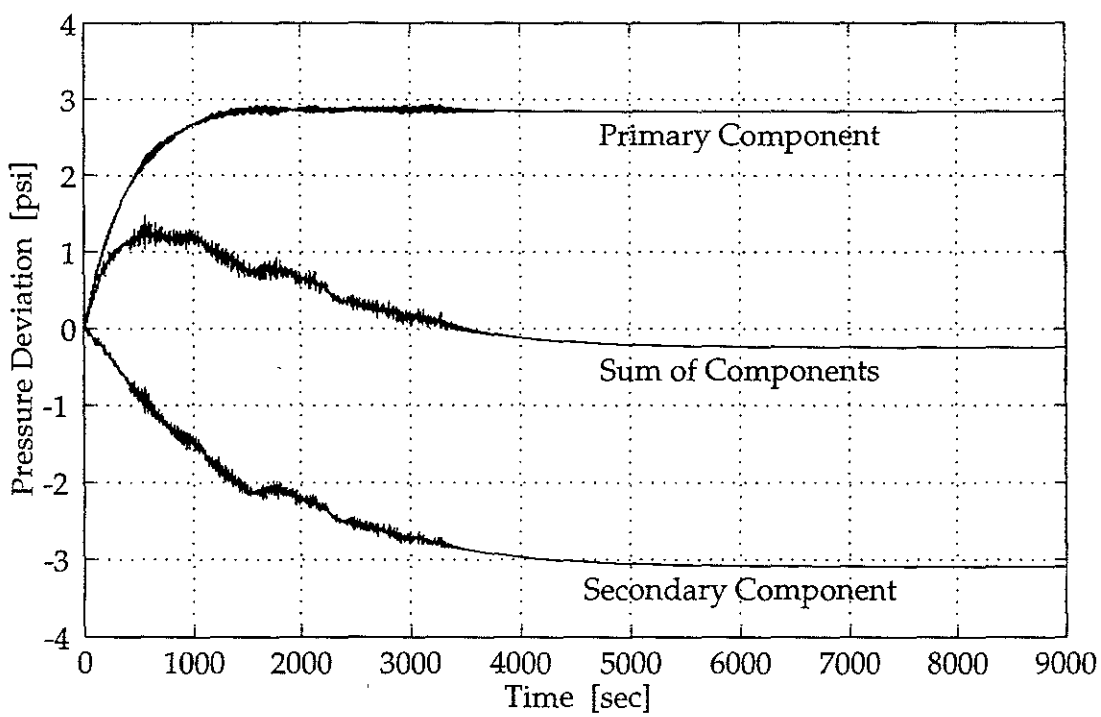

Figure 55. A0 P2 Separate Flow Model Components from Data Set 3

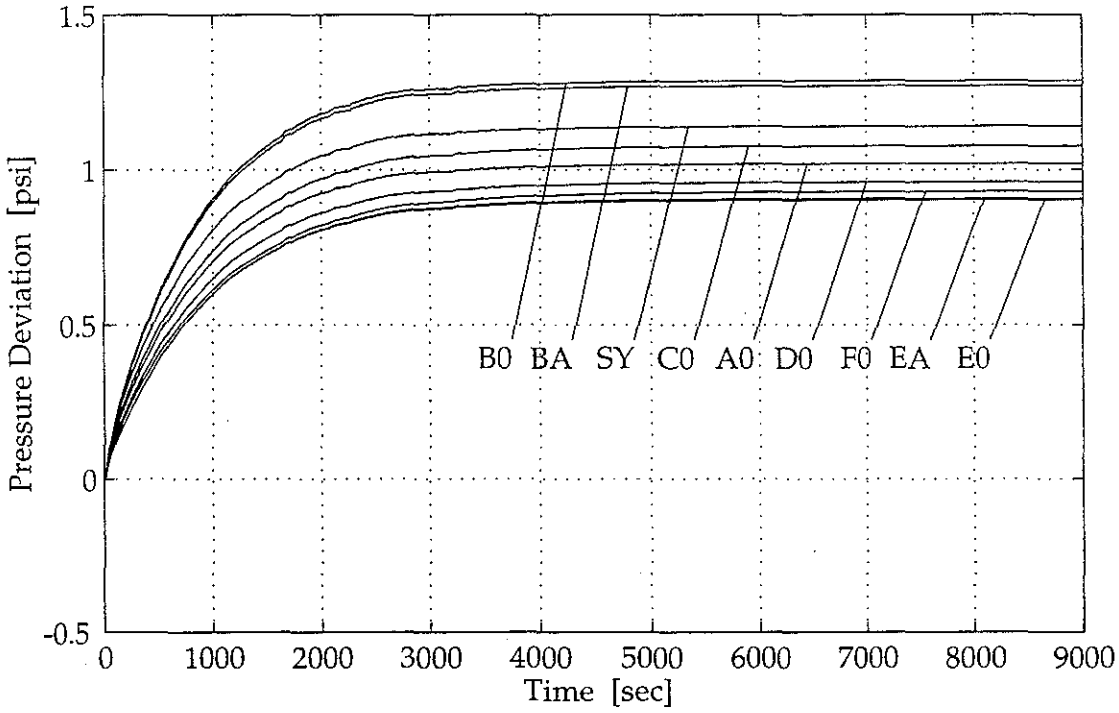

Figure 56. P1 Primary Components for Combined Model from Data Set 5

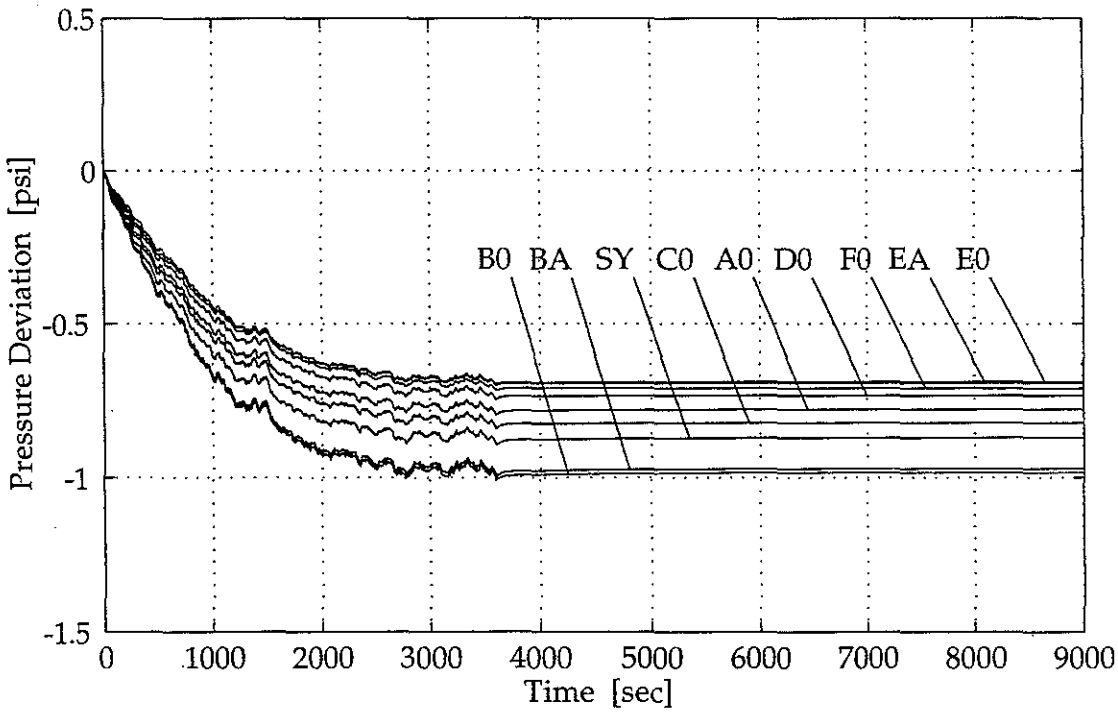

Figure 57. P1 Secondary Components for Combined Model from Data Set 5

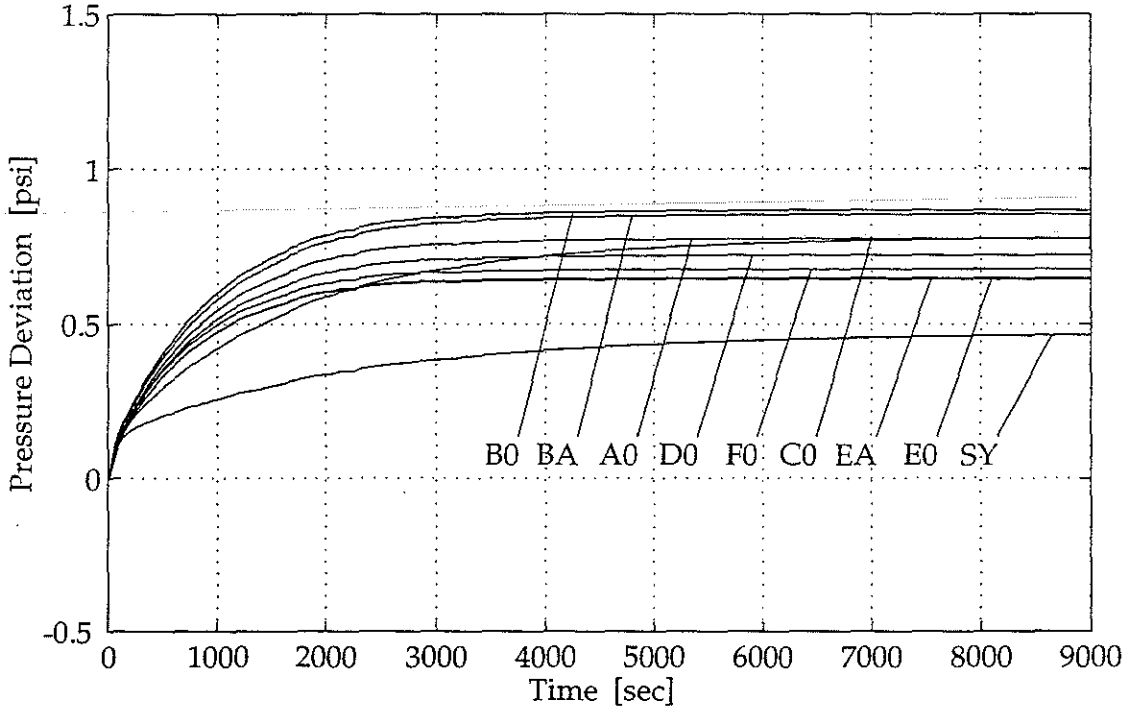

Figure 58. P1 Primary Components for Separate Model from Data Set 5

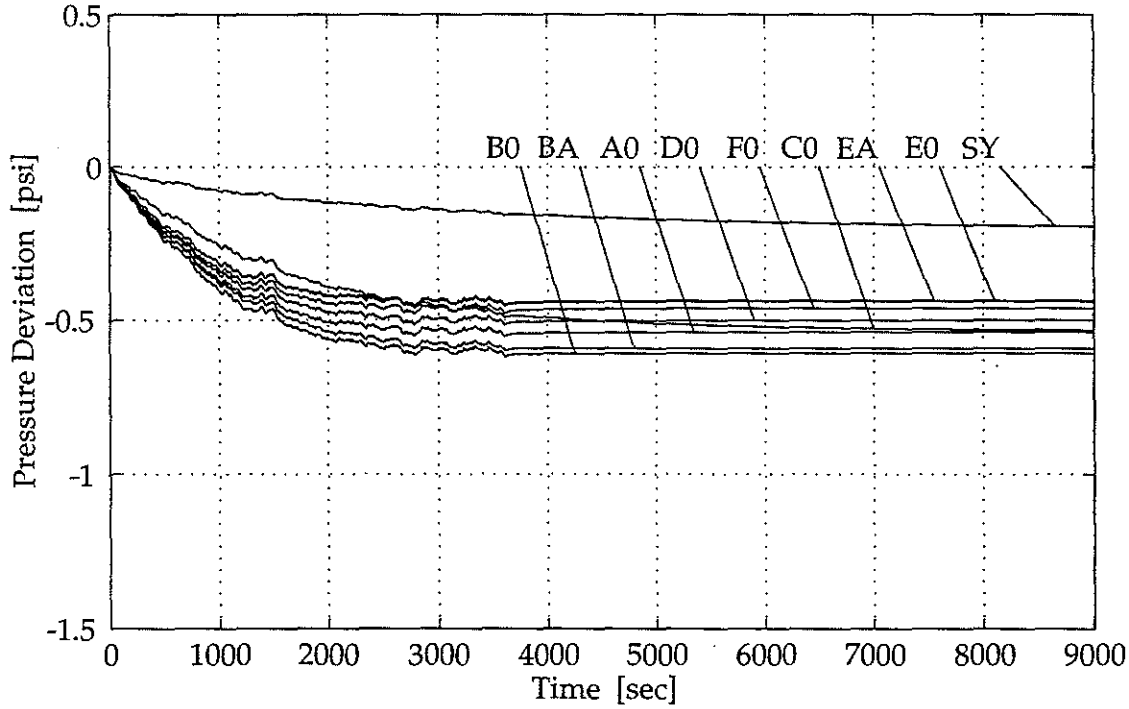

Figure 59. P1 Secondary Components for Separate Model from Data Set 5

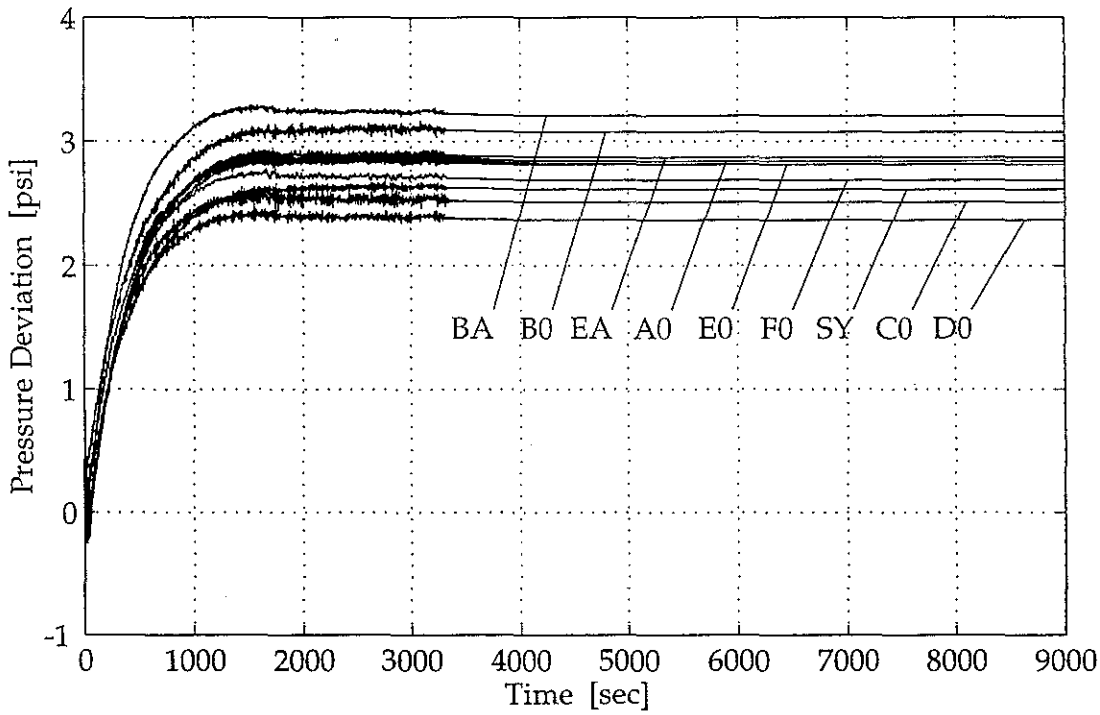

Figure 60. P2 Primary Components for Separate Model from Data Set 3

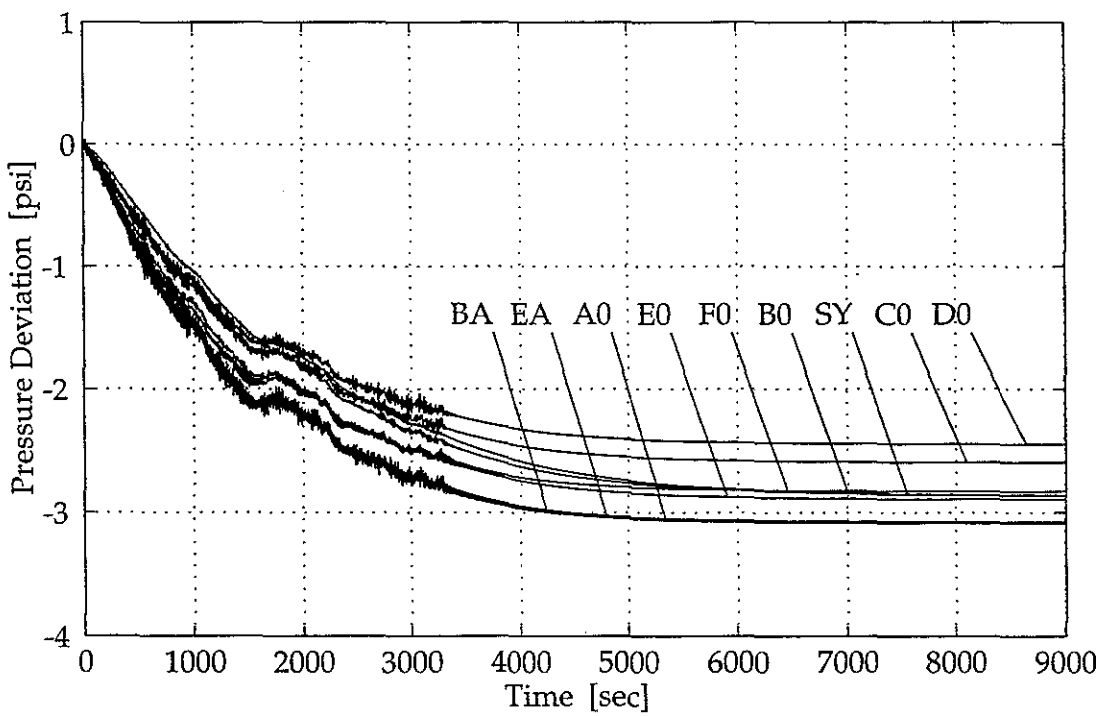

Figure 61. P2 Secondary Components for Separate Model from Data Set 3

CHAPTER IV

MODELING RESULTS AND DISCUSSION

Data sets 1 through 18 are fit to the model forms (3.21) and (3.22), and data sets 19 through 22 are fit to model form (3.23) per the procedure discussed in Chapter III. The summary results for each data set and the discussion presented here, qualitatively consider the variability of the models produced. Also, data sets 23 through 28 are used to validate averaged models and to demonstrate use of the models with input magnitudes different from those employed in creating the models. An outline for detailing the internal feedback observed within the system using the white box modeling method is presented. Characterizations of typical disturbances are given, and methods to improve system control are suggested based on the results of both the black box and white box models.

4.1 Black Box Results for Compressor Flow Input

In this section the results of modeling pressure response to 18 compressor flow step inputs are given. Responses corresponding to inputs at identical locations are then compared.

4.1.1 Parameters of Modeling. While the same MATLAB program (ModF2.m) is used to process each data set from the compressor flow input testing, a few parameters must be adjusted to accommodate differences between the sets. These modeling input parameters are summarized in Table 2. They include: location number (hs) and name, initial valve position (VPi), final valve position (VPf), atmospheric pressure (Patm), suction pressure

Table 2. Modeling Input Parameters and Sum of Squared Errors Values
(Page 1 of 7)

Parameter Name	Data Set Number 01	02	03	04
hs (house)	1.000 (A0)	1.000 (A0)	2.000 (B0)	2.000 (B0)
VPi (%)	1.0000e+02	9.2372e+01	9.2437e+01	1.0000e+02
VPf (%)	9.2460e+01	1.0000e+02	1.0000e+02	9.2393e+01
Patm (psia)	1.4291e+01	1.4291e+01	1.4557e+01	1.4548e+01
P1o (psig)	1.6414e+00	1.8466e+00	1.7000e+00	1.4265e+00
P2o (psig)	2.9140e+02	2.9067e+02	2.9390e+02	2.9033e+02
ko (-)	6.1000e+01	6.3000e+01	6.0000e+01	2.3000e+01
kn (-)	6.0000e+02	7.2000e+02	7.2000e+02	5.4000e+02
ki1 (-)	5.0000e+02	6.0000e+02	6.0000e+02	3.0000e+02
kf1 (-)	6.0000e+02	7.2000e+02	7.2000e+02	5.4000e+02
ki2 (-)	5.6000e+02	6.9000e+02	6.8000e+02	5.2000e+02
kf2 (-)	6.0000e+02	7.2000e+02	7.2000e+02	5.4000e+02
step (g/s)	-1.5659e+01	1.5915e+01	1.5992e+01	-1.5741e+01
A0 P1 sse	3.6377e-01	8.1565e-01	1.1651e-01	1.7281e-01
B0 P1 sse	3.1460e-01	6.1571e-01	1.7217e-01	2.9013e-01
BA P1 sse	3.2871e-01	6.1946e-01	1.5089e-01	2.7278e-01
C0 P1 sse	1.3504e-01	4.5141e-01	1.1510e-01	2.1997e-01
D0 P1 sse	1.2232e-01	4.0480e-01	1.0341e-01	2.1863e-01
E0 P1 sse	1.2755e-01	4.5406e-01	9.3459e-02	2.0173e-01
EA P1 sse	1.3357e-01	4.4059e-01	1.0277e-01	2.0400e-01
F0 P1 sse	2.2434e-01	6.1922e-01	1.8926e-01	2.1637e-01
SY P1 sse	3.2534e-01	6.8872e-01	1.3231e-01	2.1930e-01
A0 P2 sse	2.2234e+02	2.9983e+02	2.3775e+02	1.9567e+02
B0 P2 sse	4.7069e+01	1.0274e+02	9.6746e+01	6.7335e+01
BA P2 sse	3.9025e+01	1.0137e+02	5.8438e+01	5.2210e+01
C0 P2 sse	7.4821e+01	1.4521e+02	1.1128e+02	8.4683e+01
D0 P2 sse	6.1066e+01	1.1481e+02	8.6570e+01	6.5253e+01
E0 P2 sse	4.8550e+01	1.0778e+02	6.7492e+01	5.5003e+01
EA P2 sse	4.2496e+01	1.0230e+02	6.3646e+01	5.1416e+01
F0 P2 sse	7.7201e+01	1.6351e+02	9.9326e+01	8.6855e+01
SY P2 sse	5.3781e+01	1.1324e+02	7.3531e+01	6.0011e+01

Table 2. (Page 2 of 7)

Parameter Name	Data Set Number			
	05	06	07	08
hs (house)	2.000 (B0)	2.000 (B0)	4.000 (C0)	4.000 (C0)
VPi (%)	1.0000e+02	9.8050e+01	1.0000e+02	9.2505e+01
VPf (%)	9.2300e+01	1.0000e+02	9.4420e+01	1.0000e+02
Patm (psia)	1.4537e+01	1.4292e+01	1.4280e+01	1.4270e+01
P1o (psig)	1.4670e+00	1.5630e+00	1.4070e+00	1.6410e+00
P2o (psig)	2.9427e+02	2.9470e+02	2.9593e+02	2.9548e+02
ko (-)	2.4100e+02	1.3700e+02	1.2500e+02	1.2000e+02
kn (-)	9.6000e+02	6.6000e+02	8.4000e+02	8.0000e+02
ki1 (-)	8.0000e+02	4.6000e+02	7.0000e+02	7.0000e+02
kf1 (-)	9.6000e+02	6.6000e+02	8.4000e+02	8.0000e+02
ki2 (-)	9.4000e+02	6.3000e+02	8.0000e+02	7.6000e+02
kf2 (-)	9.6000e+02	6.6000e+02	8.4000e+02	8.0000e+02
step (g/s)	-1.5827e+01	1.0445e+01	-1.4079e+01	1.5610e+01
A0 P1 sse	3.9376e-01	2.1052e-01	3.7659e-01	2.5236e-01
B0 P1 sse	7.2269e-01	2.4910e-01	7.0351e-01	3.3174e-01
BA P1 sse	6.5943e-01	2.3118e-01	7.1341e-01	3.1931e-01
C0 P1 sse	5.0962e-01	2.0215e-01	1.2358e+00	4.8611e-01
D0 P1 sse	4.0872e-01	1.8692e-01	6.8964e-01	4.6863e-01
E0 P1 sse	3.5810e-01	1.7547e-01	4.5639e-01	5.0242e-01
EA P1 sse	3.5134e-01	1.8557e-01	4.6567e-01	5.1338e-01
F0 P1 sse	4.3499e-01	2.5472e-01	4.2752e-01	3.5112e-01
SY P1 sse	5.0453e-01	2.1773e-01	5.2816e-01	2.8546e-01
A0 P2 sse	2.8432e+02	2.1236e+02	2.3232e+02	2.1518e+02
B0 P2 sse	1.2819e+02	6.9332e+01	6.8829e+01	6.9006e+01
BA P2 sse	9.7653e+01	6.1764e+01	6.0434e+01	6.9165e+01
C0 P2 sse	1.4638e+02	9.4616e+01	1.0772e+02	1.3930e+02
D0 P2 sse	1.0882e+02	7.7787e+01	8.7222e+01	8.8270e+01
E0 P2 sse	9.1839e+01	6.0545e+01	5.8761e+01	7.0071e+01
EA P2 sse	9.2635e+01	6.6605e+01	6.3742e+01	8.7827e+01
F0 P2 sse	1.3746e+02	9.9913e+01	1.0909e+02	1.1467e+02
SY P2 sse	1.0814e+02	6.8681e+01	6.8535e+01	7.5809e+01

Table 2. (Page 3 of 7)

Parameter Name	Data Set Number			
	09	10	11	12
hs (house)	5.000 (D0)	5.000 (D0)	5.000 (D0)	5.000 (D0)
VPi (%)	1.0000e+02	1.0000e+02	9.6105e+01	9.2357e+01
VPf (%)	9.2090e+01	9.6040e+01	1.0000e+02	1.0000e+02
Patm (psia)	1.4557e+01	1.4557e+01	1.4557e+01	1.4530e+01
P1o (psig)	1.3972e+00	1.3972e+00	1.5350e+00	1.6317e+00
P2o (psig)	2.9616e+02	2.9472e+02	2.9419e+02	2.8929e+02
ko (-)	6.1000e+01	5.7000e+01	5.7000e+01	5.3000e+01
kn (-)	5.4000e+02	4.8000e+02	5.4000e+02	6.0000e+02
ki1 (-)	4.4000e+02	2.8000e+02	4.4000e+02	4.6000e+02
kf1 (-)	5.4000e+02	4.8000e+02	5.4000e+02	6.0000e+02
ki2 (-)	5.3500e+02	3.8000e+02	4.6000e+02	5.6000e+02
kf2 (-)	5.4000e+02	4.8000e+02	5.4000e+02	6.0000e+02
step (g/s)	-1.5903e+01	-1.2926e+01	1.2975e+01	1.5948e+01
A0 P1 sse	7.8249e-02	2.8974e-01	9.4939e-02	1.3911e-01
B0 P1 sse	1.0433e-01	2.6887e-01	1.0868e-01	1.6208e-01
BA P1 sse	1.1723e-01	2.7089e-01	1.0459e-01	1.6390e-01
C0 P1 sse	1.1149e-01	3.0298e-01	1.2988e-01	2.6860e-01
D0 P1 sse	1.0321e-01	4.1636e-01	1.8774e-01	3.0521e-01
E0 P1 sse	7.9682e-02	3.1241e-01	1.5008e-01	1.6712e-01
EA P1 sse	8.7995e-02	3.2674e-01	1.3445e-01	1.7557e-01
F0 P1 sse	1.2755e-01	3.1346e-01	2.7644e-01	2.7882e-01
SY P1 sse	9.9799e-02	2.8124e-01	1.0386e-01	1.5530e-01
A0 P2 sse	1.2857e+02	1.9132e+02	2.4004e+02	2.2580e+02
B0 P2 sse	4.1112e+01	8.7646e+01	1.0341e+02	6.4724e+01
BA P2 sse	3.3155e+01	8.6118e+01	9.6903e+01	5.7544e+01
C0 P2 sse	6.2638e+01	1.1001e+02	1.3902e+02	8.6752e+01
D0 P2 sse	5.7660e+01	1.1129e+02	1.3854e+02	8.4123e+01
E0 P2 sse	4.2228e+01	8.4563e+01	1.1366e+02	6.8690e+01
EA P2 sse	3.9604e+01	9.0787e+01	1.0437e+02	7.0044e+01
F0 P2 sse	7.2933e+01	1.2047e+02	1.3990e+02	9.1094e+01
SY P2 sse	4.2188e+01	9.4042e+01	1.1050e+02	6.8426e+01

Table 2. (Page 4 of 7)

Parameter Name	Data Set Number			
	13	14	15	16
hs (house)	6.000 (E0)	6.000 (E0)	8.000 (F0)	8.000 (F0)
VPi (%)	9.1060e+01	1.0000e+02	1.0000e+02	9.6090e+01
VPf (%)	1.0000e+02	9.2237e+01	9.2124e+01	1.0000e+02
Patm (psia)	1.4274e+01	1.4274e+01	1.4557e+01	1.4565e+01
P1o (psig)	1.2408e+00	1.0450e+00	1.3776e+00	1.4753e+00
P2o (psig)	2.9356e+02	2.9410e+02	2.9186e+02	2.9029e+02
ko (-)	4.7000e+01	3.4000e+01	1.3400e+02	1.8000e+01
kn (-)	5.0000e+02	5.2000e+02	6.0000e+02	4.8000e+02
ki1 (-)	3.0000e+02	2.2000e+02	4.4000e+02	4.0000e+02
kf1 (-)	5.0000e+02	5.2000e+02	6.0000e+02	4.8000e+02
ki2 (-)	4.6000e+02	5.1000e+02	5.0000e+02	4.4000e+02
kf2 (-)	5.0000e+02	5.2000e+02	6.0000e+02	4.8000e+02
step (g/s)	1.6019e+01	-1.5186e+01	-1.5863e+01	1.2948e+01
A0 P1 sse	2.4488e-01	2.0321e-01	2.0041e-01	8.0800e-02
B0 P1 sse	2.9391e-01	2.2420e-01	1.9008e-01	7.1033e-02
BA P1 sse	3.0452e-01	2.2992e-01	1.8572e-01	6.7268e-02
C0 P1 sse	2.9380e-01	2.7153e-01	2.3922e-01	8.3813e-02
D0 P1 sse	3.1490e-01	3.1766e-01	2.3750e-01	7.7971e-02
E0 P1 sse	4.2731e-01	4.8193e-01	2.2399e-01	7.5543e-02
EA P1 sse	4.3815e-01	4.5063e-01	2.1847e-01	7.8809e-02
F0 P1 sse	2.6382e-01	3.0039e-01	4.5193e-01	1.9118e-01
SY P1 sse	2.5877e-01	2.1216e-01	1.9414e-01	7.4796e-02
A0 P2 sse	1.7404e+02	2.0046e+02	1.6377e+02	1.6667e+02
B0 P2 sse	5.1925e+01	5.7936e+01	5.1860e+01	5.7955e+01
BA P2 sse	4.2426e+01	4.7166e+01	3.5990e+01	5.4886e+01
C0 P2 sse	6.9897e+01	7.4200e+01	7.3417e+01	7.6972e+01
D0 P2 sse	6.4120e+01	7.3441e+01	5.0683e+01	6.3919e+01
E0 P2 sse	5.0826e+01	5.8036e+01	4.2837e+01	5.6239e+01
EA P2 sse	4.9319e+01	6.1078e+01	4.2964e+01	5.2010e+01
F0 P2 sse	7.3606e+01	8.2338e+01	7.2907e+01	7.7324e+01
SY P2 sse	4.8021e+01	6.3525e+01	3.8926e+01	5.8728e+01

Table 2. (Page 5 of 7)

Parameter Name	Data Set Number			
	17	18	19	20
hs (house)	8.000 (F0)	8.000 (F0)	10.00 (IN)	10.00 (IN)
VPi (%)	1.0000e+02	1.0000e+02	5.2701e+01	5.7115e+01
VPf (%)	9.6206e+01	9.2296e+01	5.7117e+01	5.9291e+01
Patm (psia)	1.4557e+01	1.4503e+01	1.4424e+01	1.4380e+01
P1o (psig)	1.3190e+00	1.3874e+00	1.6439e+00	1.6875e+00
P2o (psig)	2.9207e+02	2.8749e+02	3.0775e+02	2.9311e+02
ko (-)	4.0000e+01	9.0000e+01	1.1500e+02	6.3000e+01
kn (-)	4.8000e+02	6.0000e+02	1.5600e+03	8.4000e+02
ki1 (-)	3.0000e+02	5.0000e+02	n/a	n/a
kf1 (-)	4.8000e+02	6.0000e+02	n/a	n/a
ki2 (-)	4.0000e+02	5.0000e+02	n/a	n/a
kf2 (-)	4.8000e+02	6.0000e+02	n/a	n/a
step (g/s)	-1.2705e+01	-1.5717e+01	8.9284e+00	4.6722e+00
A0 P1 sse	1.1917e-01	3.5996e-01	n/a	n/a
B0 P1 sse	1.1001e-01	2.8093e-01	n/a	n/a
BA P1 sse	1.0145e-01	2.7813e-01	n/a	n/a
C0 P1 sse	1.3157e-01	2.9370e-01	n/a	n/a
D0 P1 sse	1.4572e-01	3.3067e-01	n/a	n/a
E0 P1 sse	1.3454e-01	3.4536e-01	n/a	n/a
EA P1 sse	1.4299e-01	3.5844e-01	n/a	n/a
F0 P1 sse	2.2397e-01	5.3486e-01	n/a	n/a
SY P1 sse	1.1696e-01	3.1771e-01	n/a	n/a
A0 P2 sse	2.2874e+02	1.7110e+02	5.8864e+02	3.3343e+02
B0 P2 sse	1.0962e+02	5.5747e+01	2.3544e+02	1.4504e+02
BA P2 sse	1.0298e+02	4.2645e+01	2.2816e+02	1.3511e+02
C0 P2 sse	1.2959e+02	4.7227e+01	3.3178e+02	1.9694e+02
D0 P2 sse	1.3342e+02	7.5185e+01	2.7521e+02	1.7868e+02
E0 P2 sse	1.0962e+02	5.9059e+01	2.4470e+02	1.3555e+02
EA P2 sse	1.1708e+02	5.9836e+01	2.3306e+02	1.4110e+02
F0 P2 sse	1.5053e+02	9.0771e+01	3.3133e+02	1.7844e+02
SY P2 sse	1.1064e+02	6.0224e+01	2.3562e+02	1.4970e+02

Table 2. (Page 6 of 7)

Parameter Name	Data Set Number			
	21	22	23	24
hs (house)	10.00 (IN)	10.00 (IN)	2.000 (B0)	6.000 (E0)
VPi (%)	5.6951e+01	6.1193e+01	9.2278e+01	0.0000e+00
VPf (%)	6.1143e+01	5.7092e+01	1.0000e+02	1.0000e+02
Patm (psia)	1.4380e+01	1.4380e+01	1.4548e+01	1.4300e+01
P1o (psig)	1.6598e+00	1.6207e+00	1.5930e+00	1.5828e+00
P2o (psig)	2.9213e+02	2.7628e+02	2.8955e+02	2.9136e+02
ko (-)	5.9000e+01	7.2000e+01	6.1000e+01	2.0900e+02
kn (-)	1.3200e+03	1.3200e+03	6.0000e+02	1.0800e+03
ki1 (-)	n/a	n/a	n/a	n/a
kf1 (-)	n/a	n/a	n/a	n/a
ki2 (-)	n/a	n/a	n/a	n/a
kf2 (-)	n/a	n/a	n/a	n/a
step (g/s)	9.3818e+00	-8.7032e+00	1.5980e+01	5.4730e+01
A0 P1 sse	n/a	n/a	5.9108e-01	8.4639e+00
B0 P1 sse	n/a	n/a	8.7155e-01	8.2867e+00
BA P1 sse	n/a	n/a	8.7863e-01	8.0191e+00
C0 P1 sse	n/a	n/a	7.4510e-01	5.5900e+00
D0 P1 sse	n/a	n/a	7.6290e-01	6.5205e+00
E0 P1 sse	n/a	n/a	6.9311e-01	9.7224e+00
EA P1 sse	n/a	n/a	7.0641e-01	1.0241e+01
F0 P1 sse	n/a	n/a	1.0335e+00	9.4466e+00
SY P1 sse	n/a	n/a	7.0435e-01	8.4149e+00
A0 P2 sse	4.0807e+02	5.0160e+02	2.2622e+02	1.2312e+03
B0 P2 sse	1.2343e+02	2.1543e+02	1.1792e+02	4.5228e+02
BA P2 sse	1.0305e+02	1.9835e+02	1.3199e+02	5.2986e+02
C0 P2 sse	1.9952e+02	3.0539e+02	1.3139e+02	4.7486e+02
D0 P2 sse	1.3834e+02	2.4719e+02	1.7115e+02	4.6212e+02
E0 P2 sse	1.2502e+02	2.3219e+02	1.4571e+02	6.3801e+02
EA P2 sse	1.1913e+02	2.1594e+02	1.1736e+02	6.4900e+02
F0 P2 sse	2.0600e+02	3.0992e+02	1.4147e+02	1.0141e+03
SY P2 sse	1.2424e+02	2.2304e+02	9.7930e+01	4.4925e+02

Table 2. (Page 7 of 7)

Parameter Name	Data Set Number			
	25	26	27	28
hs (house)	8.000 (F0)	8.000 (F0)	10.00 (IN)	10.00 (IN)
VPi (%)	1.0000e+02	90.90/85.64	5.2747e+01	5.9335e+01
VPf (%)	0.0000e+00	100.0/100.0	6.0635e+01	5.6997e+01
Patm (psia)	1.4468e+01	1.4459e+01	1.4318e+01	1.4438e+01
P1o (psig)	1.3776e+00	2.0518e+00	1.4943e+00	1.6881e+00
P2o (psig)	2.8879e+02	2.8782e+02	3.0093e+02	2.8525e+02
ko (-)	1.2100e+02	3.1000e+02	1.2400e+02	6.8000e+01
kn (-)	1.4400e+03	1.8000e+03	7.2000e+02	8.4000e+02
ki1 (-)	n/a	n/a	n/a	n/a
kf1 (-)	n/a	n/a	n/a	n/a
ki2 (-)	n/a	n/a	n/a	n/a
kf2 (-)	n/a	n/a	n/a	n/a
step (g/s)	-5.4610e+01	3.6440e+01	1.6232e+01	-4.6852e+00
A0 P1 sse	1.0762e+01	2.4738e+00	n/a	n/a
B0 P1 sse	5.4977e+00	1.9999e+00	n/a	n/a
BA P1 sse	5.7043e+00	2.0600e+00	n/a	n/a
C0 P1 sse	1.0221e+01	2.8026e+00	n/a	n/a
D0 P1 sse	8.9437e+00	2.4517e+00	n/a	n/a
E0 P1 sse	8.1232e+00	2.1351e+00	n/a	n/a
EA P1 sse	8.3707e+00	2.2296e+00	n/a	n/a
F0 P1 sse	6.7810e+00	1.9326e+00	n/a	n/a
SY P1 sse	5.9355e+00	1.9078e+00	n/a	n/a
A0 P2 sse	4.8920e+03	1.6491e+03	2.4460e+03	3.2399e+02
B0 P2 sse	4.7056e+03	1.6704e+03	2.7595e+03	1.1165e+02
BA P2 sse	4.4144e+03	1.6387e+03	2.5166e+03	9.0318e+01
C0 P2 sse	5.8863e+03	2.2757e+03	2.6378e+03	1.4194e+02
D0 P2 sse	4.8668e+03	1.9081e+03	2.4073e+03	1.2130e+02
E0 P2 sse	6.2445e+03	2.1444e+03	2.5075e+03	1.0934e+02
EA P2 sse	4.8087e+03	1.7224e+03	2.4285e+03	1.0430e+02
F0 P2 sse	4.4509e+03	1.4092e+03	2.4015e+03	1.6529e+02
SY P2 sse	5.6413e+03	2.0306e+03	2.4695e+03	1.0981e+02

prior to the valve step (P1o), discharge pressure prior to the valve step (P2o), sample number at start of valve step (ko), final sample number (kn), initial and final sample number of the data values on which P1 data extension is based (ki1 and kf1), initial and final sample number of data the values on which P2 data extension in based (ki2 and kf2), and calculated flow change (step).

Table 2 also gives the sum of the squared errors (sse) for each model. This value represents the sum of all the squares of the differences between model output and unfiltered data at each data point. The portion of the model covering the extended data is not included in the sum. Without showing plots of all data and model simulations (which would require inclusion of over 300 additional figures), the sse give a numerical value to the goodness of fit between the model and the data it was created from. To put the numbers in perspective, note that sse values for data set 3 (investigated in detail in Chapter III) are on the same order as sse values for other data sets in the group 1 through 18. One can expect, therefore, that the model fit will be very similar to that shown in Figures 6 through 23 for data set 3.

4.1.2 Comparison of Results at Individual Locations. Simulations using the models created for each of the 18 compressor flow data sets are shown in Figures 62 through 97. To bring all simulations to an equal basis, a unit step of 1 g/s is used as the compressor flow input. These figures are arranged in the same order as the columns of Table 2, and thus appear in location order. Each pressure curve in a figure is labeled from left to right starting with the output location having the greatest response and ending with the output location

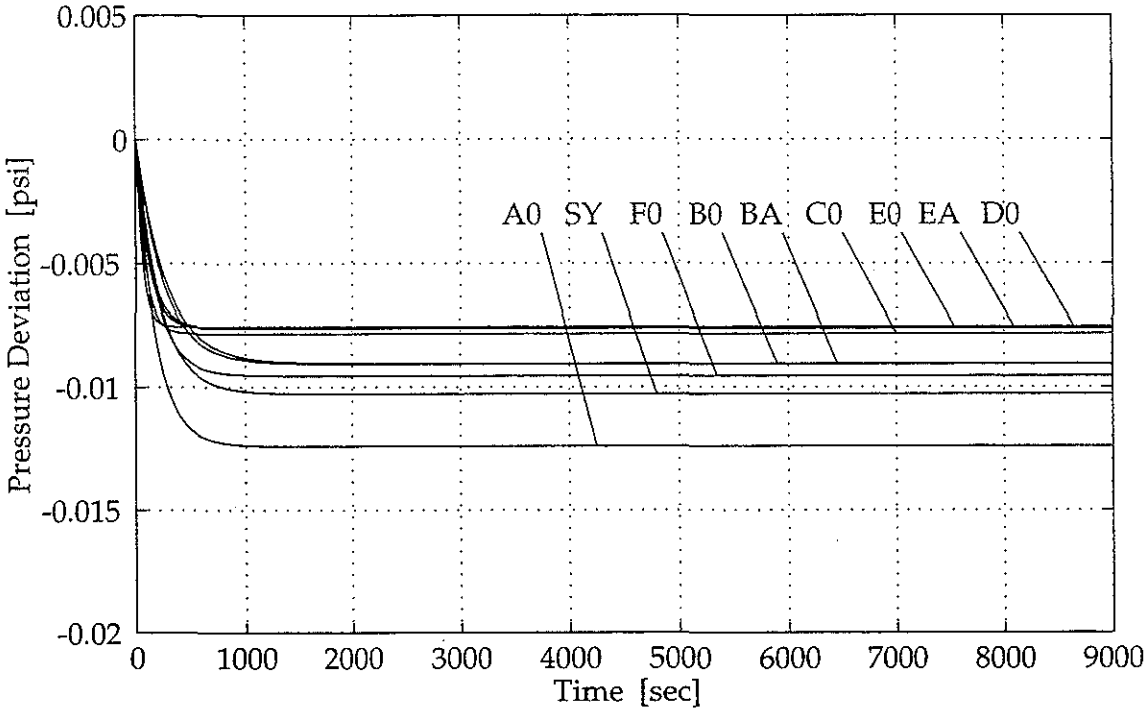

Figure 62. P1 Model for Unit Step in A0 F2 based on Data Set 1

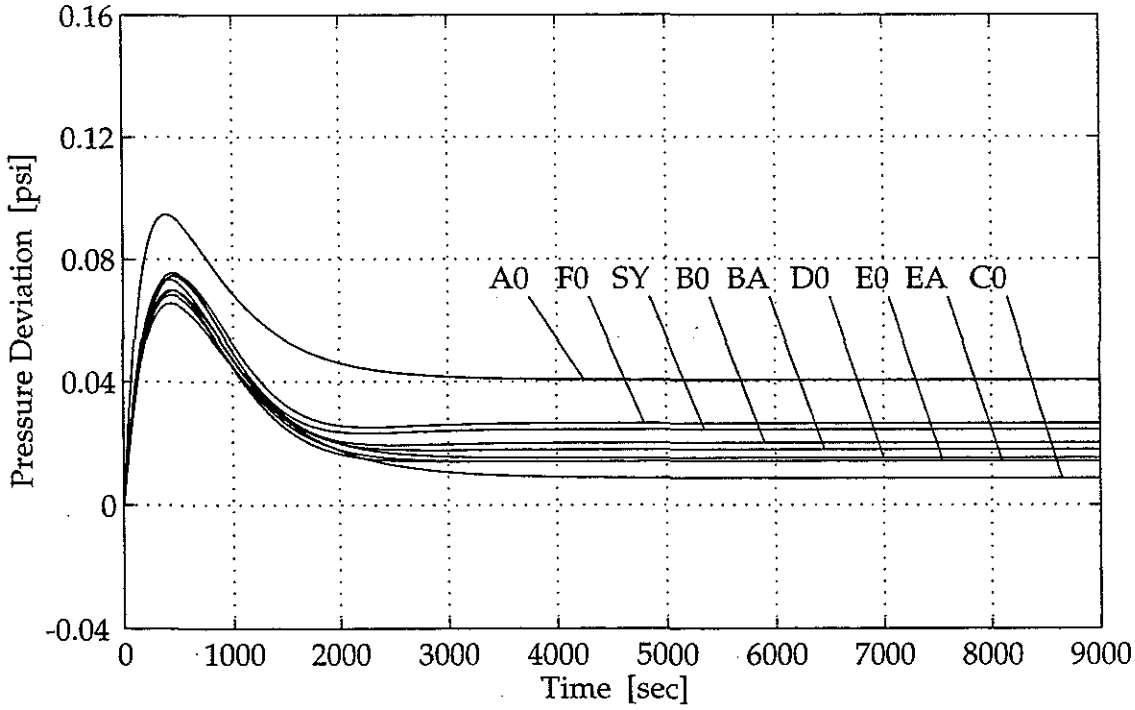

Figure 63. P2 Model for Unit Step in A0 F2 based on Data Set 1

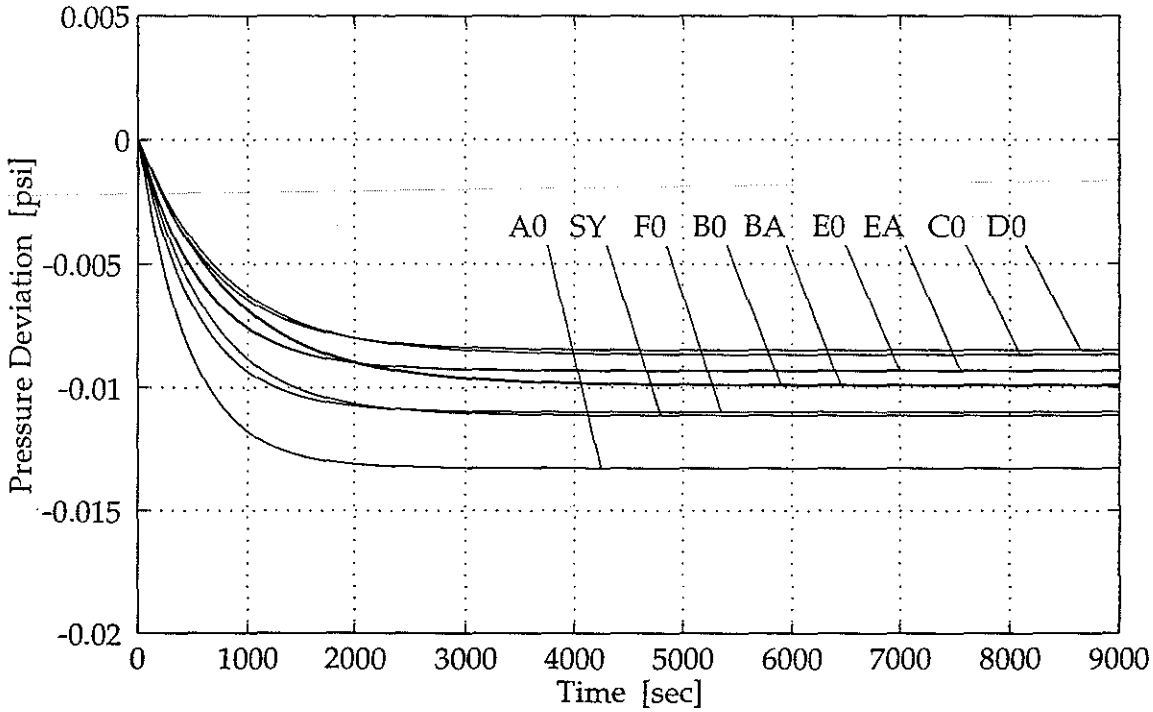

Figure 64. P1 Model for Unit Step in A0 F2 based on Data Set 2

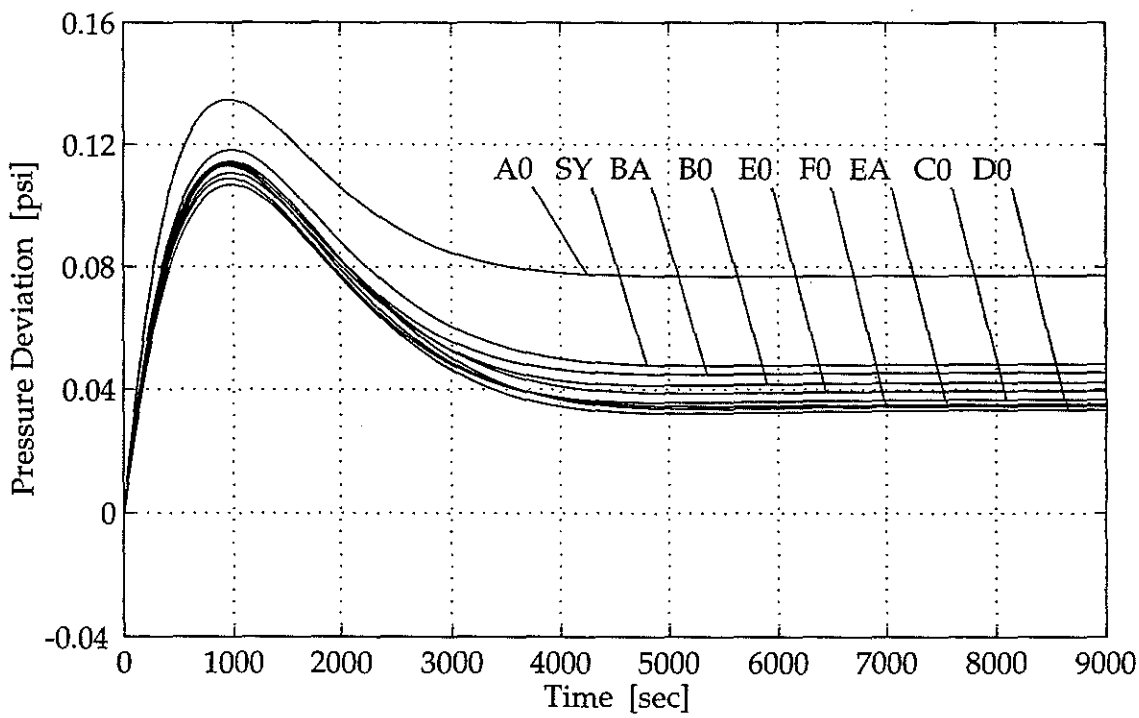

Figure 65. P2 Model for Unit Step in A0 F2 based on Data Set 2

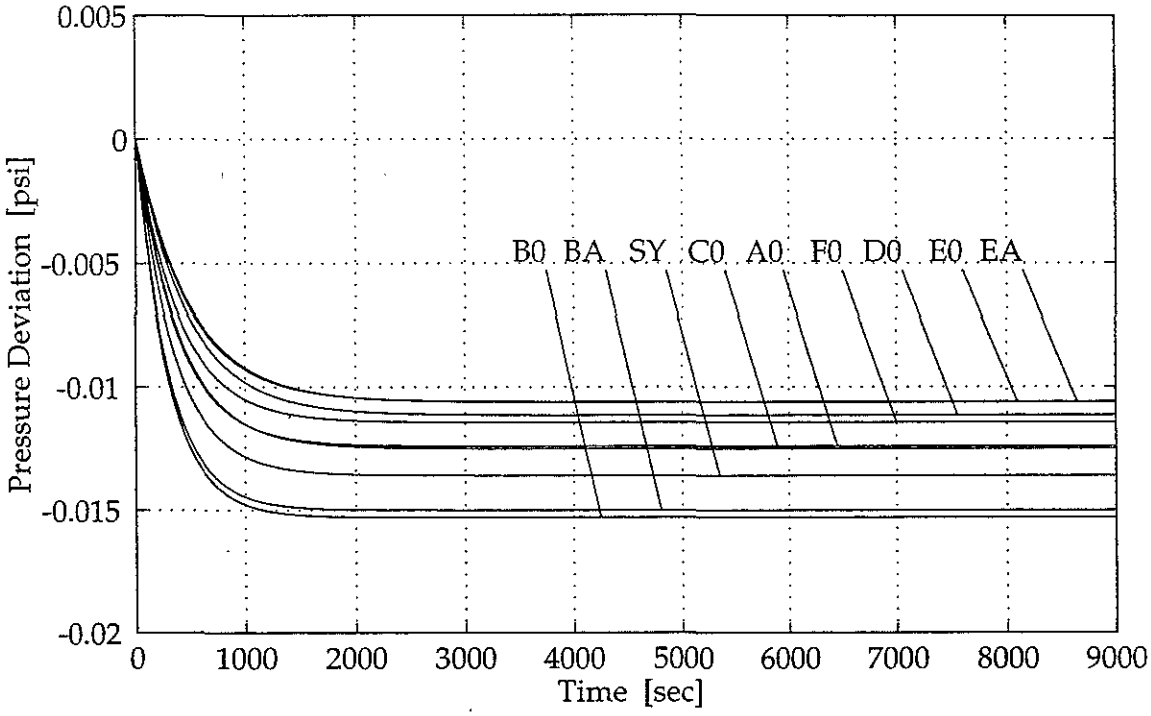

Figure 66. P1 Model for Unit Step in B0 F2 based on Data Set 3

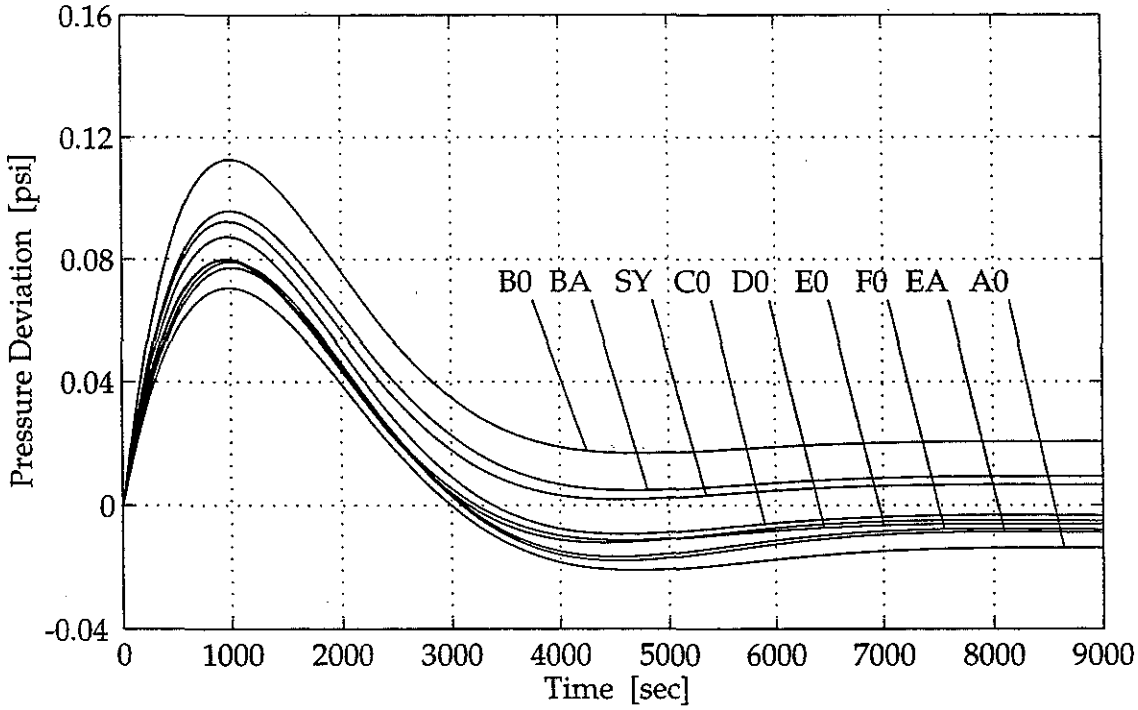

Figure 67. P2 Model for Unit Step in B0 F2 based on Data Set 3

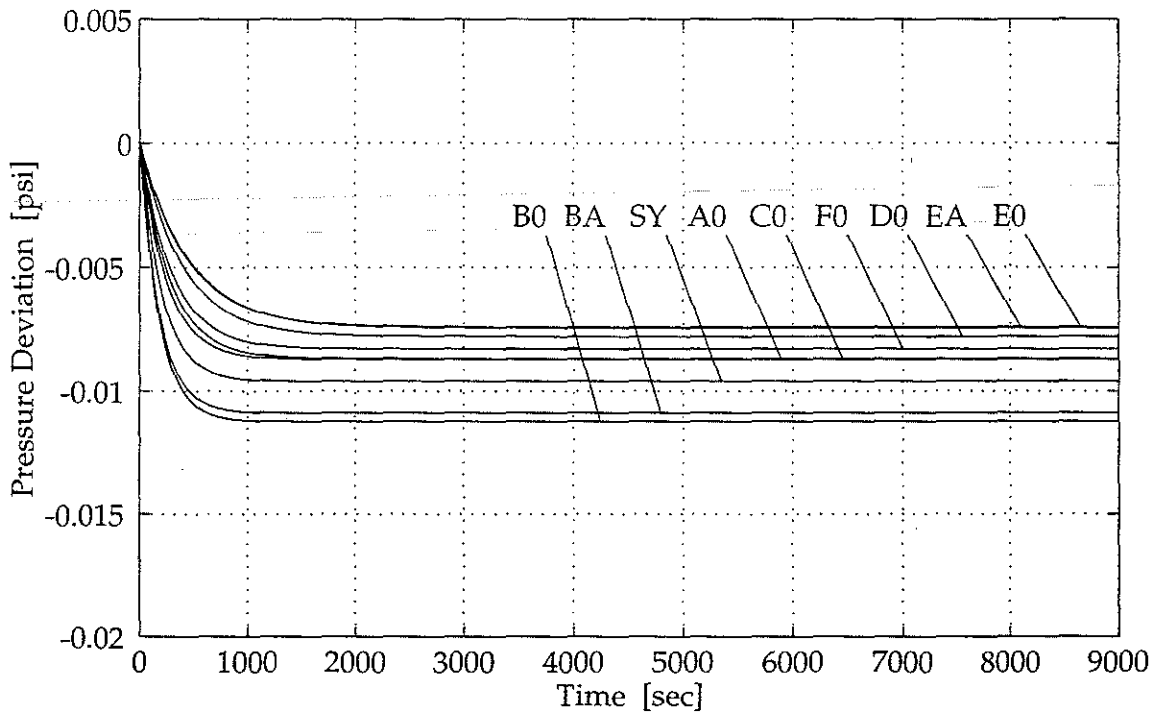

Figure 68. P1 Model for Unit Step in B0 F2 based on Data Set 4

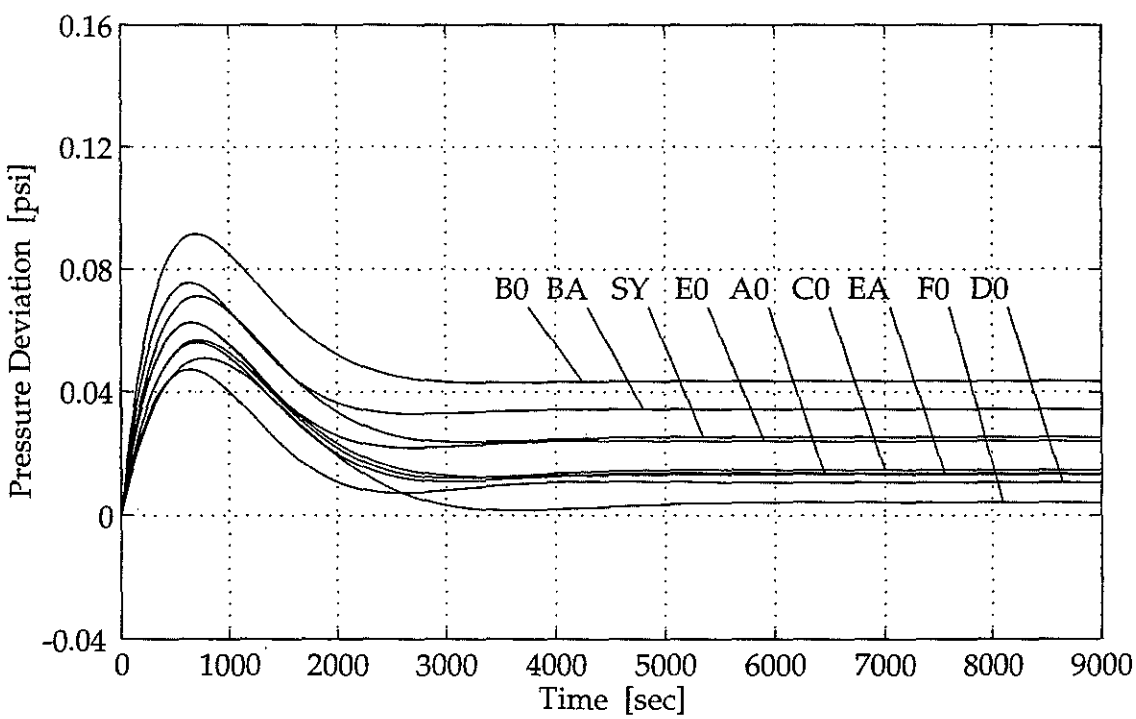

Figure 69. P2 Model for Unit Step in B0 F2 based on Data Set 4

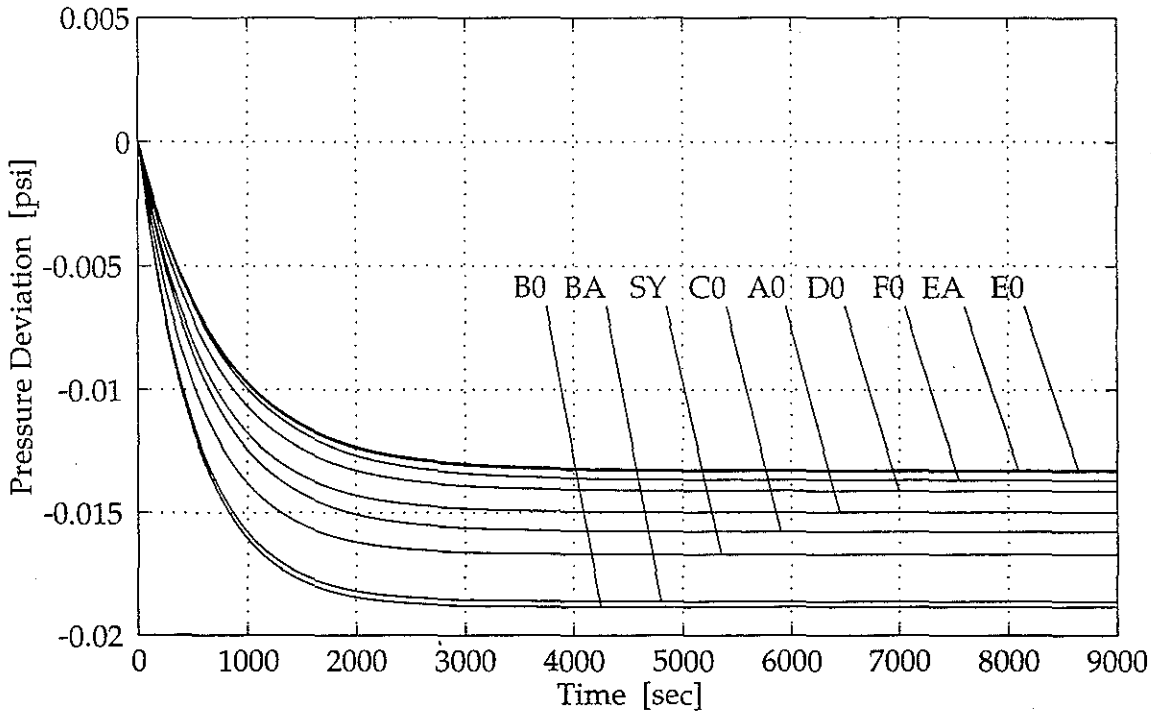

Figure 70. P1 Model for Unit Step in B0 F2 based on Data Set 5

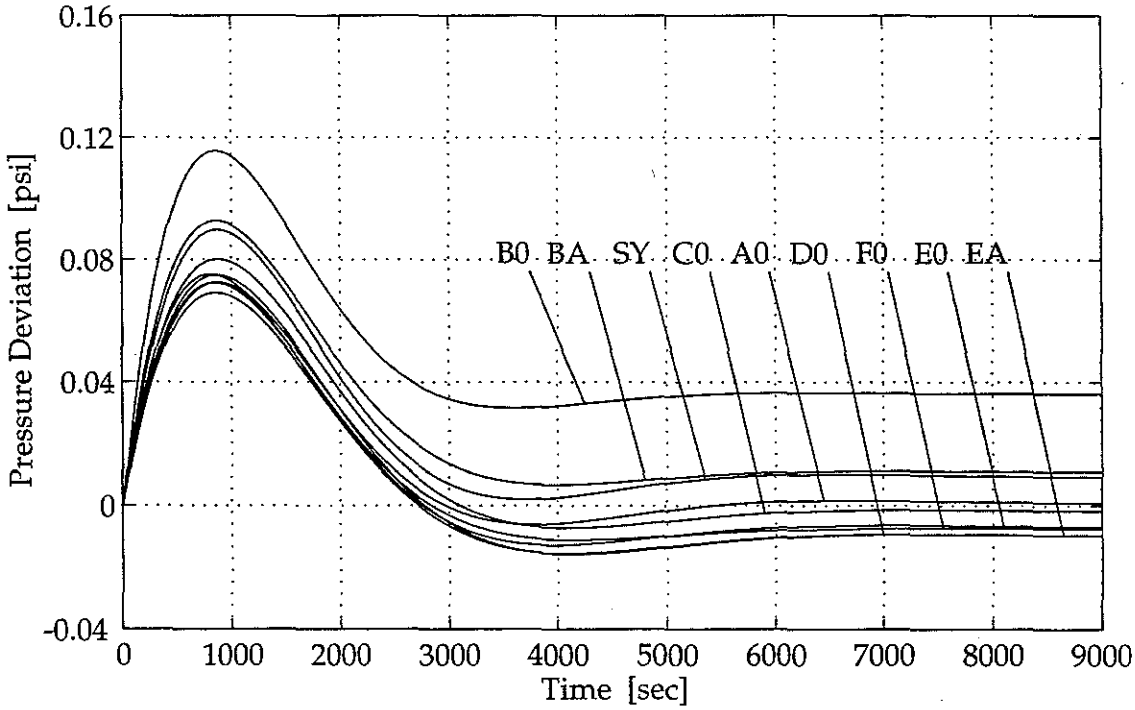

Figure 71. P2 Model for Unit Step in B0 F2 based on Data Set 5

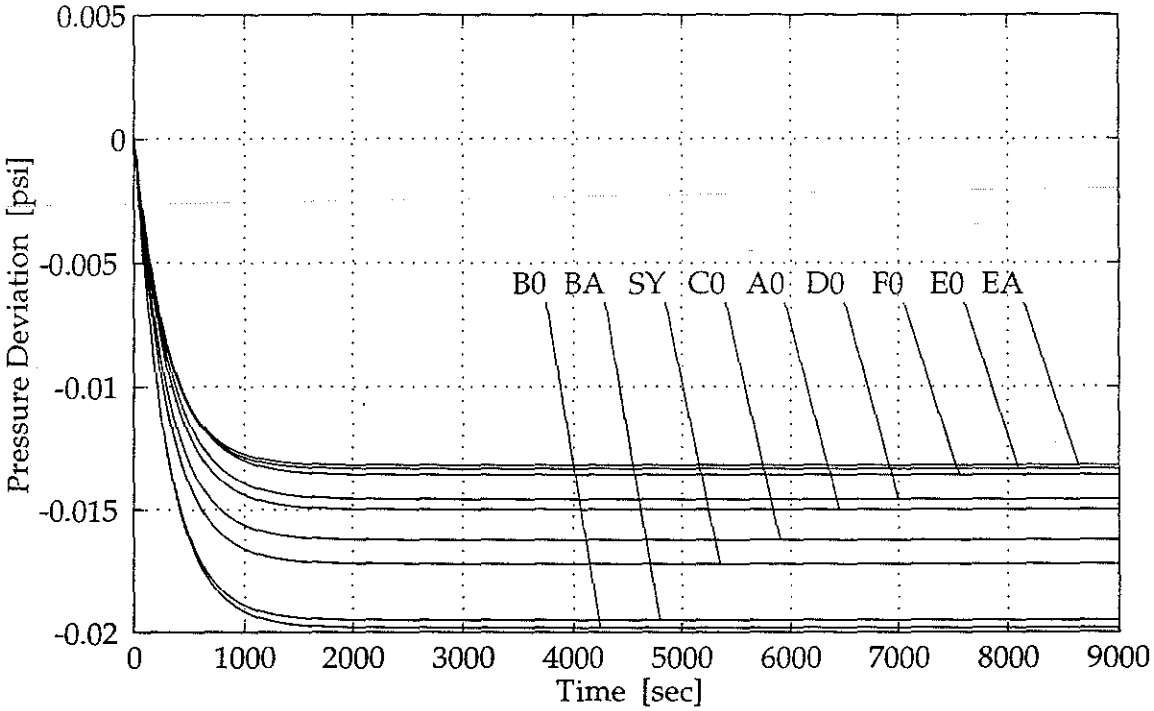

Figure 72. P1 Model for Unit Step in B0 F2 based on Data Set 6

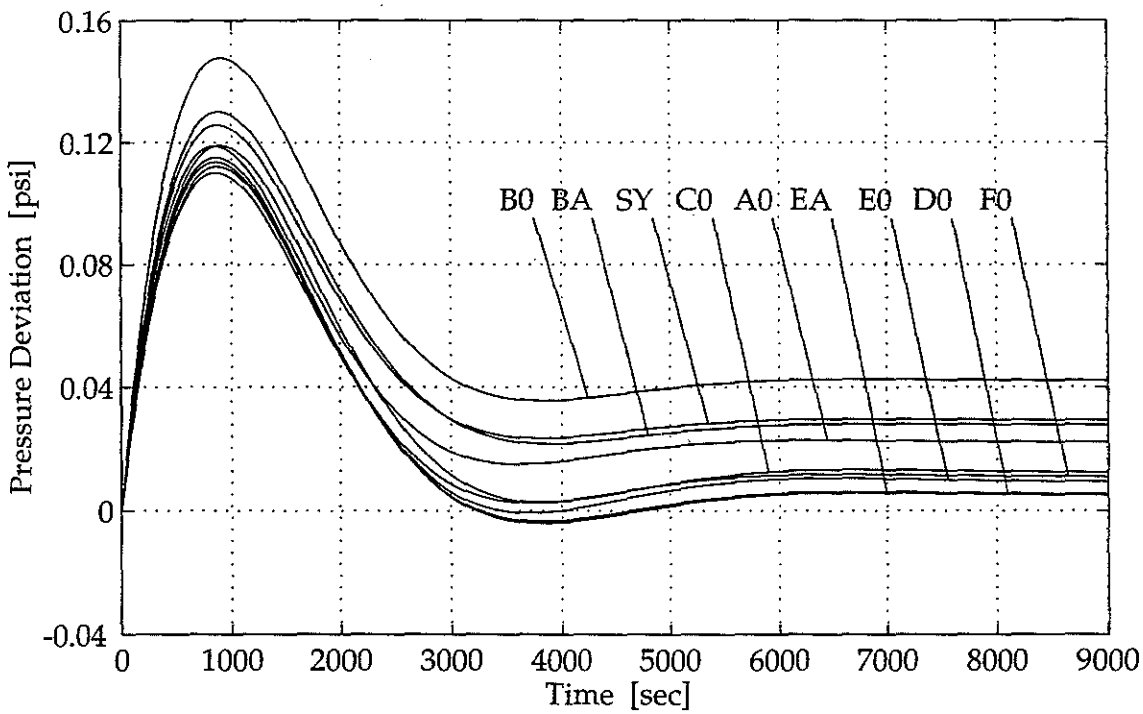

Figure 73. P2 Model for Unit Step in B0 F2 based on Data Set 6

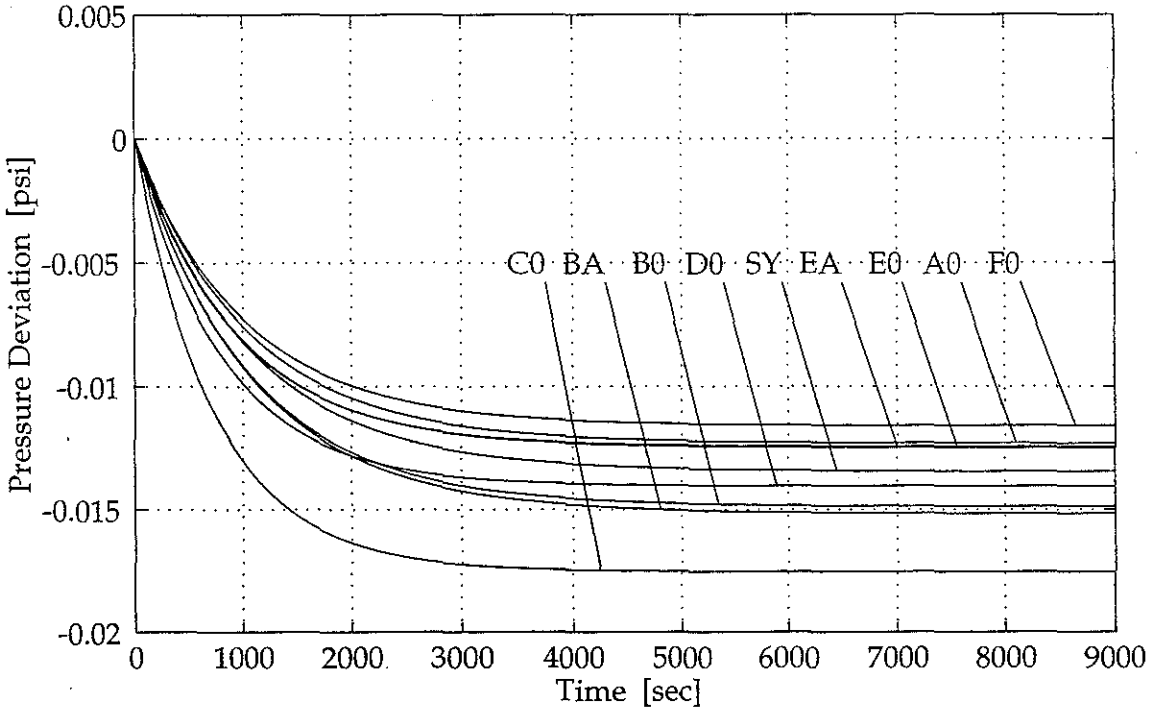

Figure 74. P1 Model for Unit Step in C0 F2 based on Data Set 7

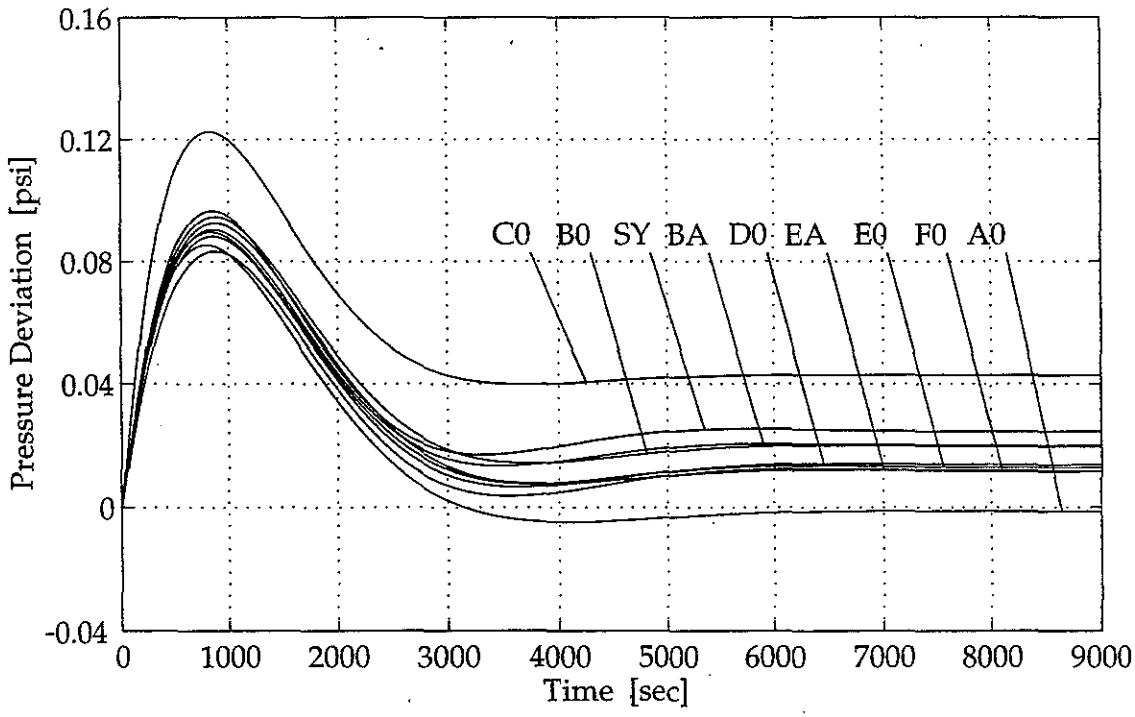

Figure 75. P2 Model for Unit Step in C0 F2 based on Data Set 7

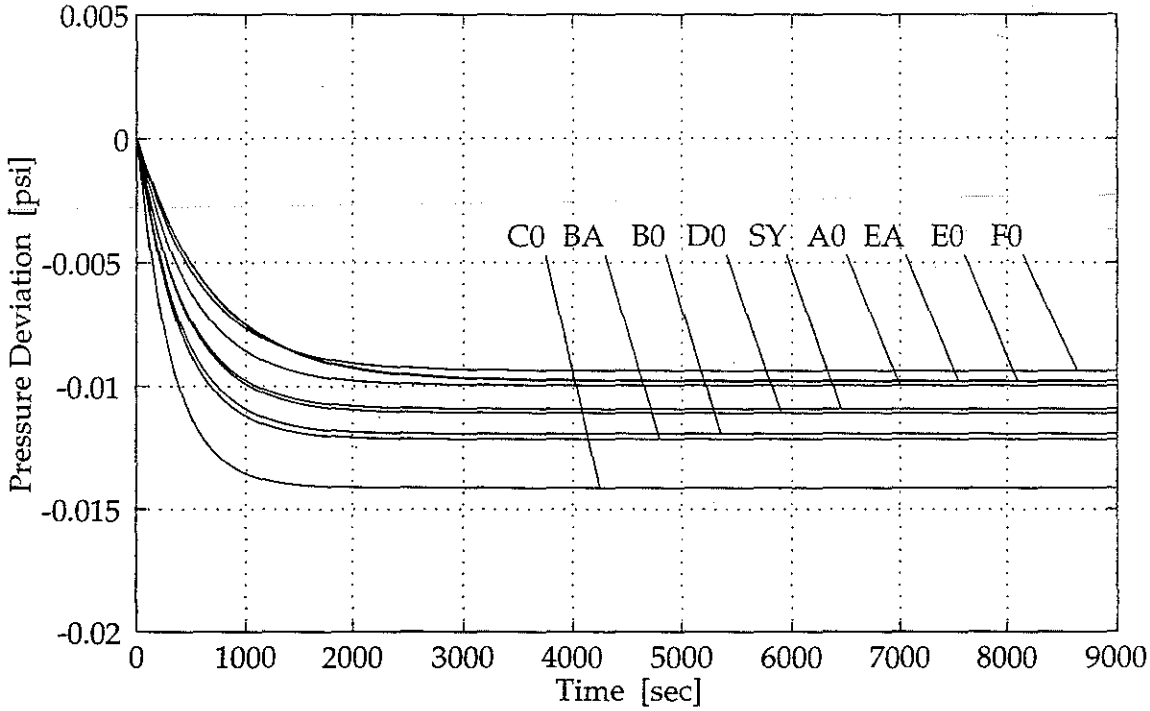

Figure 76. P1 Model for Unit Step in C0 F2 based on Data Set 8

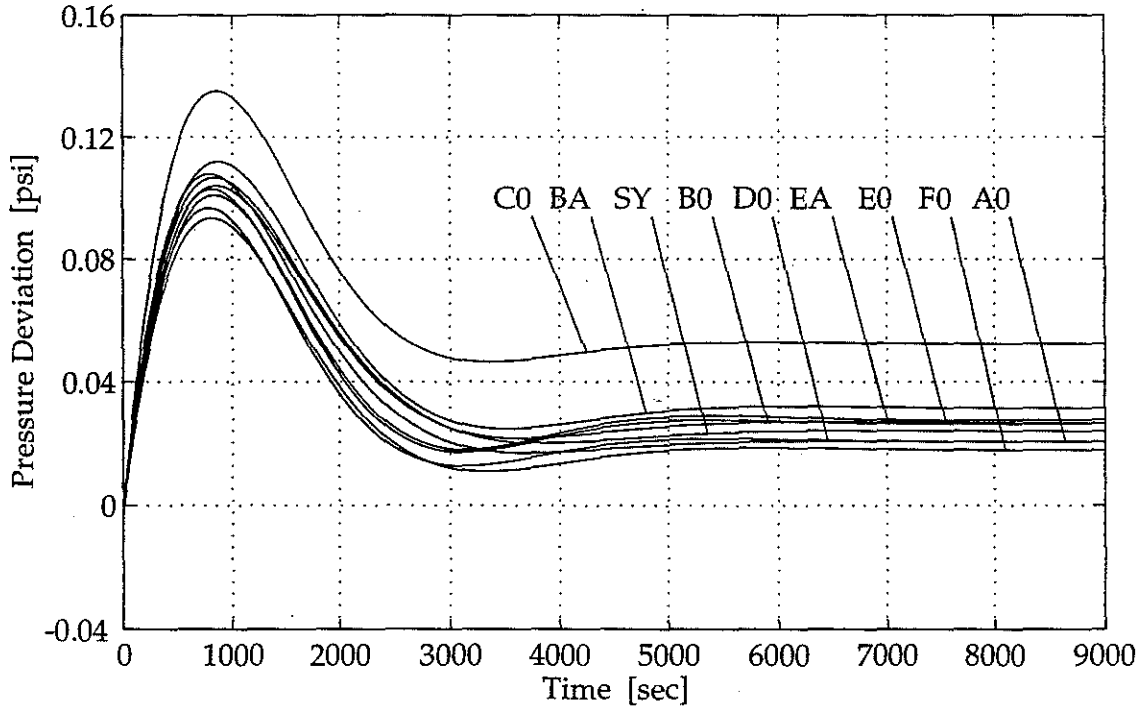

Figure 77. P2 Model for Unit Step in C0 F2 based on Data Set 8

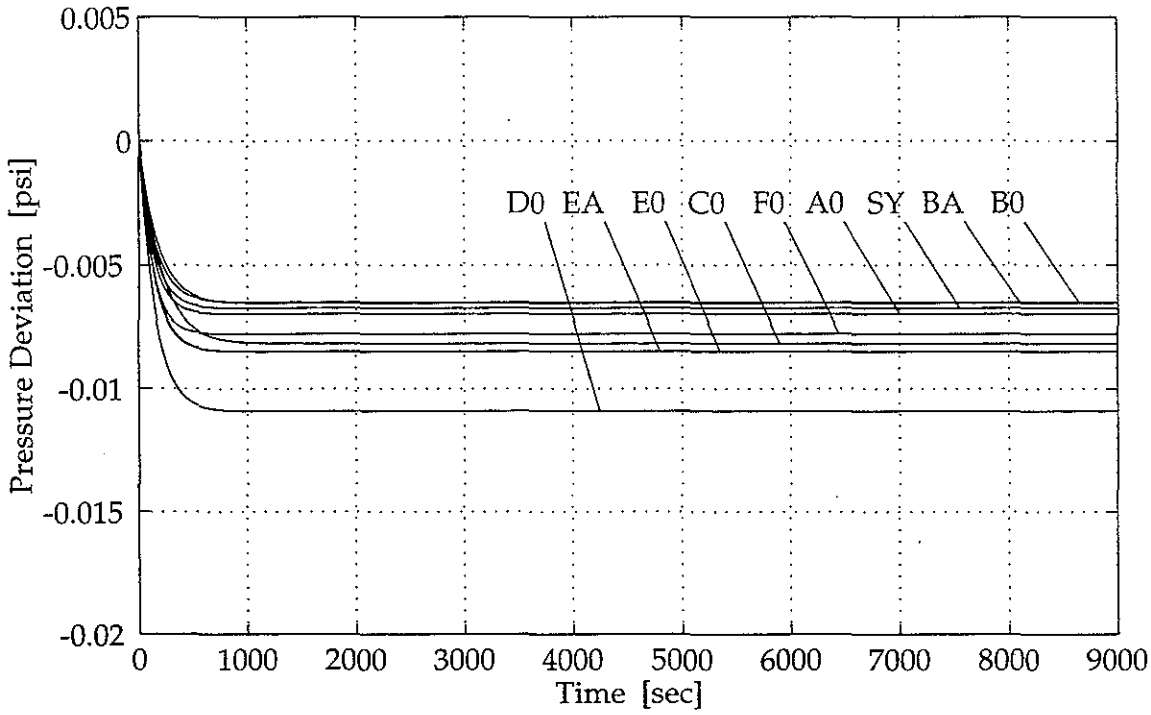

Figure 78. P1 Model for Unit Step in D0 F2 based on Data Set 9

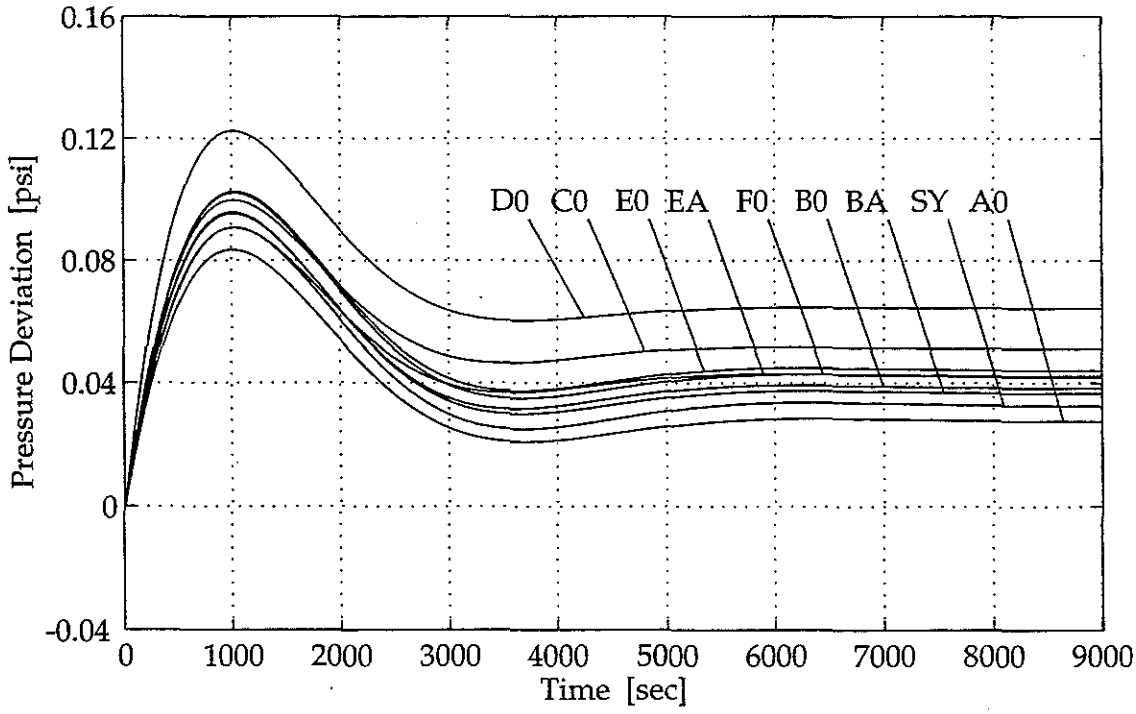

Figure 79. P2 Model for Unit Step in D0 F2 based on Data Set 9

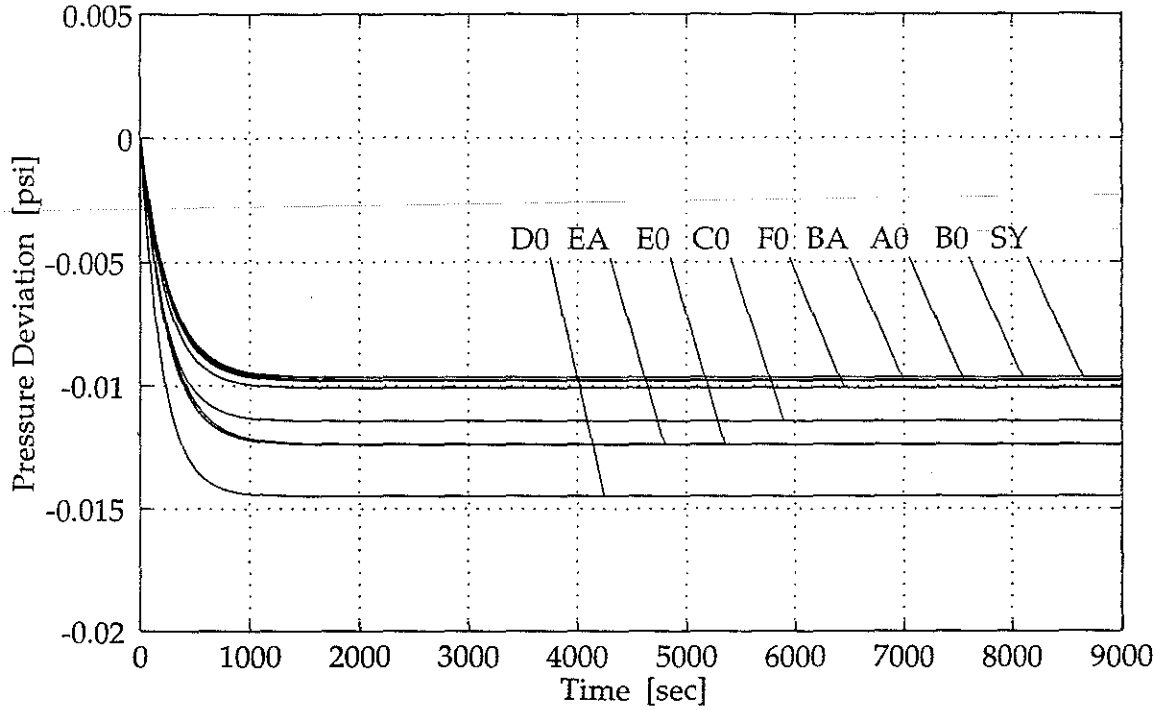

Figure 80. P1 Model for Unit Step in D0 F2 based on Data Set 10

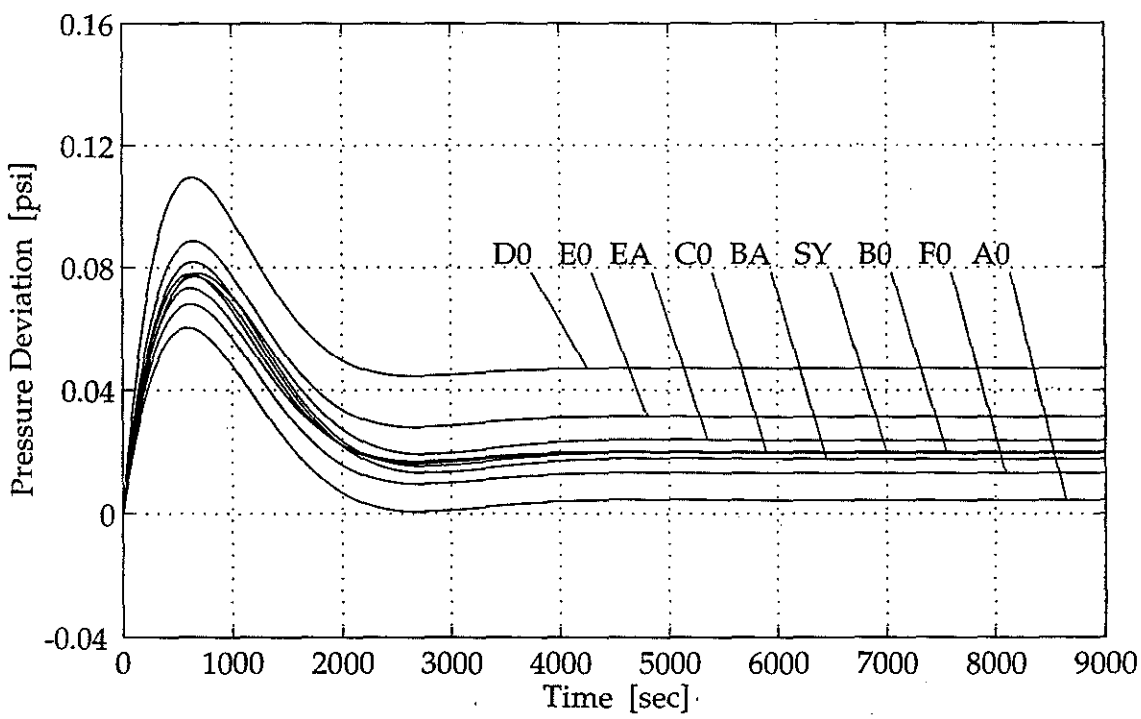

Figure 81. P2 Model for Unit Step in D0 F2 based on Data Set 10

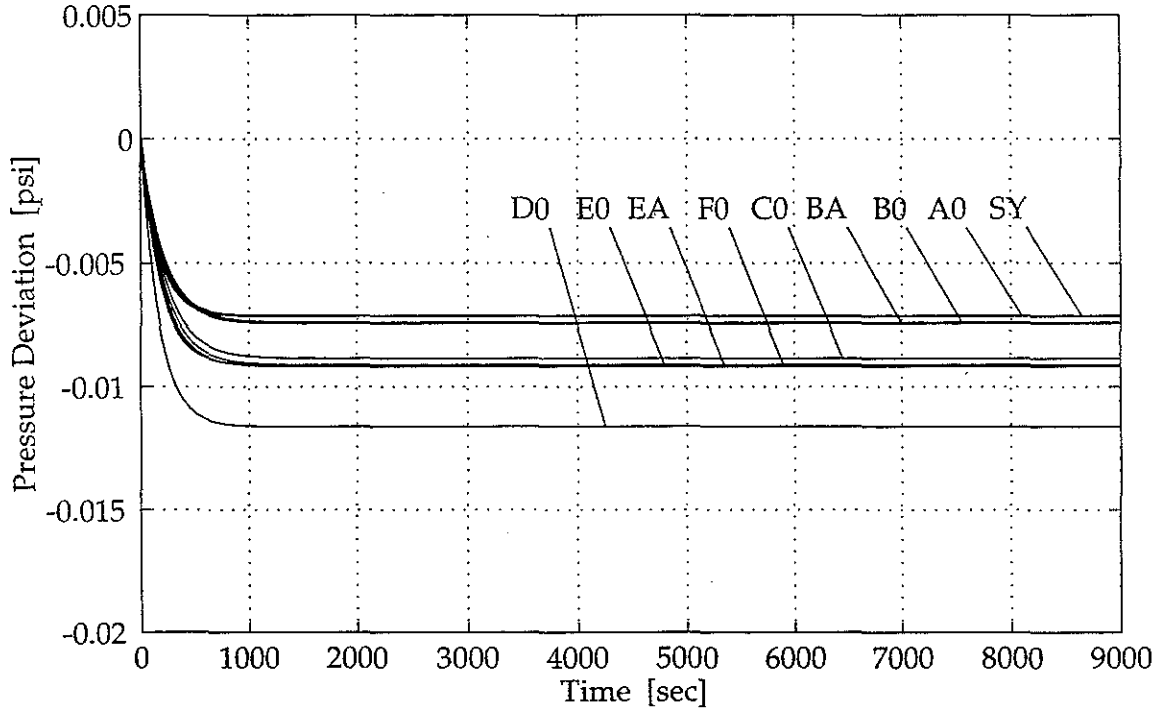

Figure 82. P1 Model for Unit Step in D0 F2 based on Data Set 11

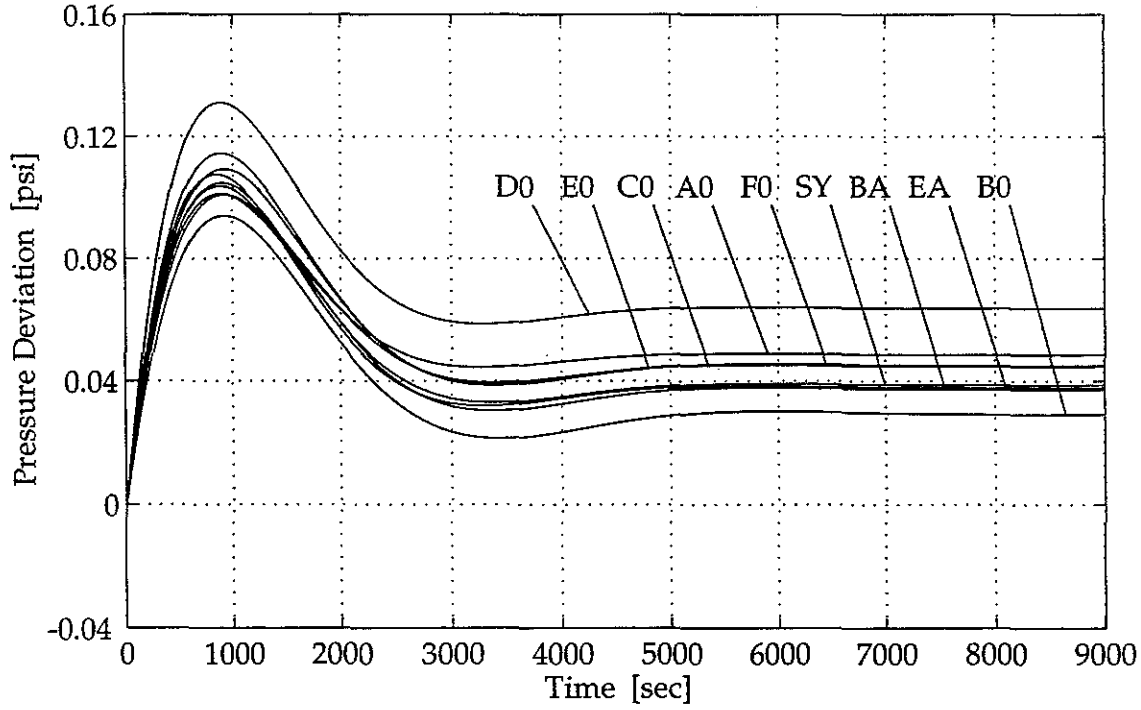

Figure 83. P2 Model for Unit Step in D0 F2 based on Data Set 11

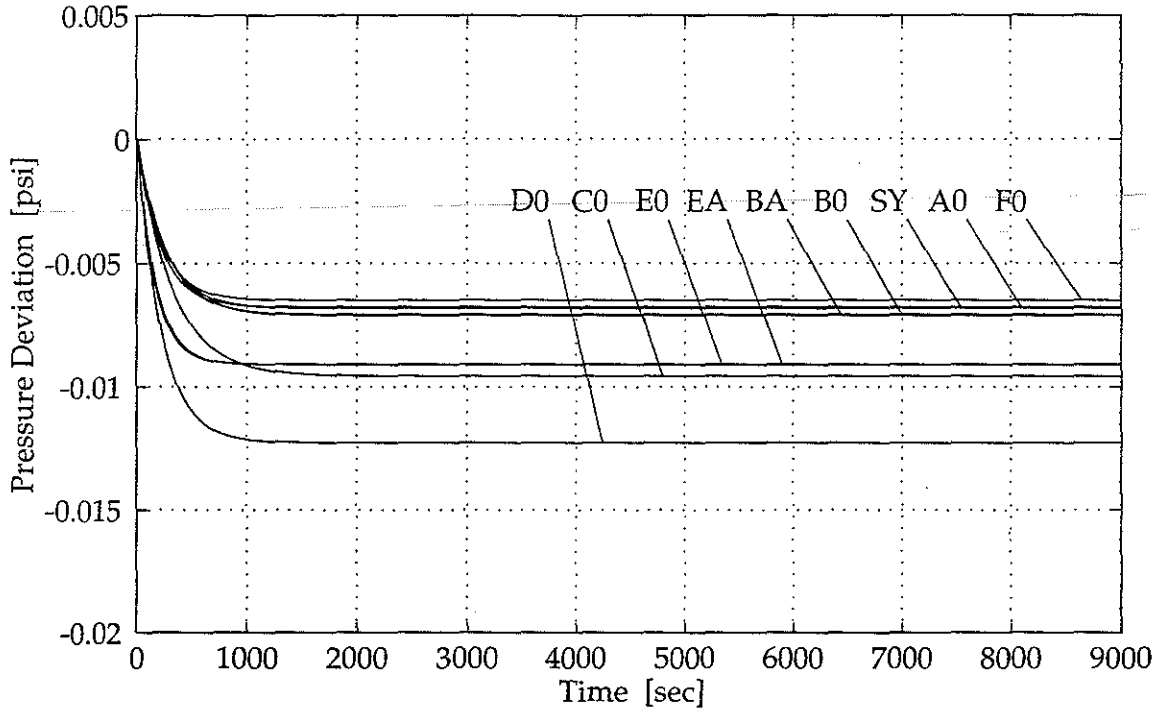

Figure 84. P1 Model for Unit Step in D0 F2 based on Data Set 12

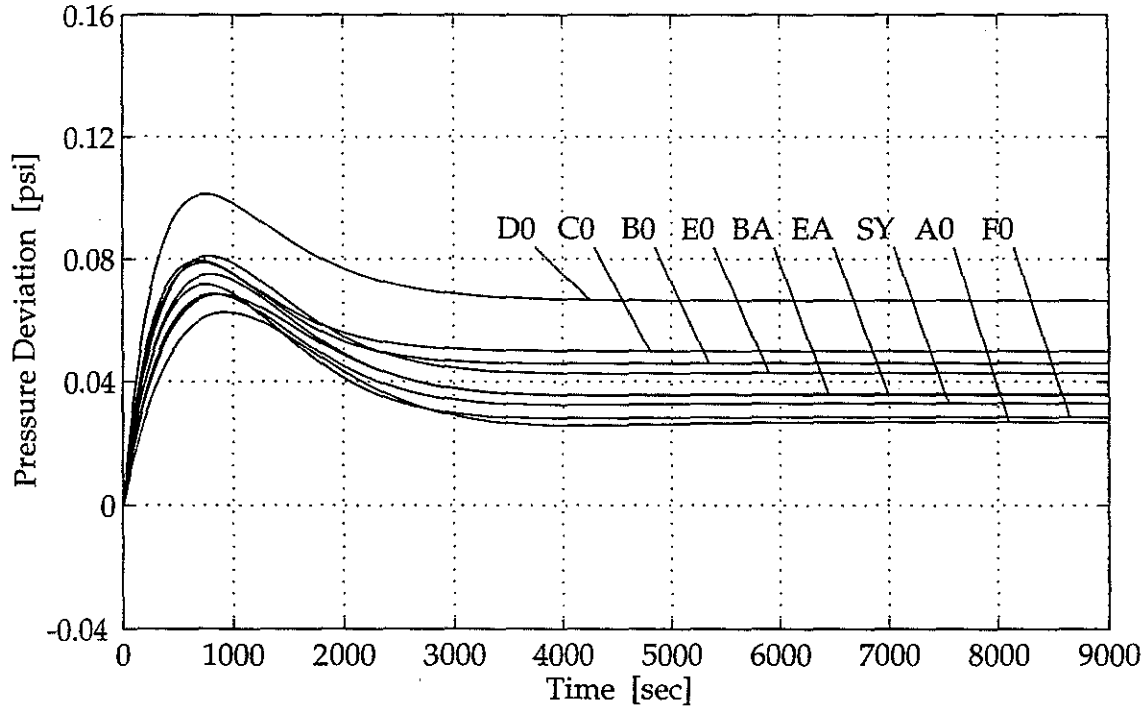

Figure 85. P2 Model for Unit Step in D0 F2 based on Data Set 12

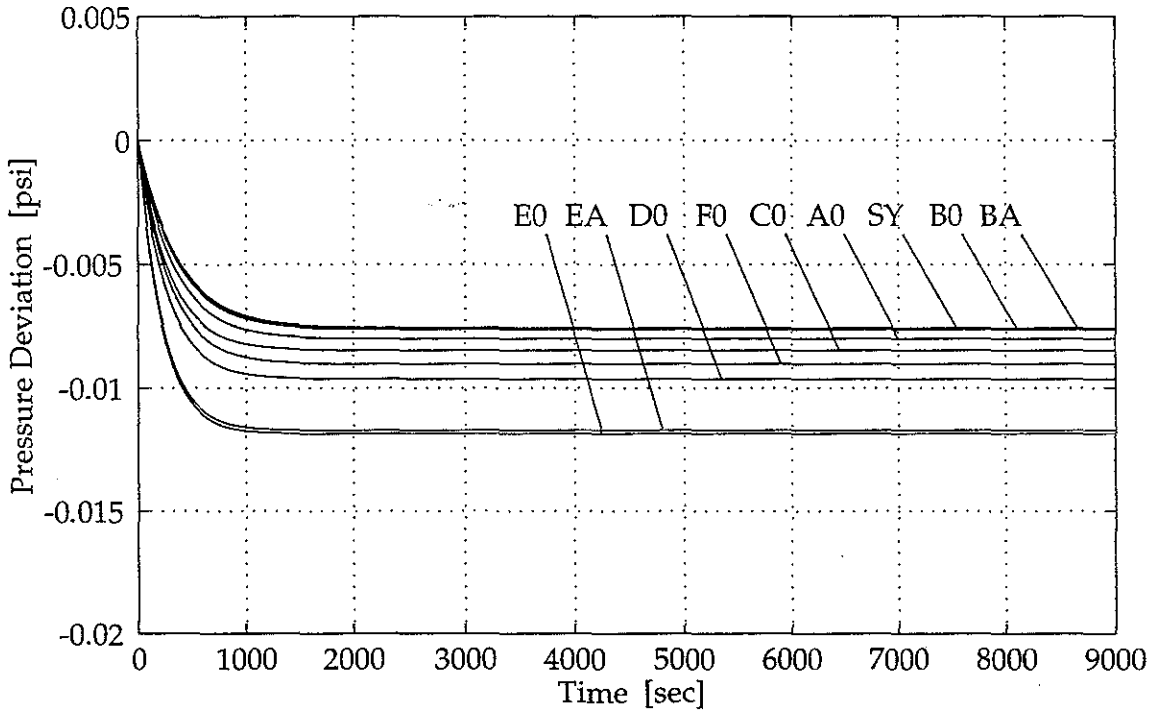

Figure 86. P1 Model for Unit Step in E0 F2 based on Data Set 13

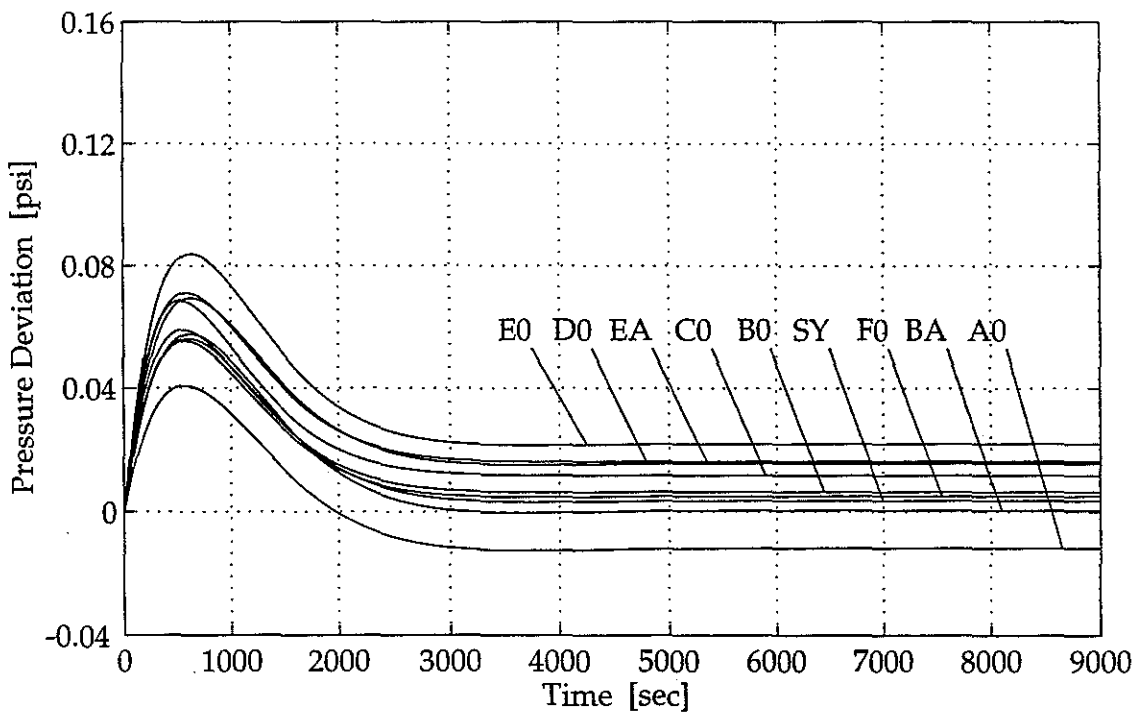

Figure 87. P2 Model for Unit Step in E0 F2 based on Data Set 13

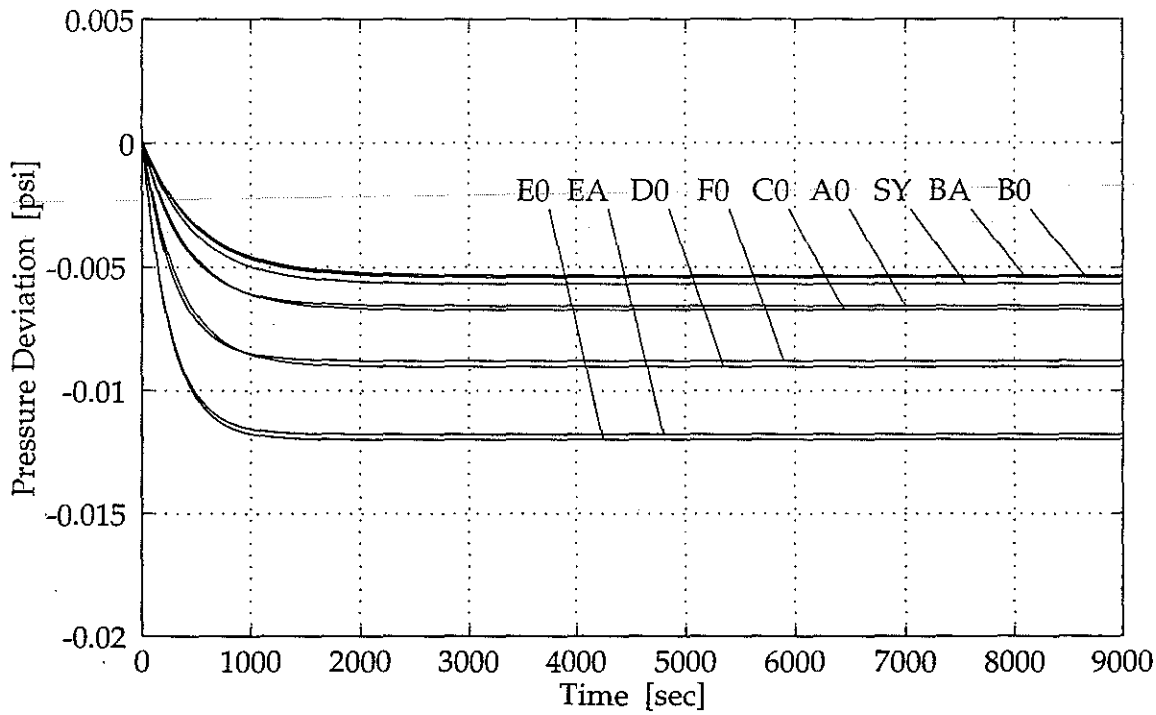

Figure 88. P1 Model for Unit Step in E0 F2 based on Data Set 14

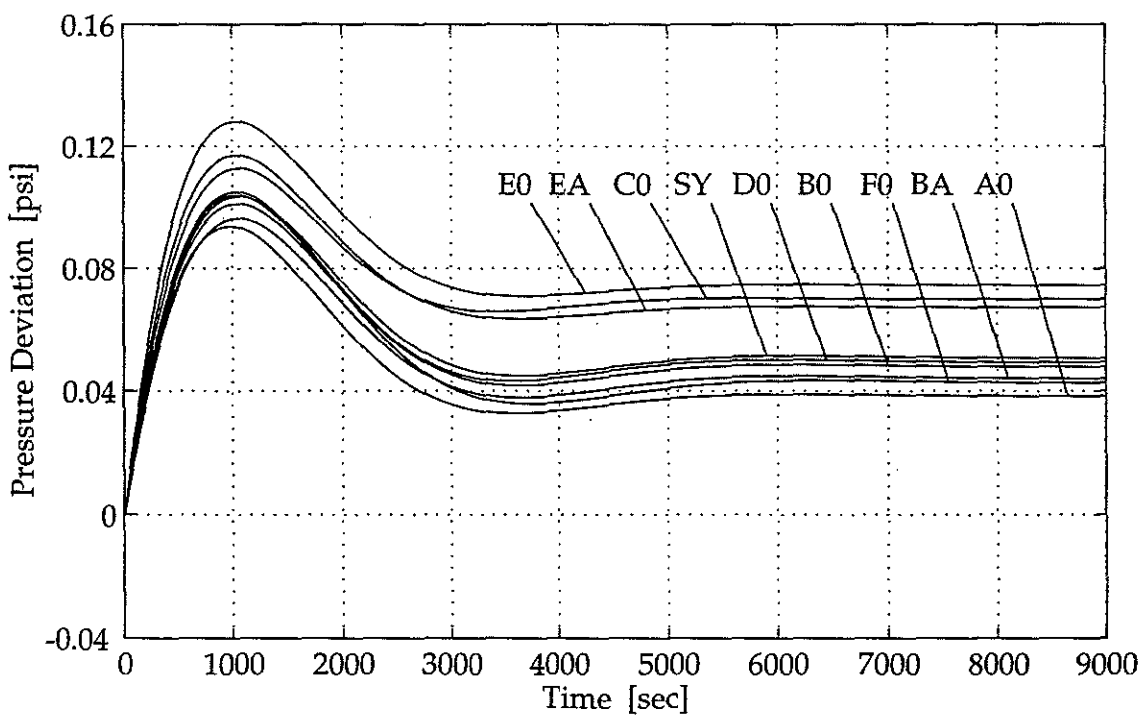

Figure 89. P2 Model for Unit Step in E0 F2 based on Data Set 14

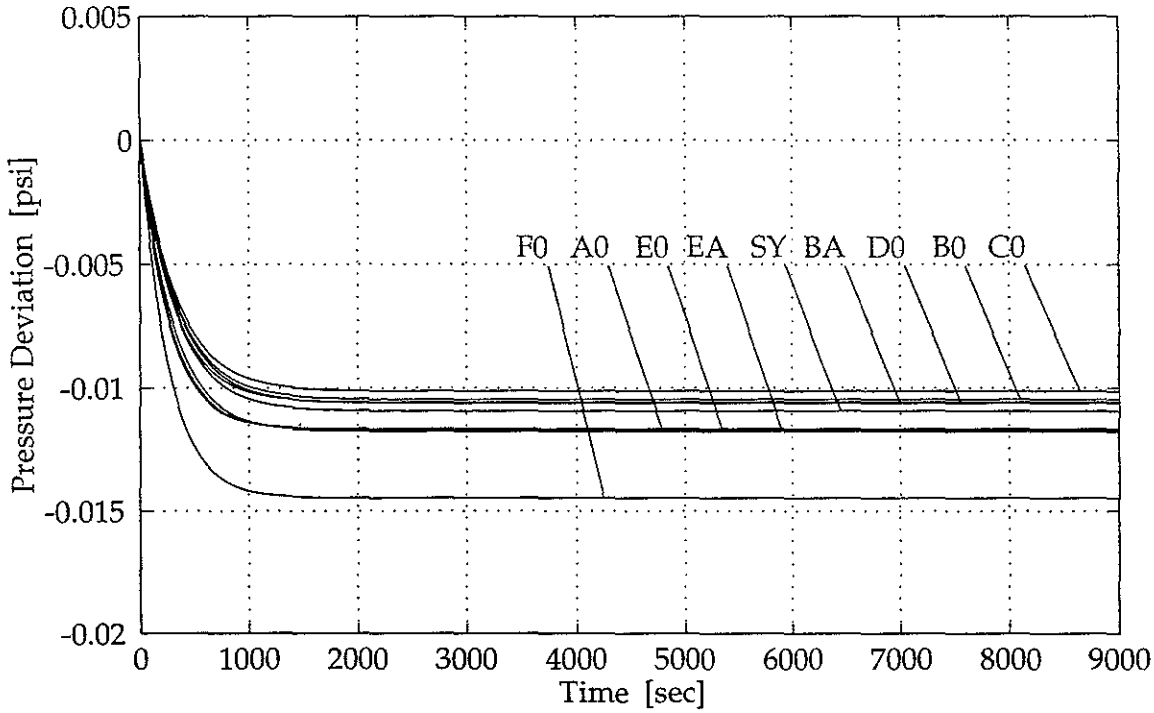

Figure 90. P1 Model for Unit Step in F0 F2 based on Data Set 15

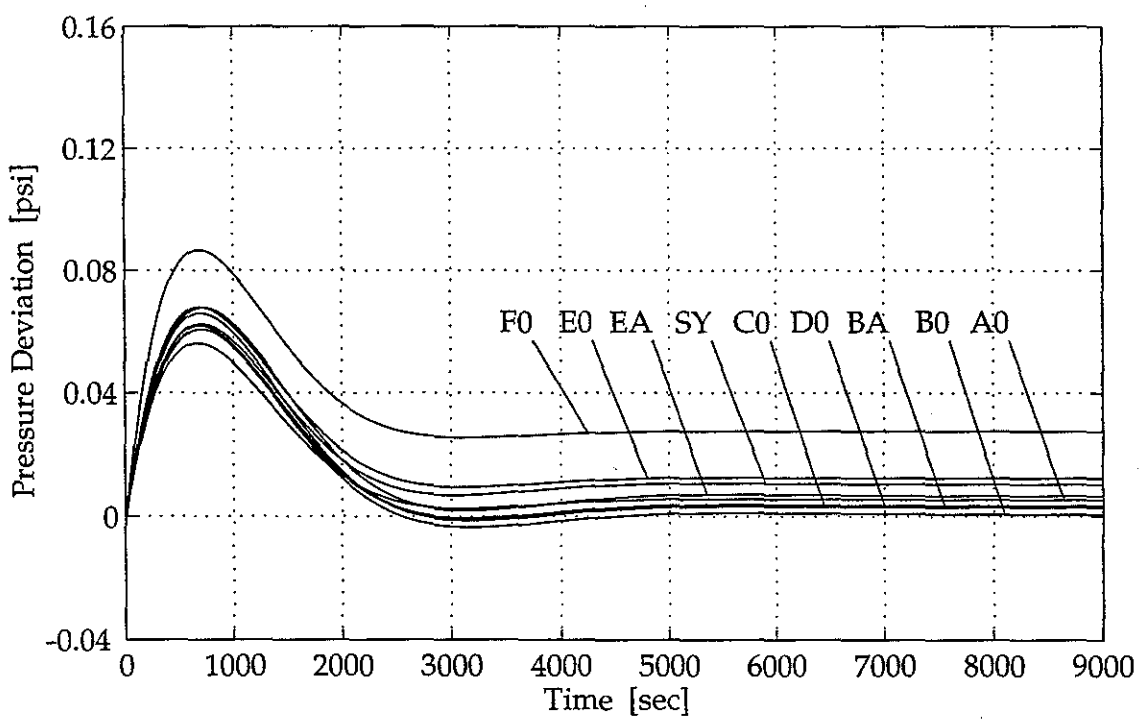

Figure 91. P2 Model for Unit Step in F0 F2 based on Data Set 15

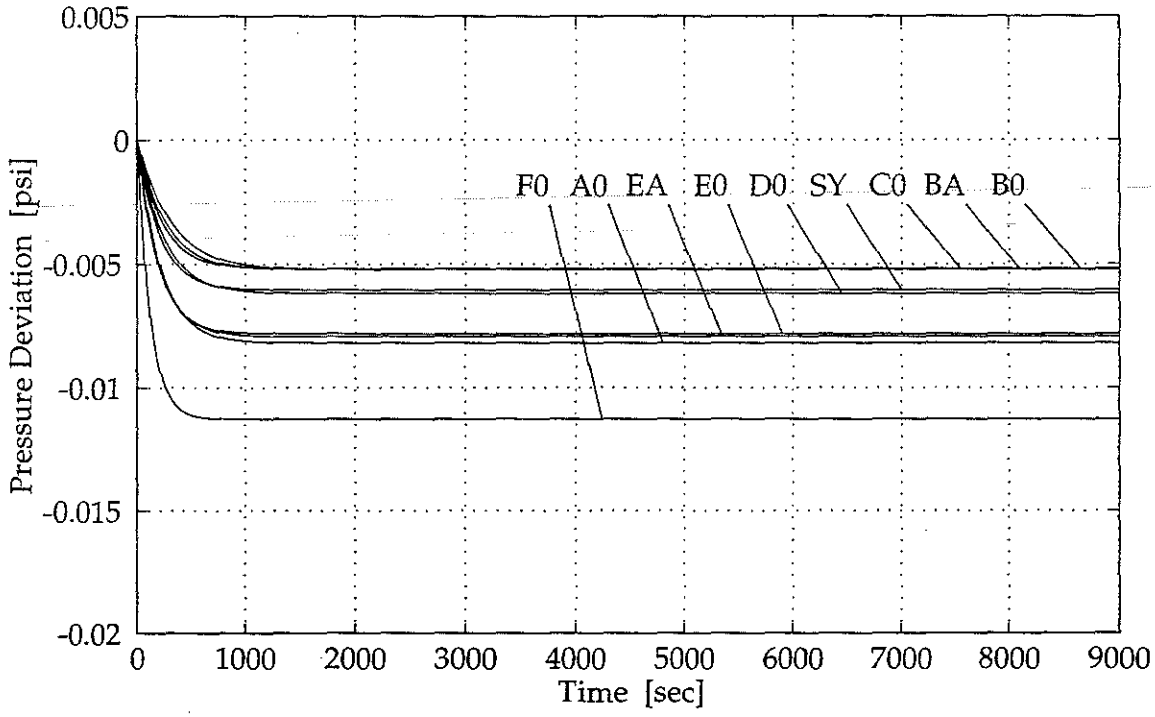

Figure 92. P1 Model for Unit Step in F0 F2 based on Data Set 16

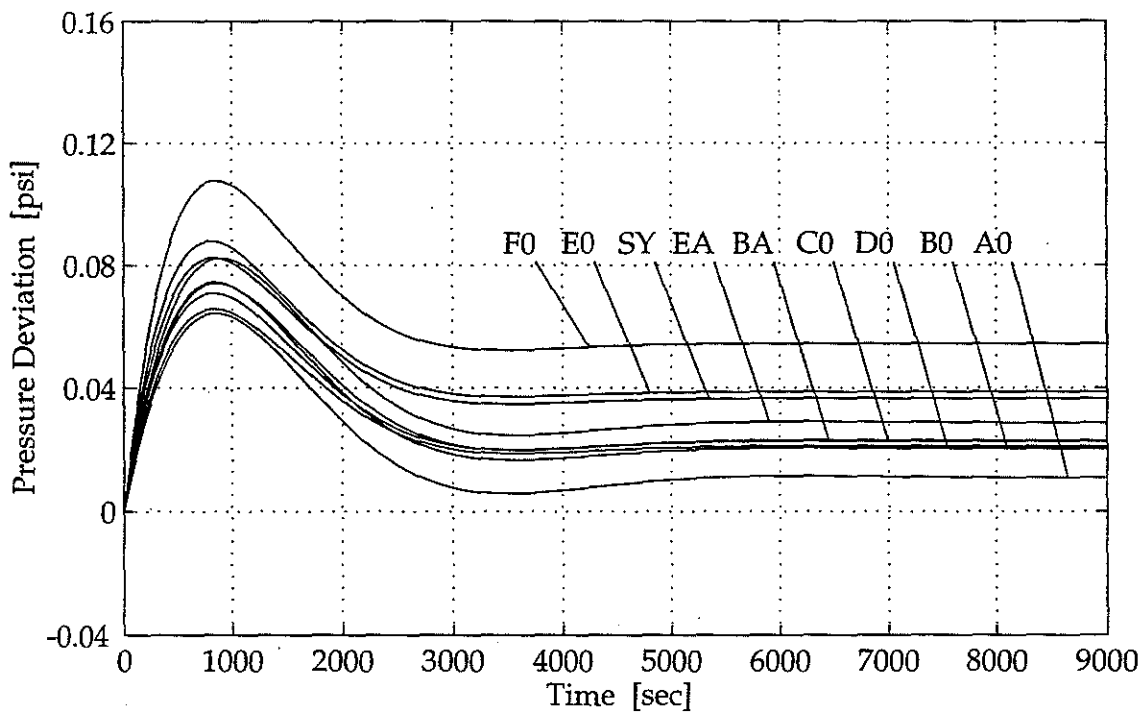

Figure 93. P2 Model for Unit Step in F0 F2 based on Data Set 16

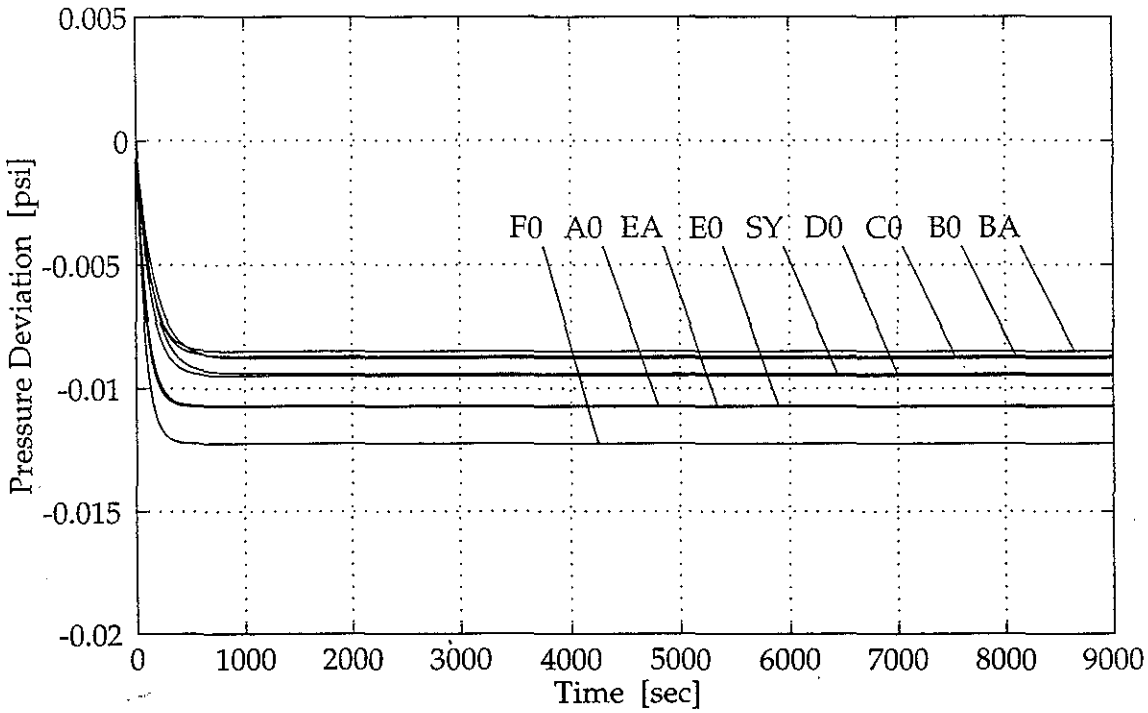

Figure 94. P1 Model for Unit Step in F0 F2 based on Data Set 17

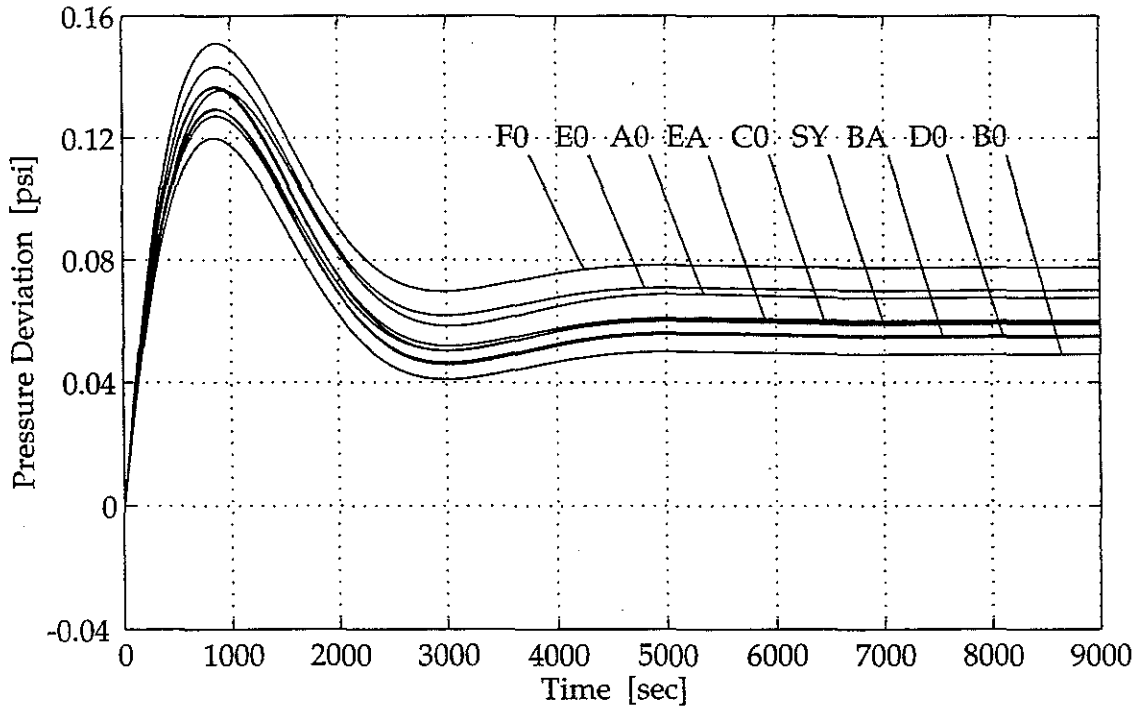

Figure 95. P2 Model for Unit Step in F0 F2 based on Data Set 17

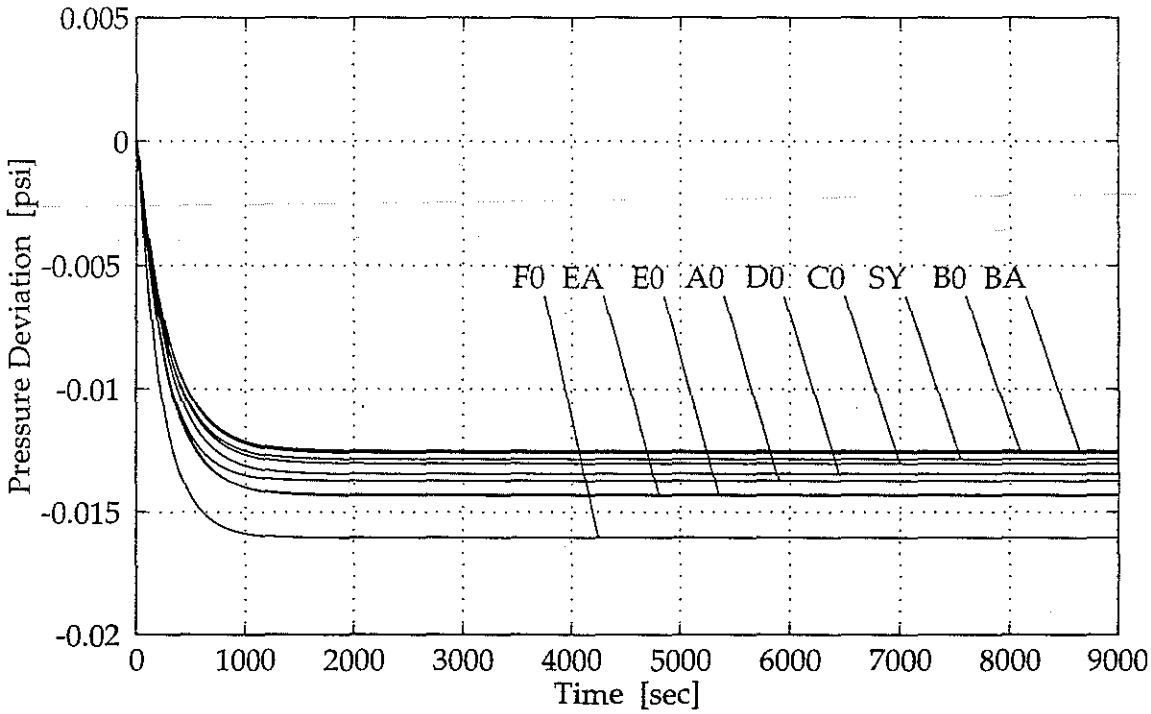

Figure 96. P1 Model for Unit Step in F0 F2 based on Data Set 18

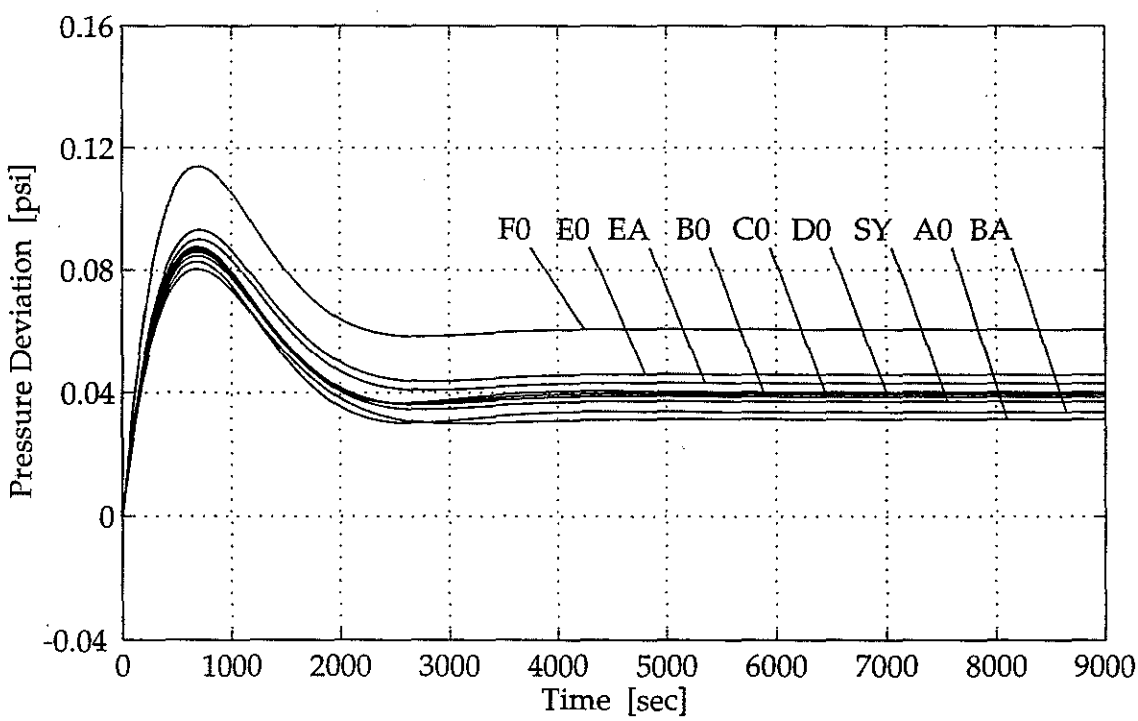

Figure 97. P2 Model for Unit Step in F0 F2 based on Data Set 18

having the least response. The magnitude of the response is gauged at the time corresponding to the last collected data. With this definition of magnitude based on a single value from each curve, the labeling doesn't always correspond to peek values or final values in the figure because the curves do cross in some instances. Note that with a unit positive step input, models for each data set give the same response direction regardless of whether the original data flow step was positive or negative.

The largest difference in model output results among data sets having the same location occurs for the A0 input location. This is seen in Figures 62 and 64 where the P1 responses have similar final values but settling times that are different by a factor of two. P2 responses in Figures 63 and 65 have peak values that differ by 30%, while settling times and final values differ by 50%. The cause for these differences can only be speculated. Measurements at A0 are observed to be the noisiest, and perhaps that noise can be attributed to real flow phenomenon peculiar to that location. Something about the A0 location, such as its position near piping to SY, may cause the model to change with time depending on some subtle unobserved variable. Alternatively, a problem with instrument installation at this location might cause the high noise level. With this very limited number of data sets for the A0 input (or anywhere else), little can be said about an appropriate mean value for model coefficients resulting from data sets containing random disturbances.

Other input locations show somewhat better agreement between model results than does A0. Most simulation characteristics compared

between data sets with like input location are reasonably similar considering the noise and disturbances inherent in the data, with the exception of the final value for P2. However, known or recorded disturbances don't appear to account for all discrepancies. For instance, the large difference between the final P1 values of Figure 68 and Figure 72 are not consistent with the level of noise associated with the P1 data. Nor can the differences be explained by the direction and magnitude of the flow step input. The implication is that either the true system description is time varying or that unknown and unmodeled inputs are influencing the response.

4.2 Black Box Results for Inventory Flow Inputs

Modeling parameters for the inventory flow input data sets are given in Table 2 along with their compressor flow test counterparts. The parameters have largely the same definition as for compressor flow data sets, except no sample numbers are specified for data extension since no extension is required. The parameters are used by program ModFC.m to calculate model coefficients, and simulations based on those coefficients are used to obtain the sse values also presented in Table 2. Since no model is made for suction pressure response, no sse values appear for P1 in Table 2 for inventory flow data sets.

The unit step response associated with models of data sets 19 through 22 are shown in Figures 98 through 101. Compared to the compressor flow models, these four data sets produce remarkably similar models, despite differences in flow step size and direction. For practical purposes, the responses of P2 at each compressor location are also identical. Notice also the

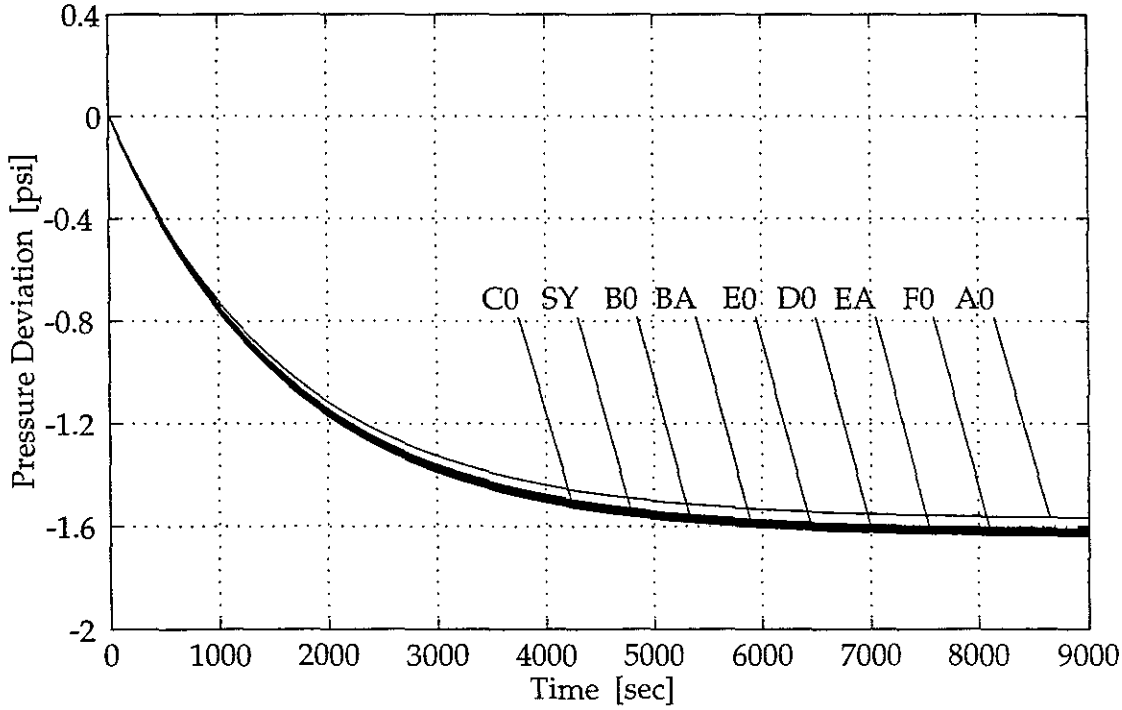

Figure 98. P2 Model for Unit Step in FC Based on Data Set 19

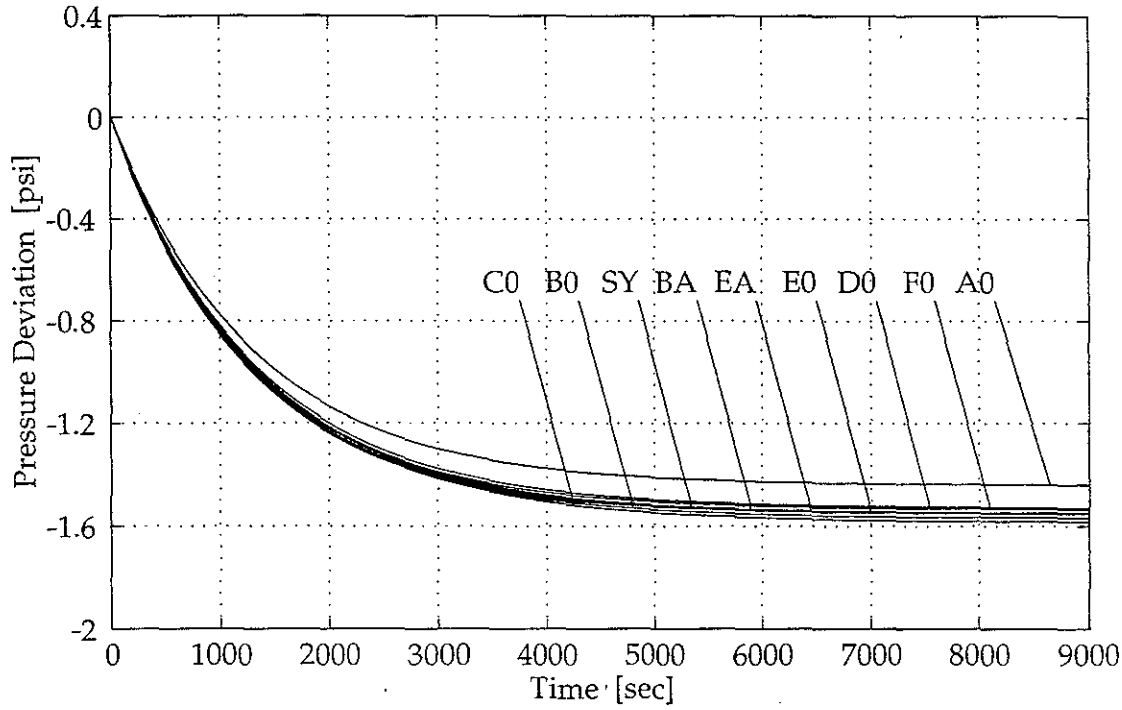

Figure 99. P2 Model for Unit Step in FC Based on Data Set 20

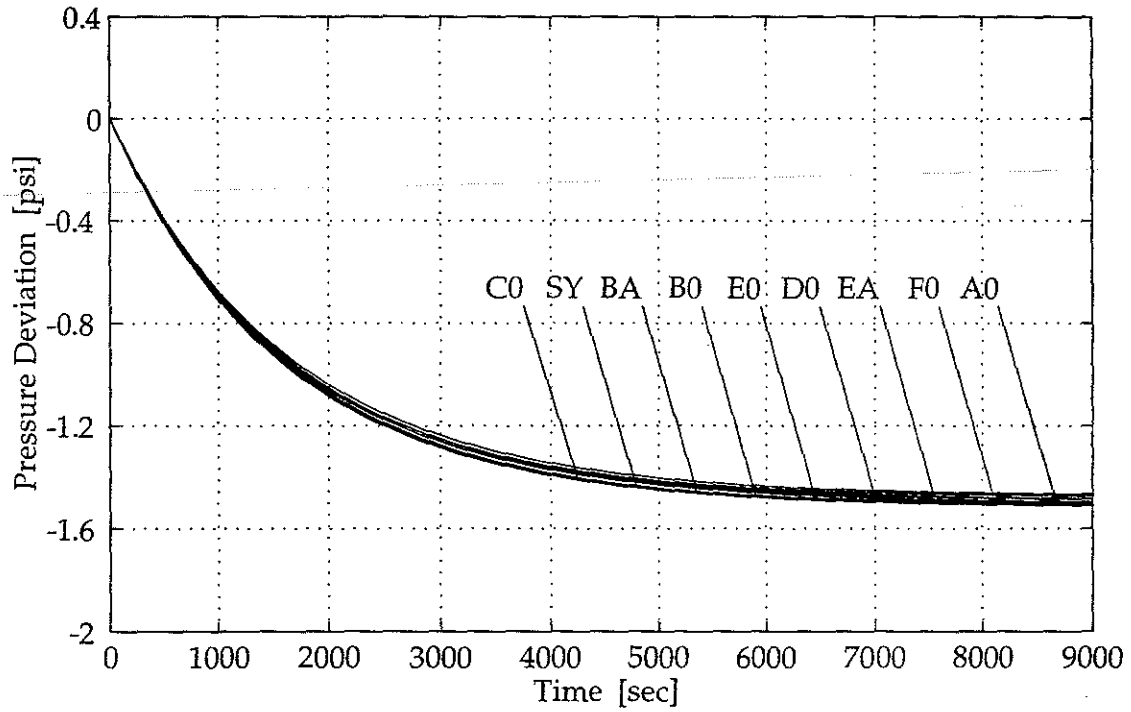

Figure 100. P2 Model for Unit Step in FC Based on Data Set 21

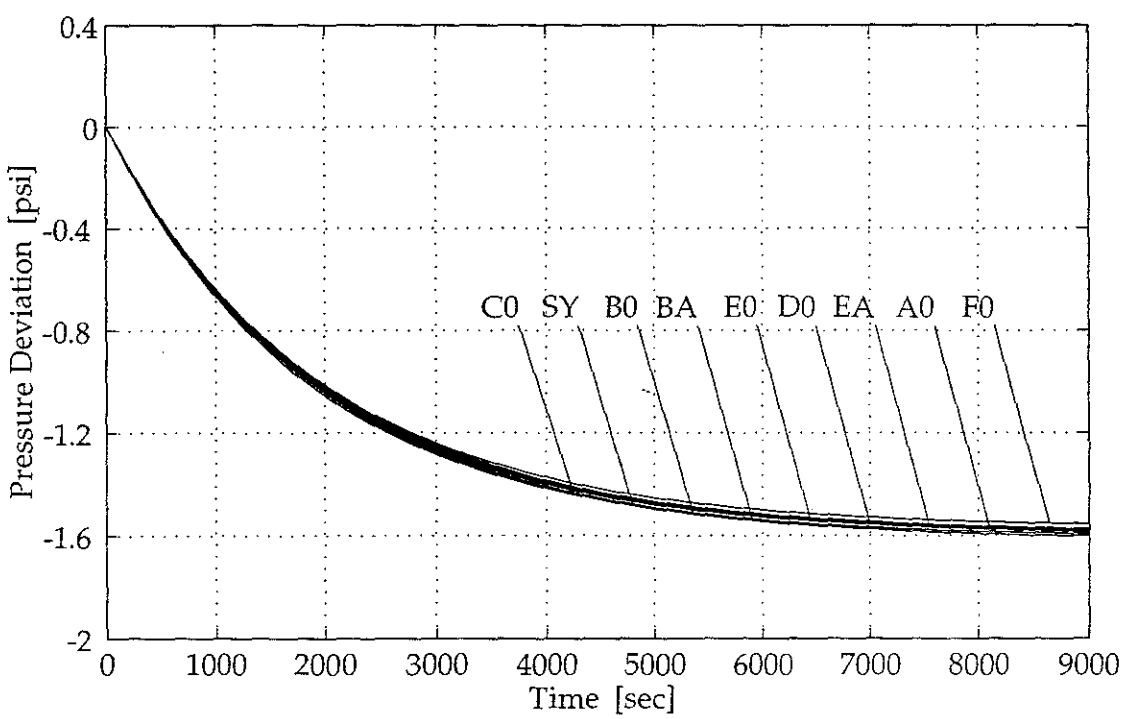

Figure 101. P2 Model for Unit Step in FC Based on Data Set 22

relatively large effect a 1 g/s inventory flow change has upon discharge pressure, as compared to a 1 g/s compressor flow change.

4.3 Averaged and Validated Black Box System Model

One method to deal with the variability in modeling results at a given input location is to simply average the model coefficients found for that location. In this section an averaged model is validated against data not used to create the model. The locational averaged models, taken collectively, provide a mathematical description of the dynamics of the entire compressor and pressure header system. A resulting system model is described which consists of model equations relating F2 to P1s and to P2s around the system ring for each compressor input location, and also of model equations relating inventory flow to P2s.

4.3.1 Model Averaging. As noted in Section 3.3.1, many data sets are not long enough to capture all of the discharge pressure decay to steady state. The final P2 values produced by a model is wholly determined by the last data points on which the model is based, and therefore contain a systematic error. Thus, problems with final P2 values for compressor flow inputs arises in a well understood manner compared to other characteristic differences. The other characteristic differences between data sets of a particular input location are treated as random, for which the best available model becomes one whose coefficients are the averages of coefficients from individual data set models. Model coefficients obtained in this manner for each input location are given in Table 3. The averaged coefficients are in turn used to simulate unit step

Table 3. Location Averaged Model Coefficients (Page 1 of 3)

Input Location	Model Coeff.	Output Location		
		A0	B0	BA
A0	a11	-9.8043e-01	-9.8691e-01	-9.8783e-01
	b11	-2.4747e-04	-1.2122e-04	-1.1247e-04
	a12	-1.9765e+00	-1.9841e+00	-1.9833e+00
	a22	+9.7664e-01	+9.8417e-01	+9.8339e-01
	b12	+2.8263e-03	+1.7840e-03	+1.8017e-03
	b22	-2.8199e-03	-1.7814e-03	-1.7992e-03
B0	a11	-9.8534e-01	-9.8173e-01	-9.8186e-01
	b11	-1.8214e-04	-2.8662e-04	-2.7938e-04
	a12	-1.9900e+00	-1.9884e+00	-1.9889e+00
	a22	+9.9004e-01	+9.8849e-01	+9.8895e-01
	b12	+1.2448e-03	+1.7978e-03	+1.5224e-03
	b22	-1.2442e-03	-1.7954e-03	-1.5208e-03
C0	a11	-9.9274e-01	-9.9151e-01	-9.9134e-01
	b11	-7.7906e-05	-1.0837e-04	-1.1226e-04
	a12	-1.9902e+00	-1.9904e+00	-1.9908e+00
	a22	+9.9029e-01	+9.9050e-01	+9.9082e-01
	b12	+1.3625e-03	+1.3751e-03	+1.3728e-03
	b22	-1.3618e-03	-1.3738e-03	-1.3713e-03
D0	a11	-9.7128e-01	-9.7557e-01	-9.7680e-01
	b11	-2.1401e-04	-1.8284e-04	-1.7646e-04
	a12	-1.9894e+00	-1.9880e+00	-1.9892e+00
	a22	+9.8949e-01	+9.8803e-01	+9.8930e-01
	b12	+1.0925e-03	+1.3016e-03	+1.2218e-03
	b22	-1.0908e-03	-1.2990e-03	-1.2197e-03
E0	a11	-9.8593e-01	-9.8835e-01	-9.8825e-01
	b11	-1.0341e-04	-7.7633e-05	-7.8387e-05
	a12	-1.9888e+00	-1.9871e+00	-1.9889e+00
	a22	+9.8889e-01	+9.8721e-01	+9.8899e-01
	b12	+9.9621e-04	+1.2508e-03	+1.1083e-03
	b22	-9.9555e-04	-1.2492e-03	-1.1071e-03
F0	a11	-9.7137e-01	-9.7664e-01	-9.7692e-01
	b11	-3.1203e-04	-2.0701e-04	-2.0536e-04
	a12	-1.9888e+00	-1.9886e+00	-1.9889e+00
	a22	+9.8889e-01	+9.8870e-01	+9.8896e-01
	b12	+1.2577e-03	+1.2902e-03	+1.3241e-03
	b22	-1.2554e-03	-1.2878e-03	-1.3218e-03
IN	a13	-9.9685e-01	-9.9682e-01	-9.9680e-01
	b13	-4.7538e-03	-5.0274e-03	-5.0422e-03

Table 3. (Page 2 of 3)

Input Location	Model Coeff.	Output Location		
		C0	D0	E0
A0	a11	-9.6677e-01	-9.5389e-01	-9.7337e-01
	b11	-2.6342e-04	-3.5226e-04	-2.1035e-04
	a12	-1.9819e+00	-1.9821e+00	-1.9828e+00
	a22	+9.8200e-01	+9.8219e-01	+9.8294e-01
	b12	+1.8035e-03	+1.9070e-03	+1.8239e-03
	b22	-1.8021e-03	-1.9050e-03	-1.8220e-03
B0	a11	-9.8584e-01	-9.8812e-01	-9.8842e-01
	b11	-1.8635e-04	-1.4399e-04	-1.3287e-04
	a12	-1.9906e+00	-1.9906e+00	-1.9903e+00
	a22	+9.9063e-01	+9.9068e-01	+9.9033e-01
	b12	+1.2458e-03	+1.1651e-03	+1.2121e-03
	b22	-1.2453e-03	-1.1648e-03	-1.2114e-03
C0	a11	-9.8872e-01	-9.9141e-01	-9.9372e-01
	b11	-1.7096e-04	-1.0412e-04	-6.8385e-05
	a12	-1.9890e+00	-1.9901e+00	-1.9909e+00
	a22	+9.8905e-01	+9.9019e-01	+9.9099e-01
	b12	+1.8780e-03	+1.4618e-03	+1.3241e-03
	b22	-1.8749e-03	-1.4609e-03	-1.3228e-03
D0	a11	-9.7863e-01	-9.7193e-01	-9.7223e-01
	b11	-2.0212e-04	-3.4191e-04	-2.6377e-04
	a12	-1.9872e+00	-1.9861e+00	-1.9881e+00
	a22	+9.8724e-01	+9.8616e-01	+9.8817e-01
	b12	+1.4348e-03	+1.8679e-03	+1.4420e-03
	b22	-1.4315e-03	-1.8628e-03	-1.4390e-03
E0	a11	-9.8549e-01	-9.8323e-01	-9.7887e-01
	b11	-1.1276e-04	-1.5716e-04	-2.5175e-04
	a12	-1.9855e+00	-1.9879e+00	-1.9869e+00
	a22	+9.8562e-01	+9.8796e-01	+9.8694e-01
	b12	+1.4856e-03	+1.3018e-03	+1.6364e-03
	b22	-1.4826e-03	-1.2997e-03	-1.6331e-03
F0	a11	-9.8002e-01	-9.7775e-01	-9.6952e-01
	b11	-1.8505e-04	-2.1565e-04	-3.2854e-04
	a12	-1.9886e+00	-1.9890e+00	-1.9876e+00
	a22	+9.8872e-01	+9.8905e-01	+9.8767e-01
	b12	+1.3772e-03	+1.3186e-03	+1.5314e-03
	b22	-1.3745e-03	-1.3160e-03	-1.5277e-03
IN	a13	-9.9688e-01	-9.9683e-01	-9.9685e-01
	b13	-4.9585e-03	-4.9527e-03	-4.9433e-03

Table 3. (Page 3 of 3)

Input Location	Model Coeff.	Output Location		
		EA	F0	SY
A0	a11	-9.7532e-01	-9.8116e-01	-9.8461e-01
	b11	-1.9532e-04	-1.8664e-04	-1.6145e-04
	a12	-1.9832e+00	-1.9847e+00	-1.9837e+00
	a22	+9.8334e-01	+9.8483e-01	+9.8383e-01
	b12	+1.8381e-03	+1.7367e-03	+1.8680e-03
	b22	-1.8362e-03	-1.7337e-03	-1.8646e-03
B0	a11	-9.8842e-01	-9.8707e-01	-9.8357e-01
	b11	-1.3288e-04	-1.4972e-04	-2.2655e-04
	a12	-1.9910e+00	-1.9910e+00	-1.9898e+00
	a22	+9.9103e-01	+9.9106e-01	+9.8990e-01
	b12	+1.1427e-03	+1.1363e-03	+1.3932e-03
	b22	-1.1424e-03	-1.1362e-03	-1.3921e-03
C0	a11	-9.9376e-01	-9.9333e-01	-9.9210e-01
	b11	-6.8223e-05	-6.7948e-05	-9.2148e-05
	a12	-1.9908e+00	-1.9911e+00	-1.9903e+00
	a22	+9.9083e-01	+9.9120e-01	+9.9035e-01
	b12	+1.3221e-03	+1.2196e-03	+1.4354e-03
	b22	-1.3208e-03	-1.2187e-03	-1.4340e-03
D0	a11	-9.7081e-01	-9.7210e-01	-9.7333e-01
	b11	-2.7716e-04	-2.2987e-04	-1.9685e-04
	a12	-1.9888e+00	-1.9885e+00	-1.9888e+00
	a22	+9.8886e-01	+9.8860e-01	+9.8891e-01
	b12	+1.3149e-03	+1.2228e-03	+1.2504e-03
	b22	-1.3125e-03	-1.2207e-03	-1.2483e-03
E0	a11	-9.7901e-01	-9.8295e-01	-9.8774e-01
	b11	-2.4655e-04	-1.5197e-04	-8.3560e-05
	a12	-1.9864e+00	-1.9878e+00	-1.9882e+00
	a22	+9.8645e-01	+9.8786e-01	+9.8825e-01
	b12	+1.4845e-03	+1.2104e-03	+1.1623e-03
	b22	-1.4816e-03	-1.2091e-03	-1.1608e-03
F0	a11	-9.6942e-01	-9.6283e-01	-9.7532e-01
	b11	-3.3204e-04	-4.7710e-04	-2.3391e-04
	a12	-1.9888e+00	-1.9869e+00	-1.9881e+00
	a22	+9.8892e-01	+9.8703e-01	+9.8819e-01
	b12	+1.4081e-03	+1.8192e-03	+1.3814e-03
	b22	-1.4053e-03	-1.8142e-03	-1.3784e-03
IN	a13	-9.9686e-01	-9.9685e-01	-9.9678e-01
	b13	-4.8990e-03	-4.8749e-03	-5.0726e-03

response curves shown in Figures 102 through 114 for each investigated input.

From these figures, differences in response to compressor flow between the various ring locations are seen to be much smaller than those observed between models based on individual data sets. Averaging coefficients among similar data sets has a significant affect in producing models that give similar results for input flows at each location. Trends do appear, however, when studying the responses in location order. As the flow input location moves away from B0 and C0 in either direction, the response of both P1 and P2 decreases slightly. Since the change is small and is not based on many data sets, more testing would ideally be performed before concluding that this trend is repeatable. Still, the agreement between averaged models of pressure response at various compressors locations, and the slight but consistent trend with location, suggest the averaged models represent the system response reasonably well.

4.3.2 Model Validation. The most convincing test of a model's validity is to compare simulation with independent data. To this end, six data sets not used in the modeling are used to investigate how well the model can predict pressure responses. Some of the validation data sets challenge the limits of the models since their input flow steps are either higher or lower than for the data from which the model was derived. Selected results, from four data sets with compressor flow input and from two data sets with inventory flow input, are presented. In each case, relevant flows and sums of flows are also shown. Note that flow plots in all validation cases are shown with the same

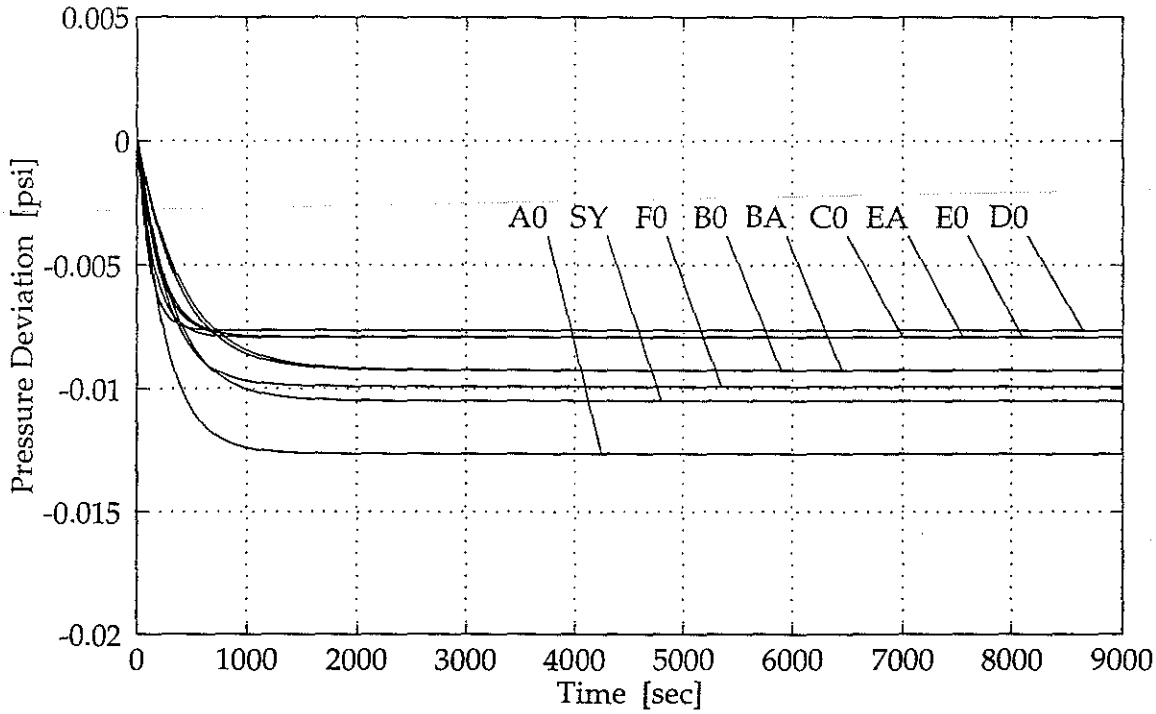

Figure 102. Averaged P1 Model for Unit Step in A0 F2

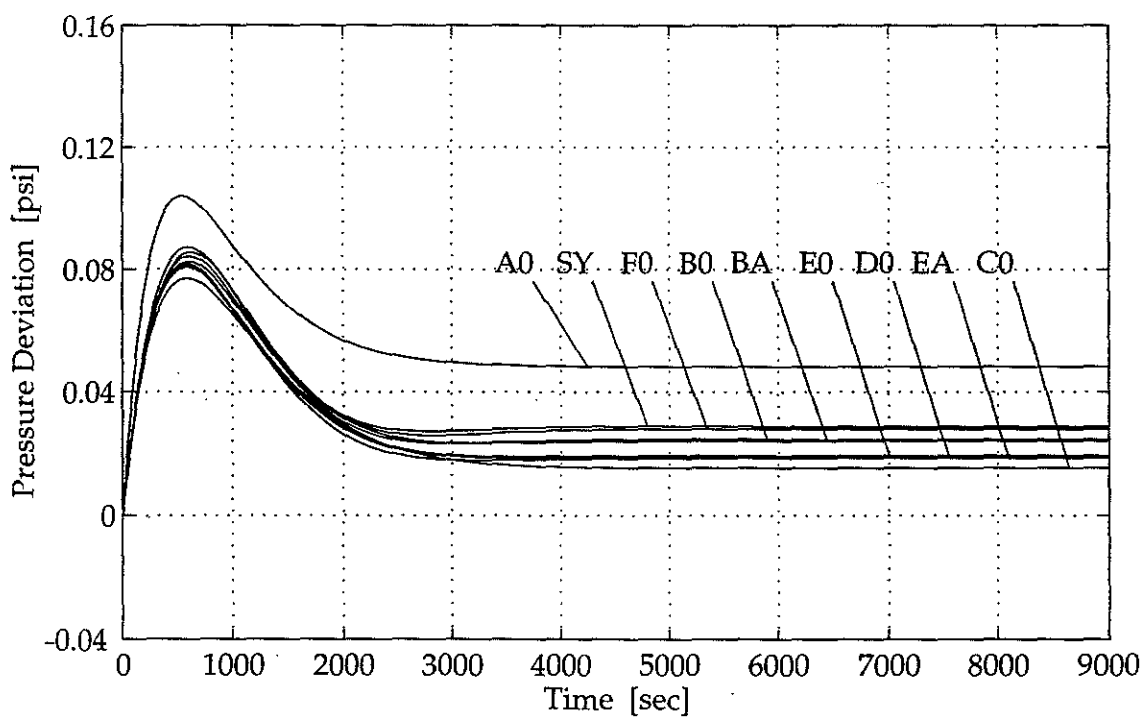

Figure 103. Averaged P2 Model for Unit Step in A0 F2

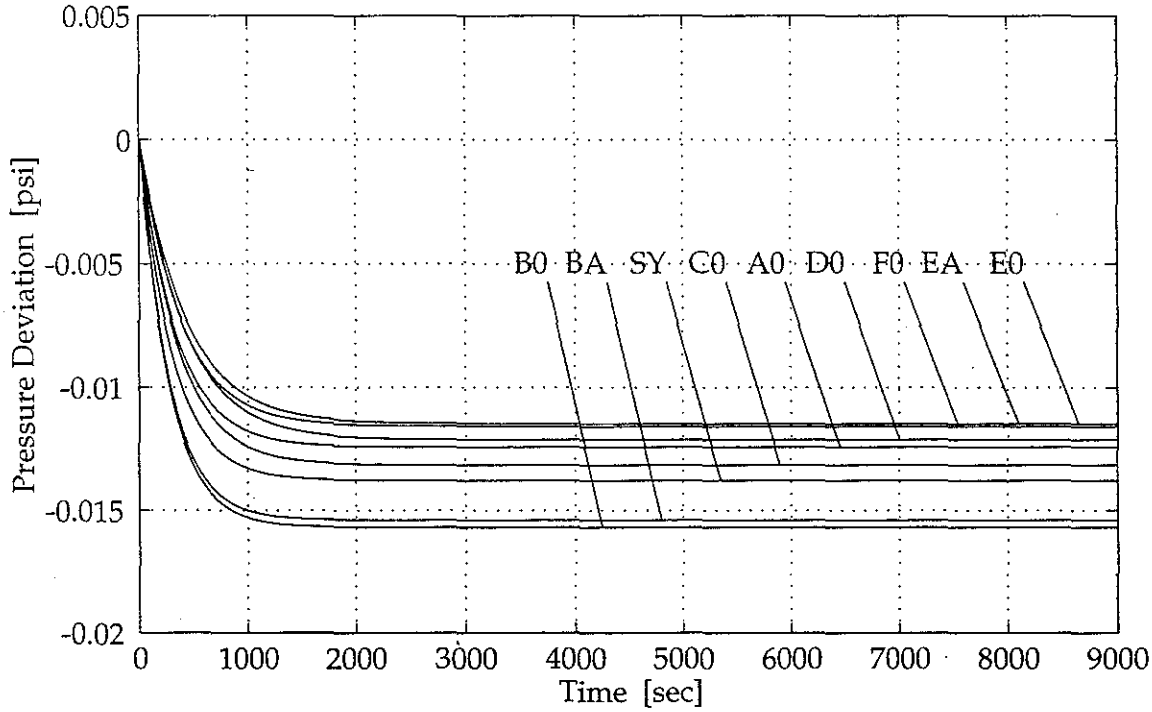

Figure 104. Averaged P1 Model for Unit Step in B0 F2

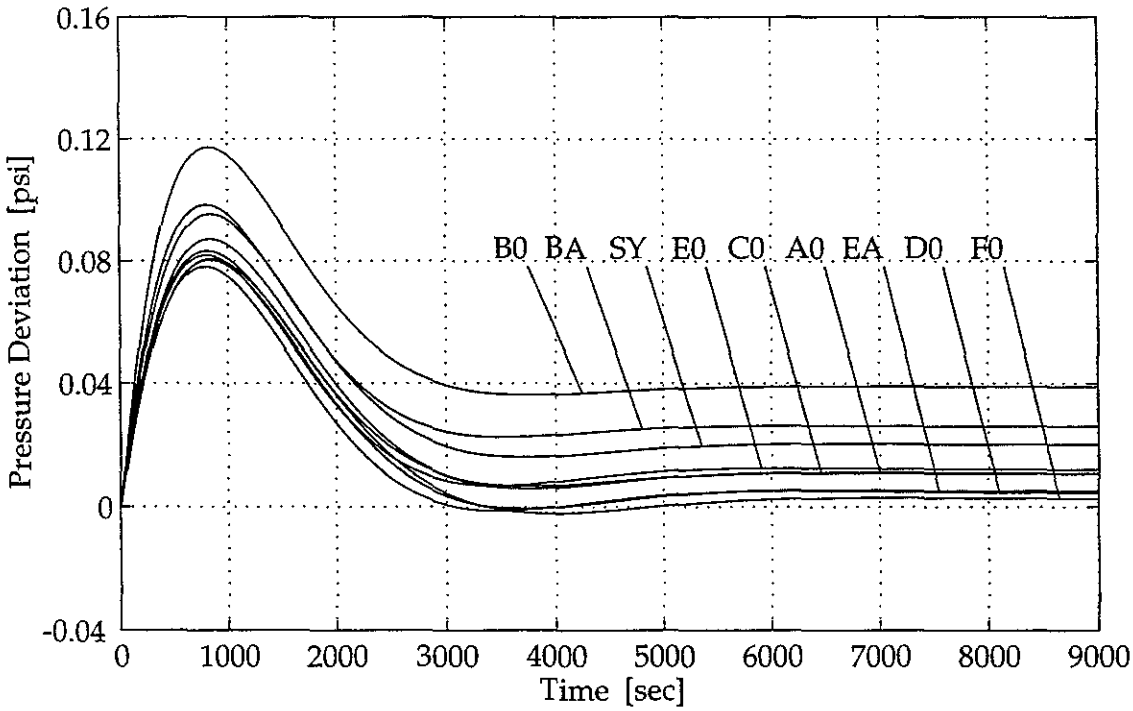

Figure 105. Averaged P2 Model for Unit Step in B0 F2

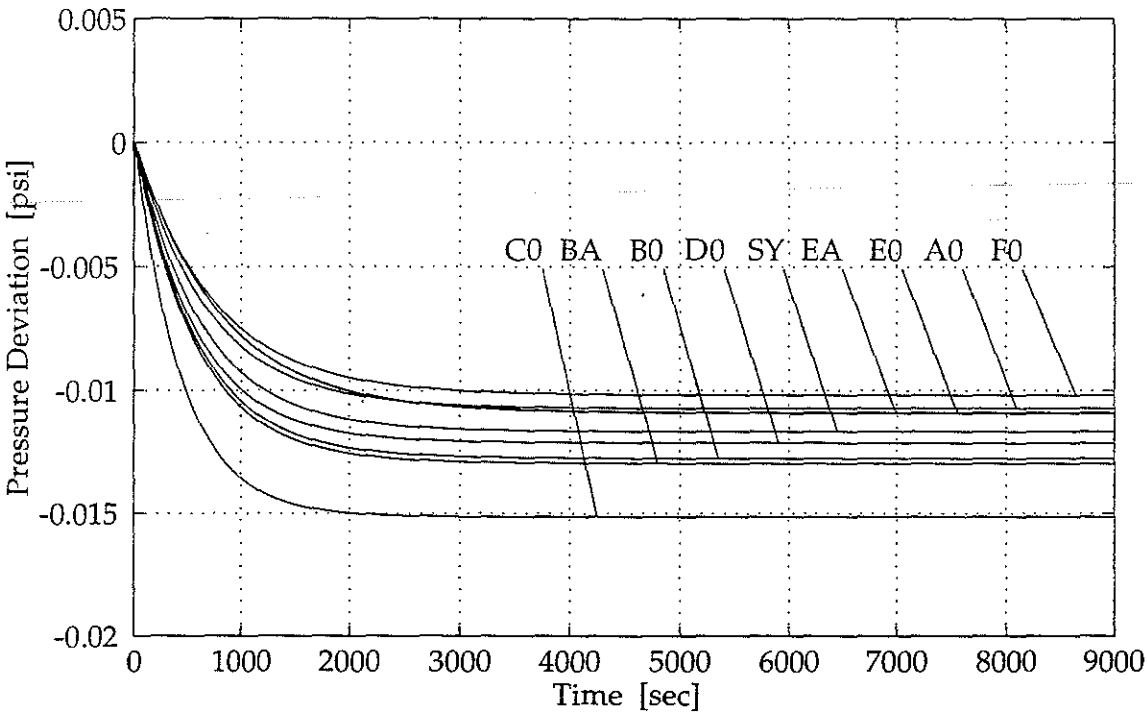

Figure 106. Averaged P1 Model for Unit Step in C0 F2

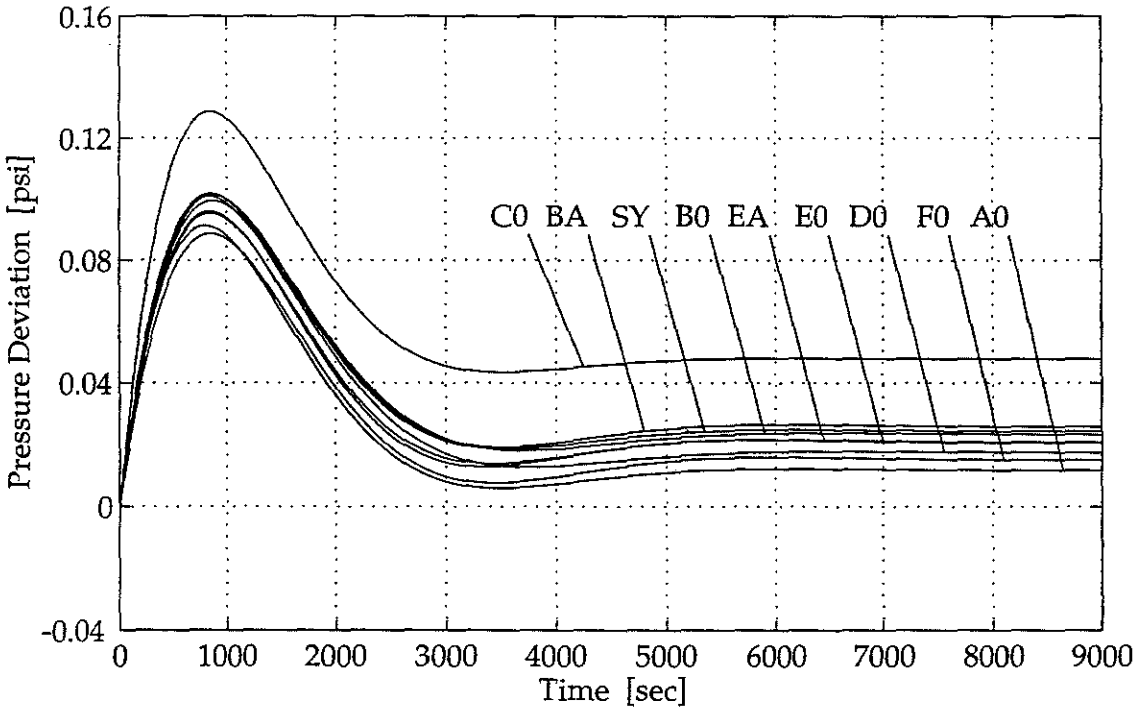

Figure 107. Averaged P2 Model for Unit Step in C0 F2

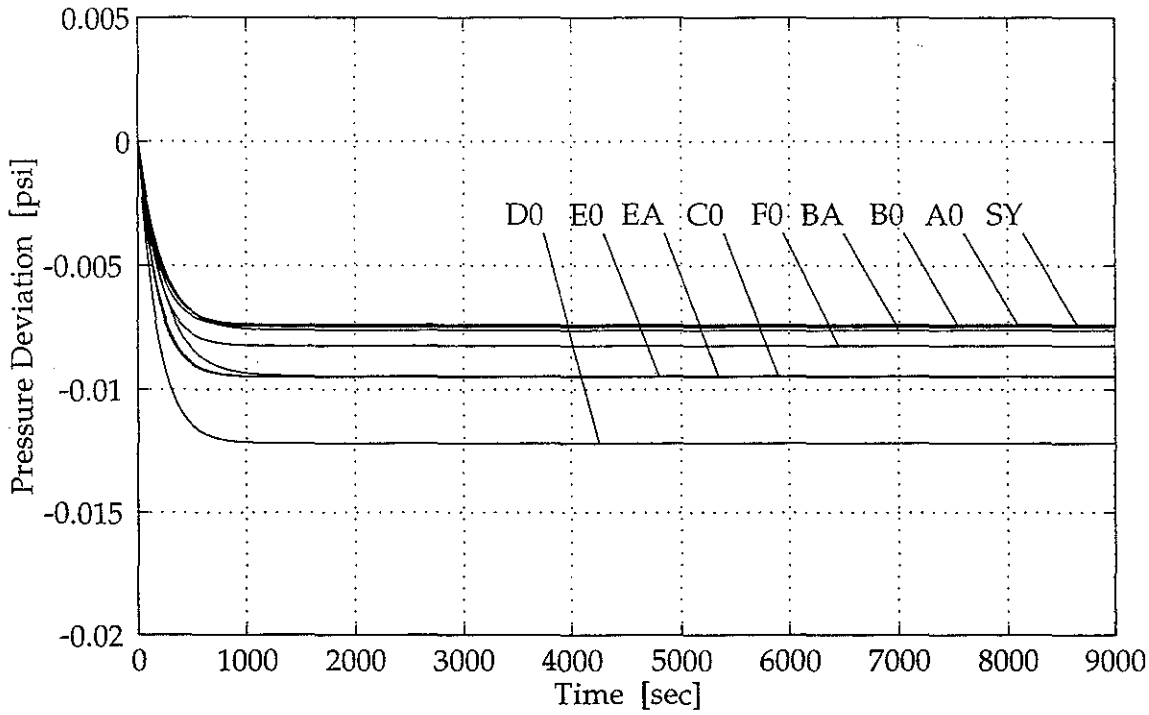

Figure 108. Averaged P1 Model for Unit Step in D0 F2

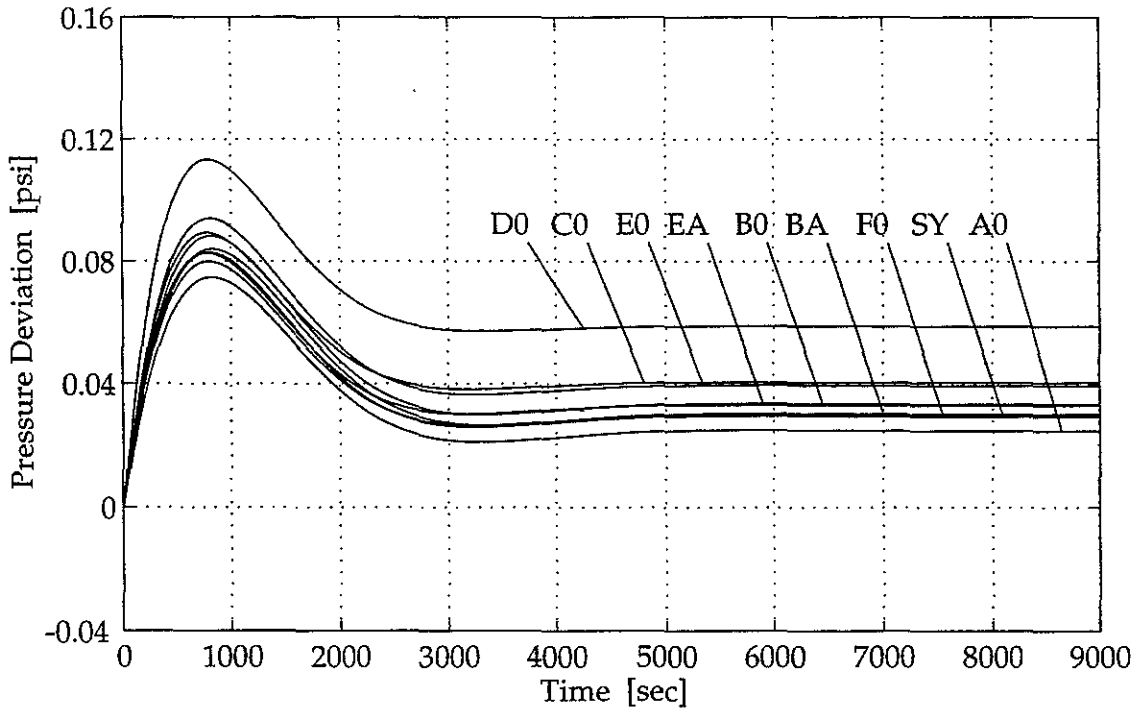

Figure 109. Averaged P2 Model for Unit Step in D0 F2

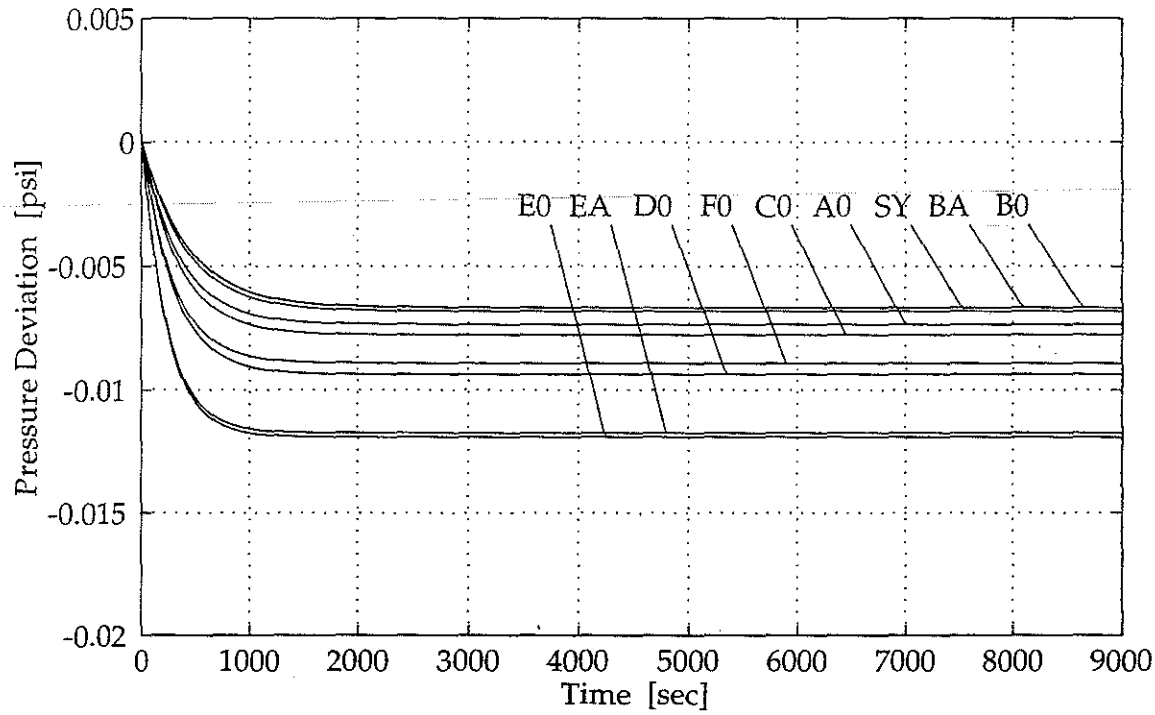

Figure 110. Averaged P1 Model for Unit Step in E0 F2

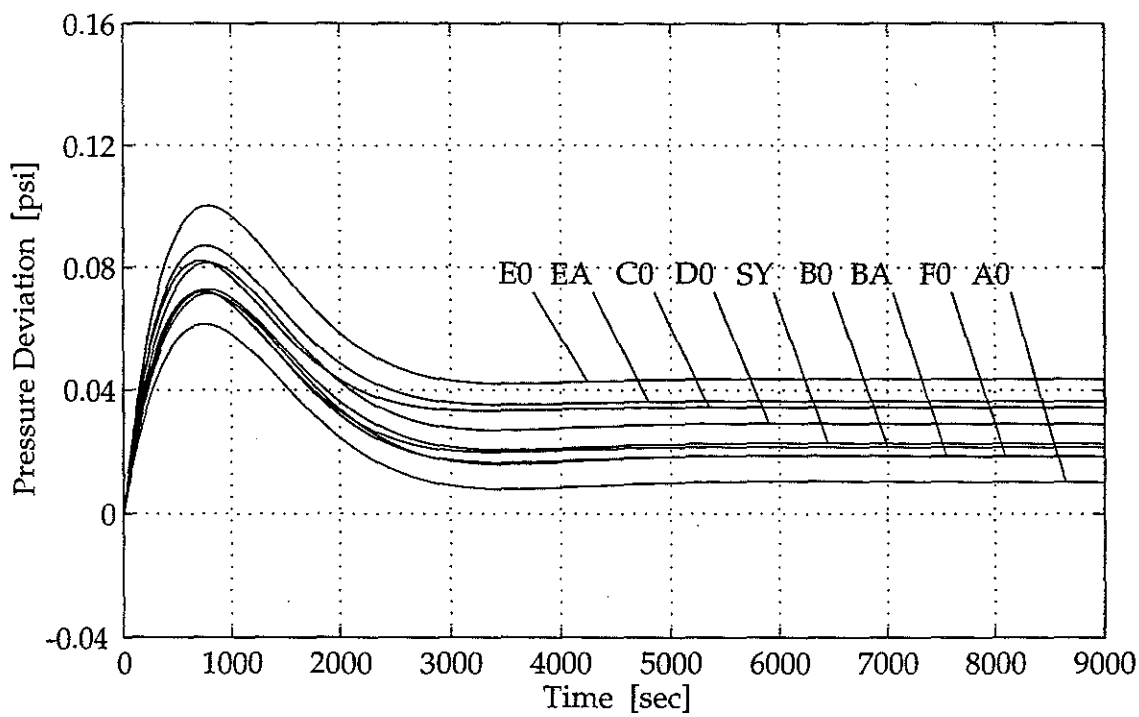

Figure 111. Averaged P2 Model for Unit Step in E0 F2

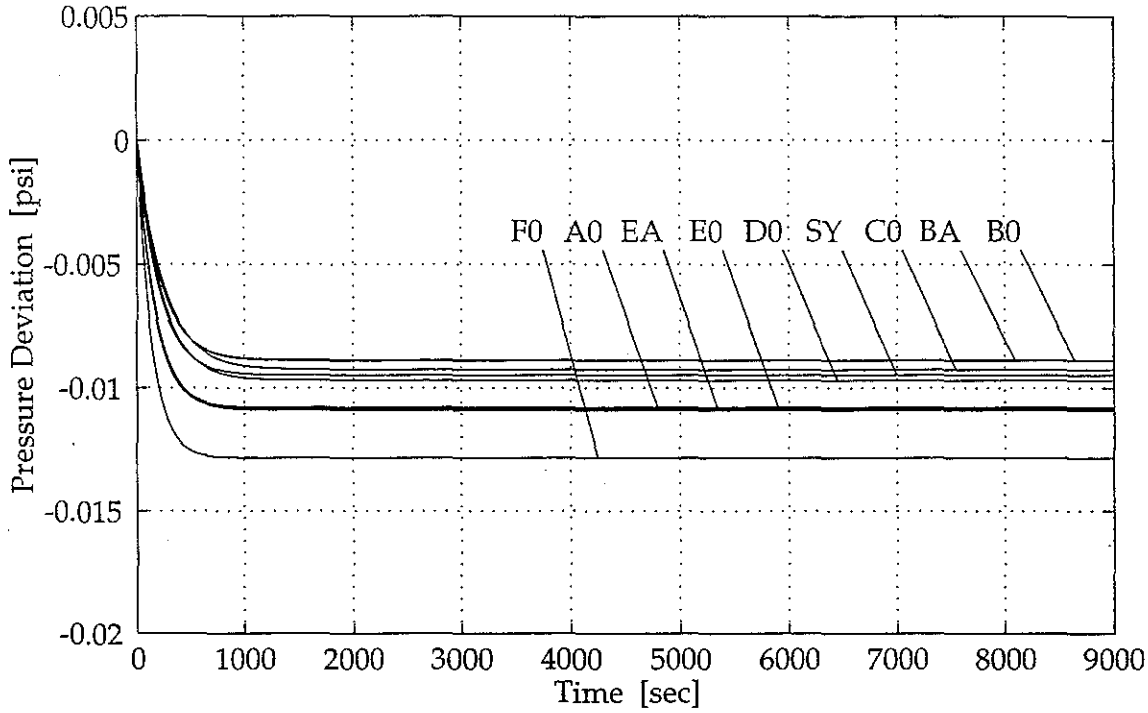

Figure 112. Averaged P1 Model for Unit Step in F0 F2

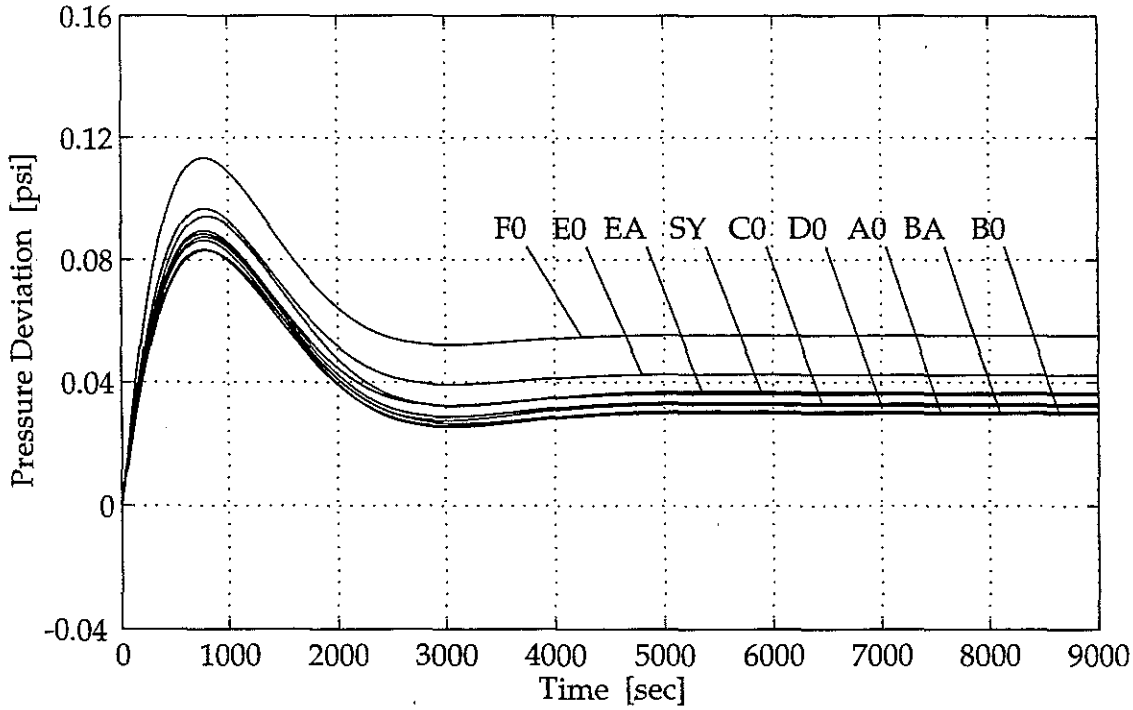

Figure 113. Averaged P2 Model for Unit Step in F0 F2

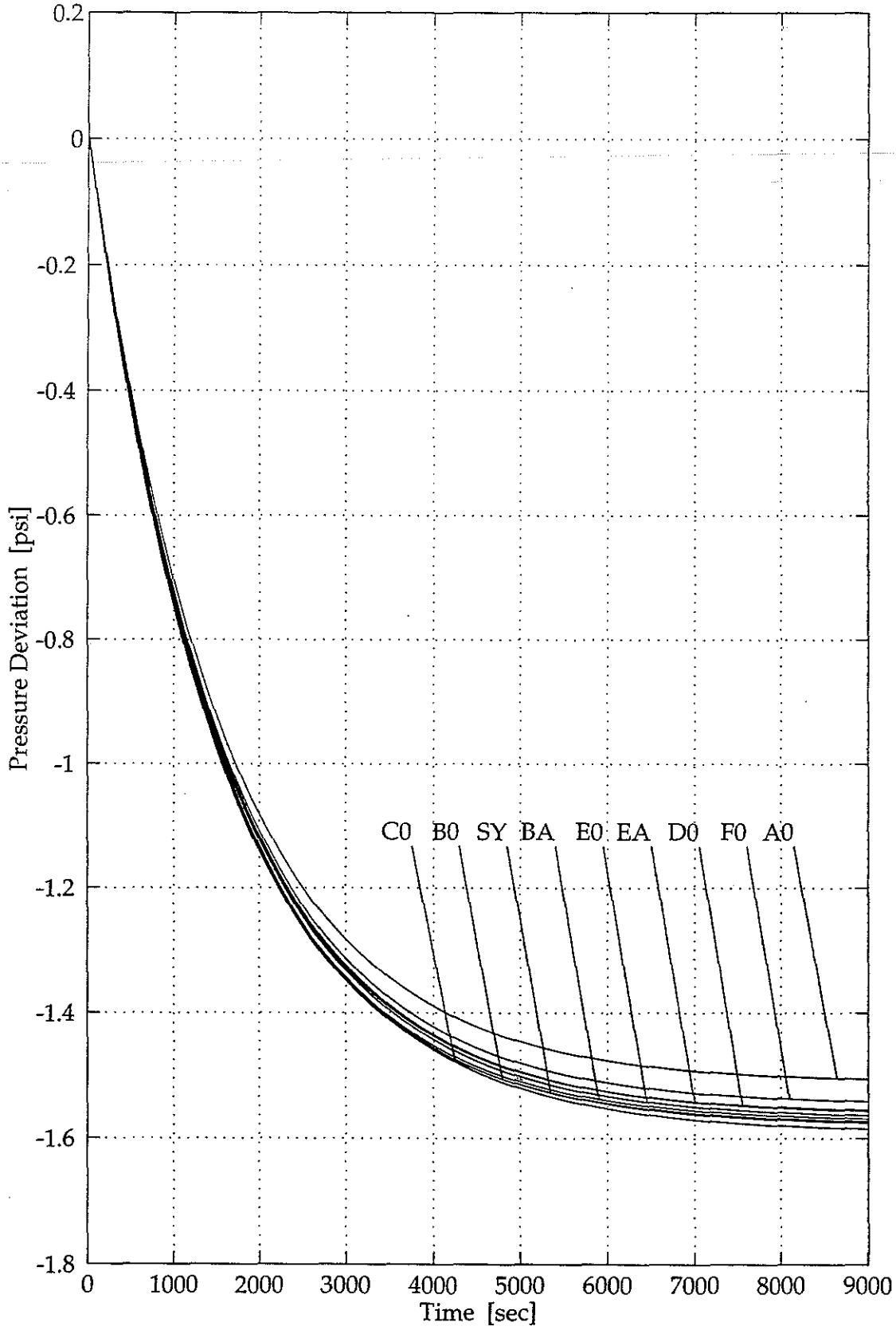

Figure 114. Averaged P2 Model for Unit Step in Inventory Flow

y-axis scale to illustrate the range of input simulated. Due to the much larger response of the system to inventory flow changes compared to those for compressor flow changes, the pressure output scales differ between plots for the two input types. The sse between data and simulation is calculated and the results are shown in Table 2, along with parameters used by the programs SimF2.m and SimFC.m. Code for these programs is found in the Appendix.

4.3.3 Validation for Compressor Flow Input. The flow from which input steps were obtained, as well as the validation results for data set 23 are shown in Figures 115 through 120. For the loader valve change from 92.3% to 100%, a step value of 16 g/s is calculated using the generic correlation in equation (2.6). Figure 115 indicated something less that 17 g/s giving an idea of the inseparable error in the correlation and in the measurement. Figures 117 through 120 show reasonable agreement between data and simulation for the B0 location and for E0 at the opposite side of the ring. For P1, the differences in simulated and actual final values are on the order of 20%, and the speed of response is well represented. Data for P2 is too short to compare final values, but the peak values agree within about 20%.

Data set 24 provides the 55 g/s step input predicted from turning on a compressor and loading it fully. Plots associated with this test are shown in Figures 121 through 126. The pause appearing in the flow step of Figure 121 is due to a controlled delay that separates motor turn-on (at which time the loader valve is at a 20% position) and a loader valve movement command (in this instance to 100%). The significance of this data set is in showing linearity of the system. Figures for pressure indicates that the model can do a

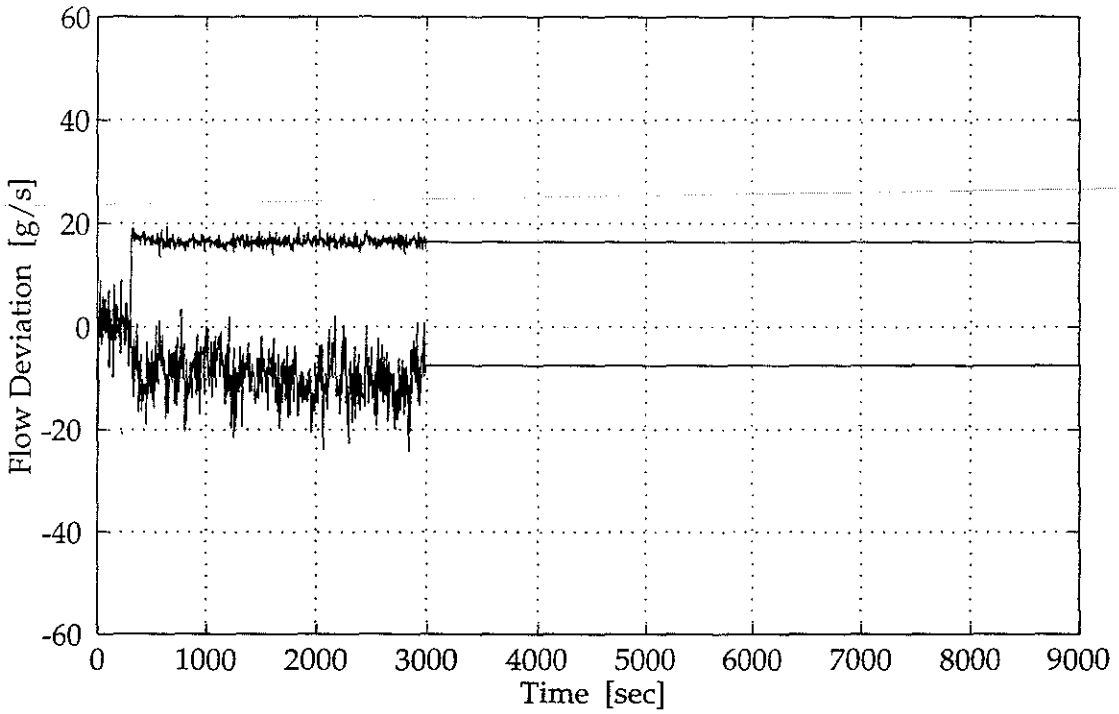

Figure 115. B0 F2 Flow and Secondary Flow from Data Set 23

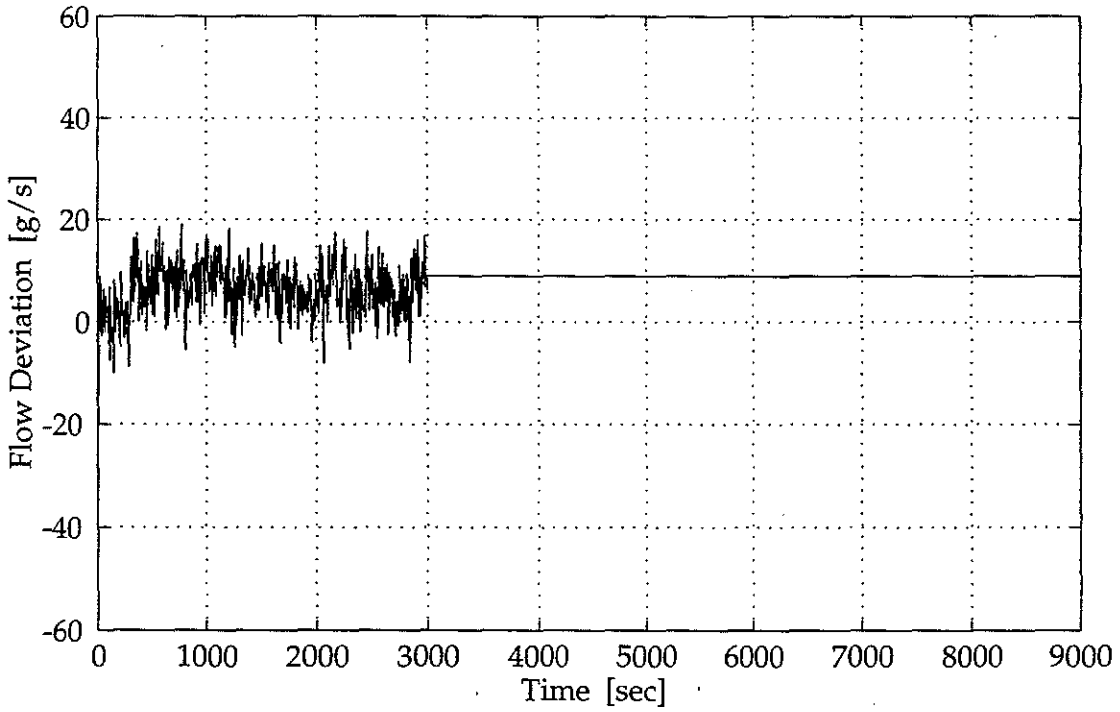

Figure 116. Sum of B0 F2 Flow and Secondary Flow from Data Set 23

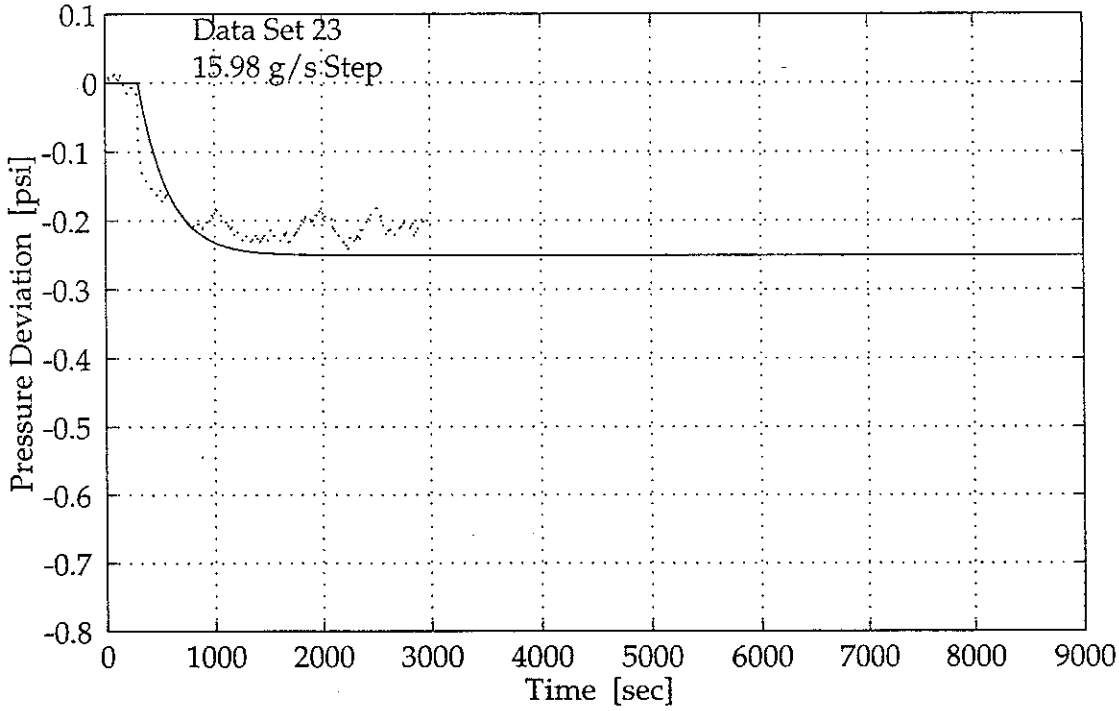

Figure 117. B0 P1 Data and Simulation for Step in B0 F2

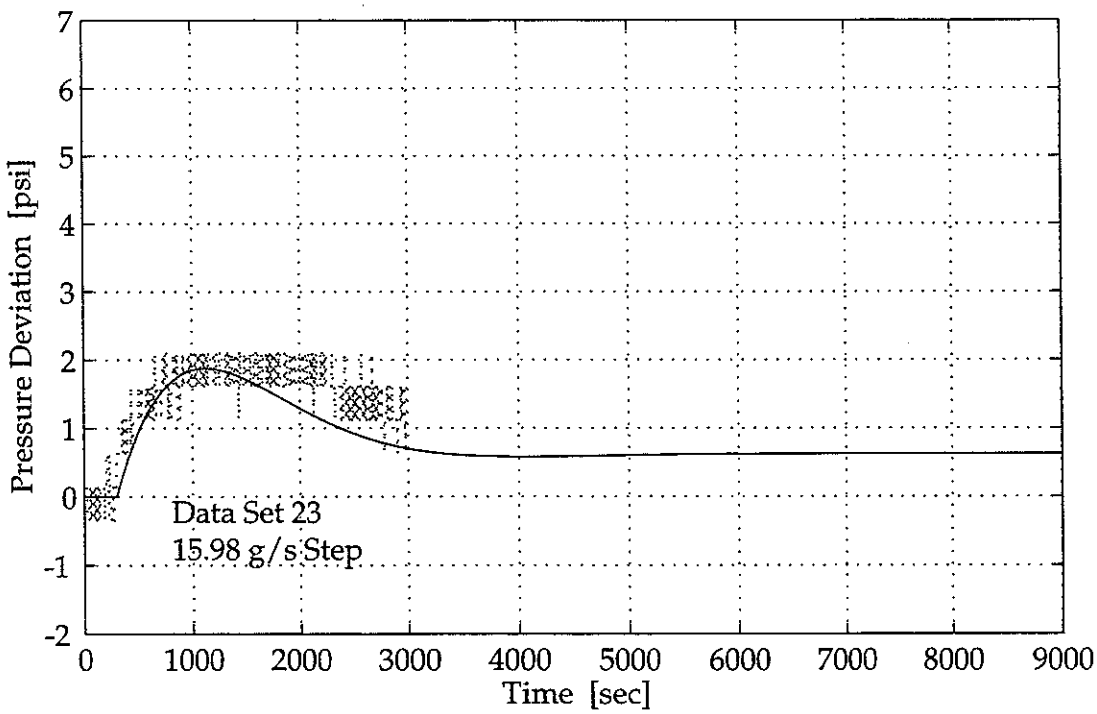

Figure 118. B0 P2 Data and Simulation for Step in B0 F2

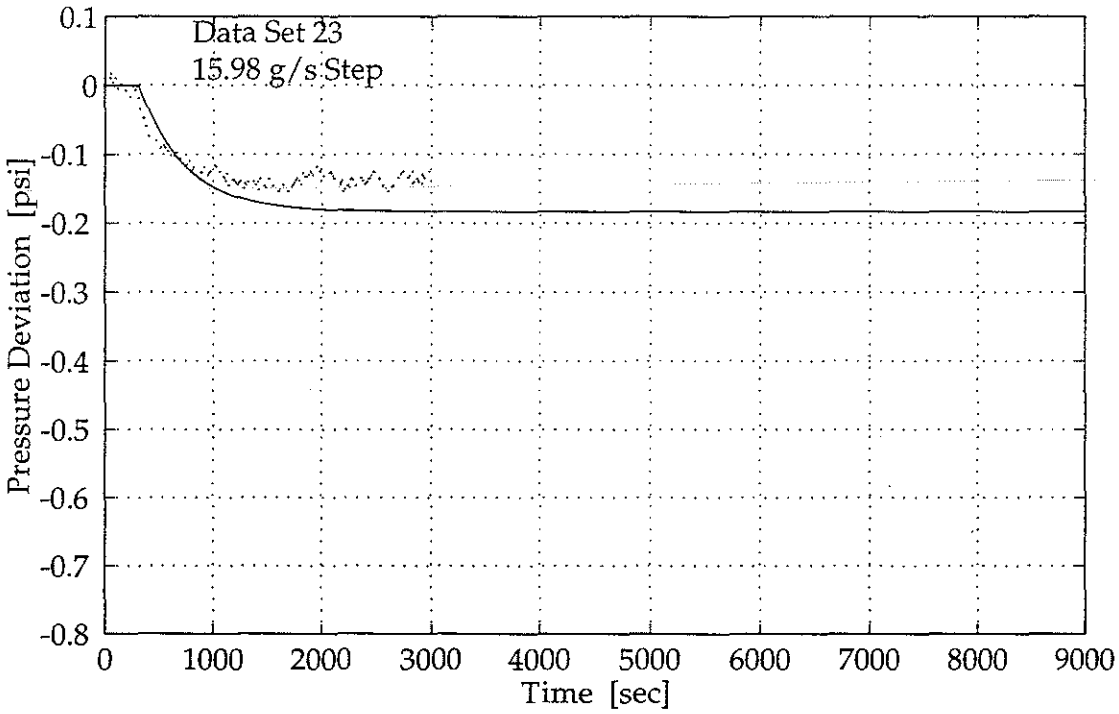

Figure 119. E0 P1 Data and Simulation for Step in B0 F2

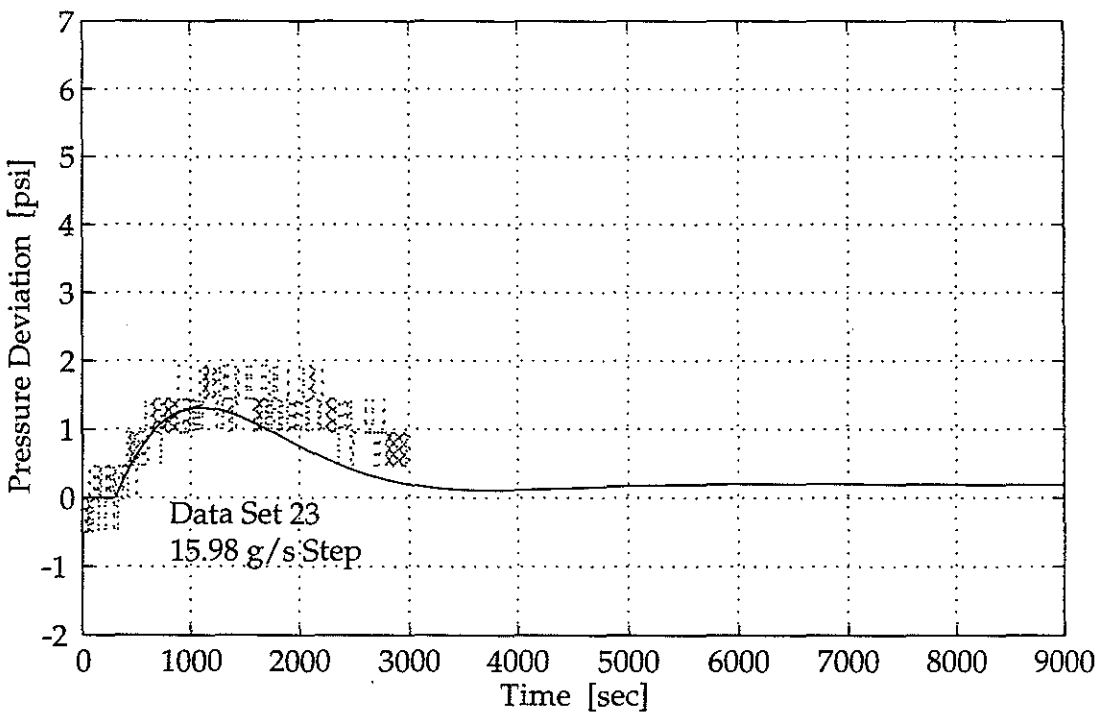

Figure 120. E0 P2 Data and Simulation for Step in B0 F2

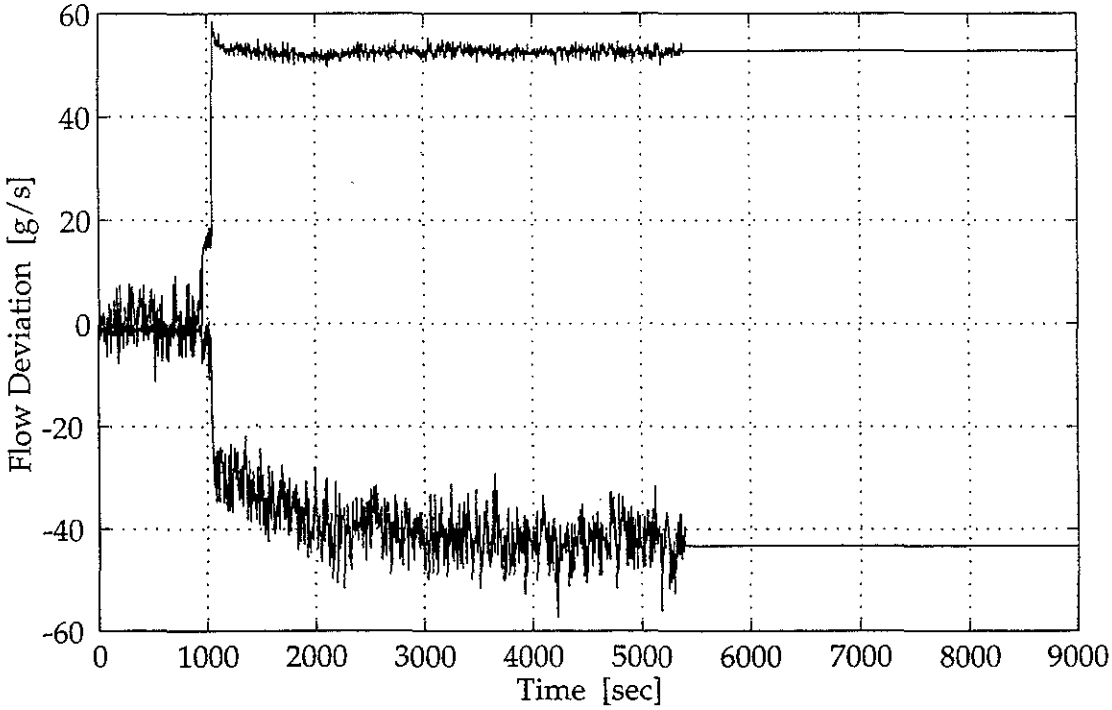

Figure 121. E0 F2 Flow and Secondary Flow from Data Set 24

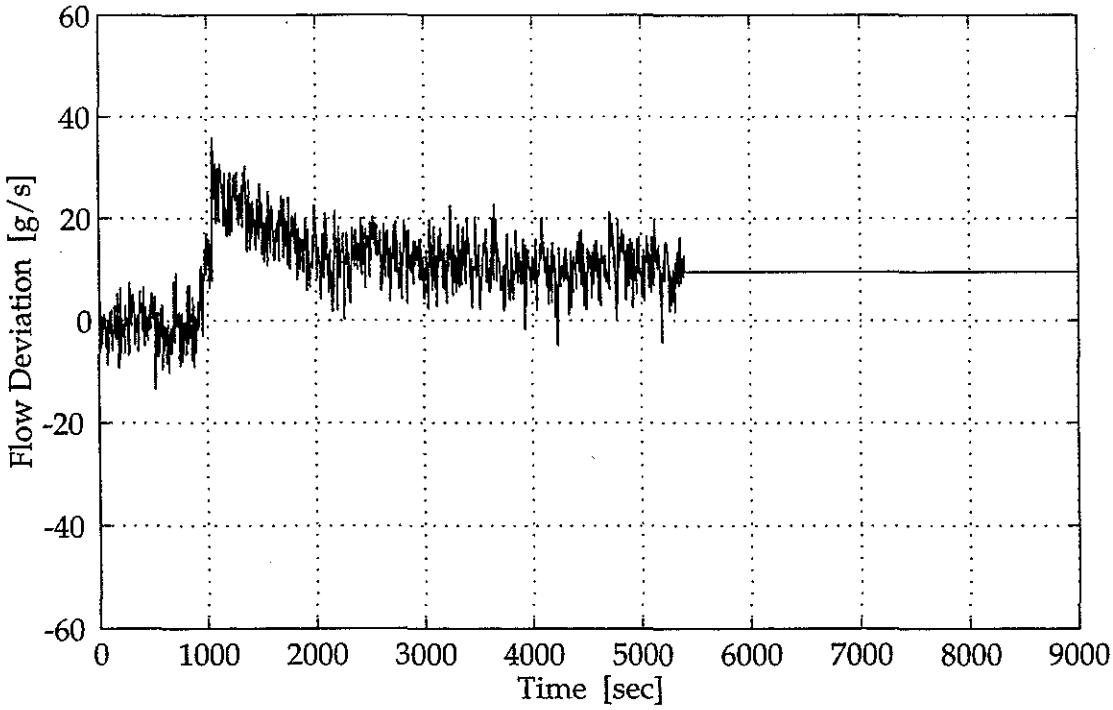

Figure 122. Sum of E0 F2 Flow and Secondary Flow from Data Set 24

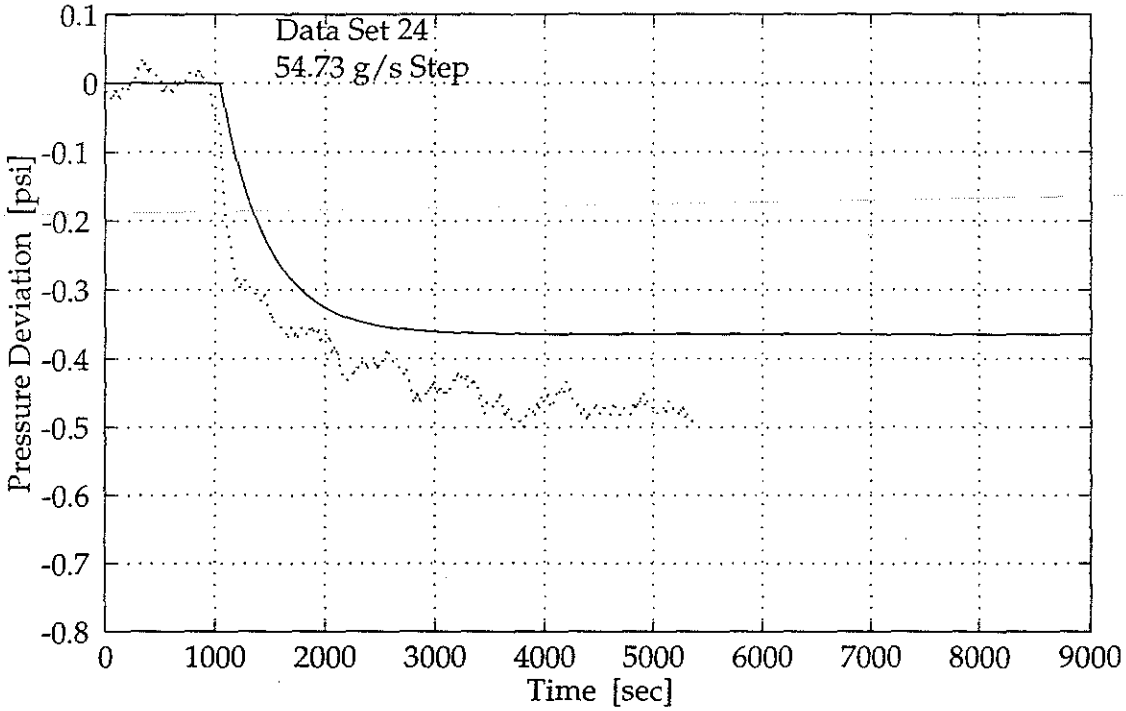

Figure 123. B0 P1 Data and Simulation for Step in E0 F2

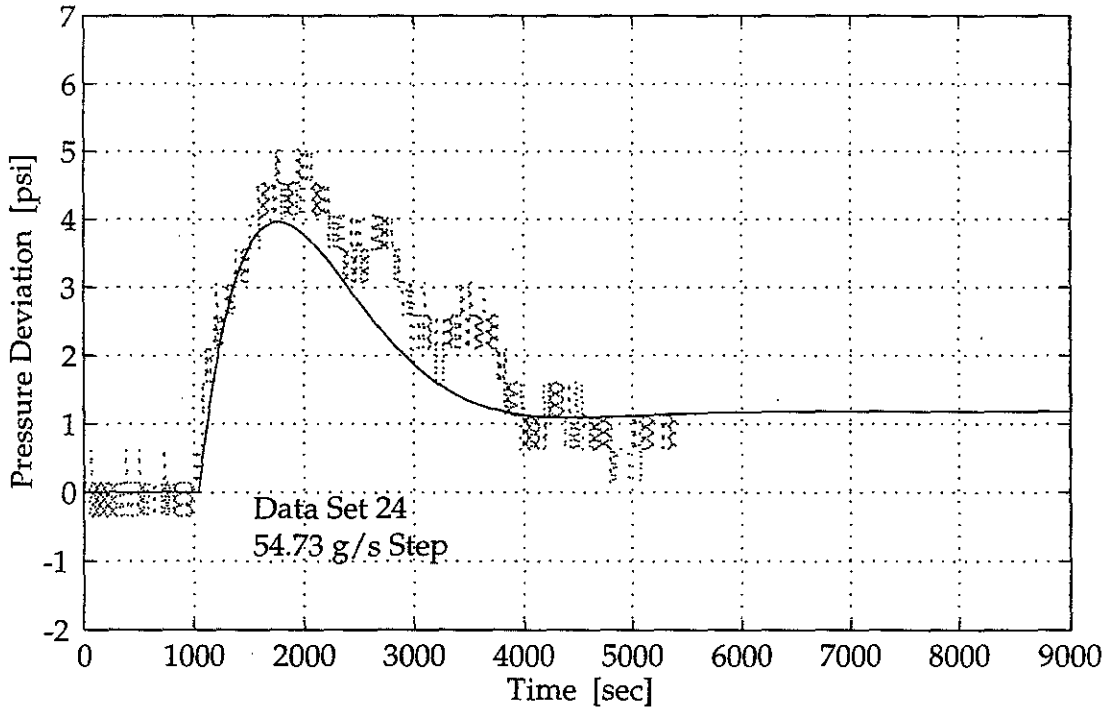

Figure 124. B0 P2 Data and Simulation for Step in E0 F2

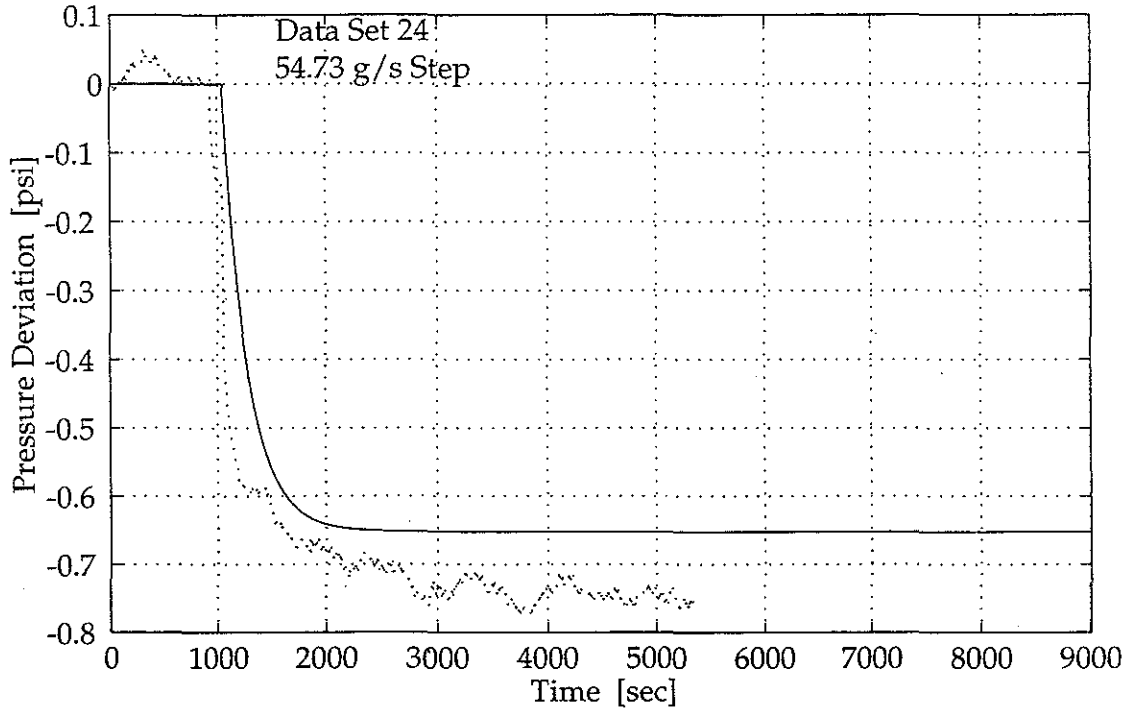

Figure 125. E0 P1 Data and Simulation for Step in E0 F2

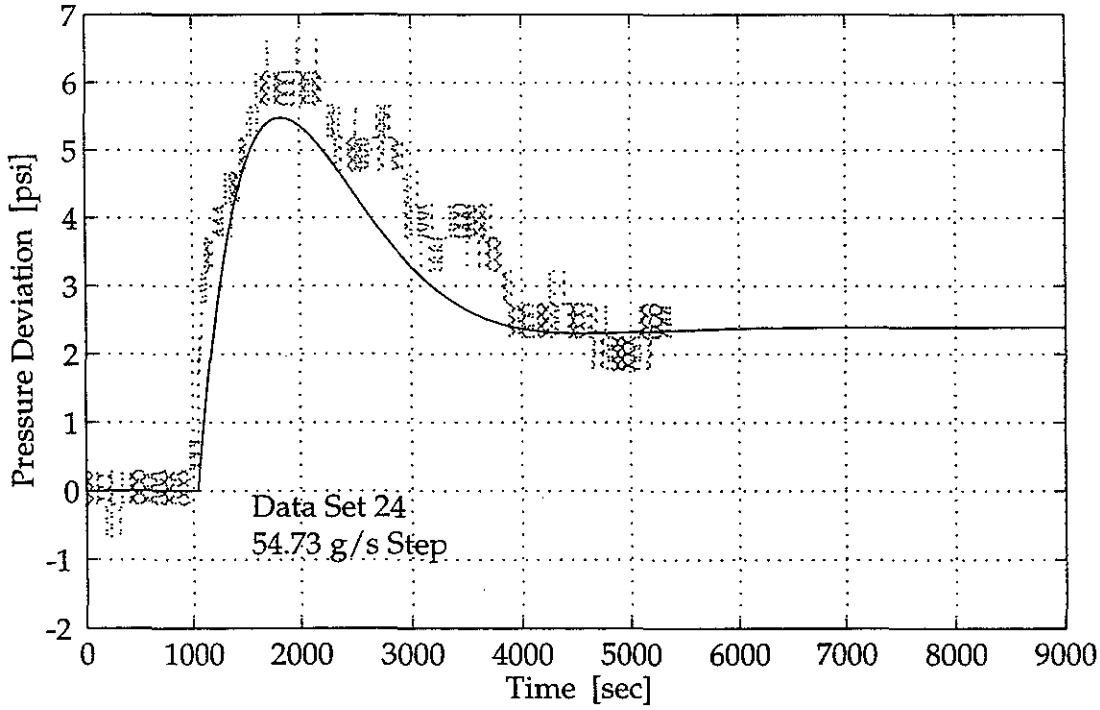

Figure 126. E0 P2 Data and Simulation for Step in E0 F2

good job of simulating the response to the largest flow step that is typically encountered. Since compressors are usually turned on one at a time, only large scale failures, such as loss of power to a ring location, create bigger steps. Again, results are presented for the locations for which the largest difference in system response is predicted, that is, at E0 where the step occurs and at B0 on the opposite side of the ring.

Turning off a compressor results in a system response like that of data set 25. Figures 127 through 132 show this data and corresponding simulations. The P1 data and model agree as well as they do for data set 24, but there is a difference. In set 24 with an increasing flow the model tends to respond less than the actual data, and in set 25 with a decreasing flow the model tends to respond more. This could represent a slight nonlinearity in the system, inaccuracy in the model due to insufficient data set averaging, or unknown disturbances in either of these two data sets. With respect to P2, the model deficiency is clearly one of poor performance after 3000 seconds. As seen in Chapter III where data set 3 was analyzed, discharge pressure comes to rest very close to its starting point with little net effect at any location. Due to the greater duration of this test, it is subject to long term drifts which can't be subtracted easily. What is significant is that P2 does attain a steady state value, something that until this point was not verified in data. The P2 model does represent the system behavior, if not the final value, and no oscillation is observed after 4000 seconds.

The fact that only one hump exists in the P2 response is further confirmed by data set 26. Figures 133 through 138 give the flow and pressure

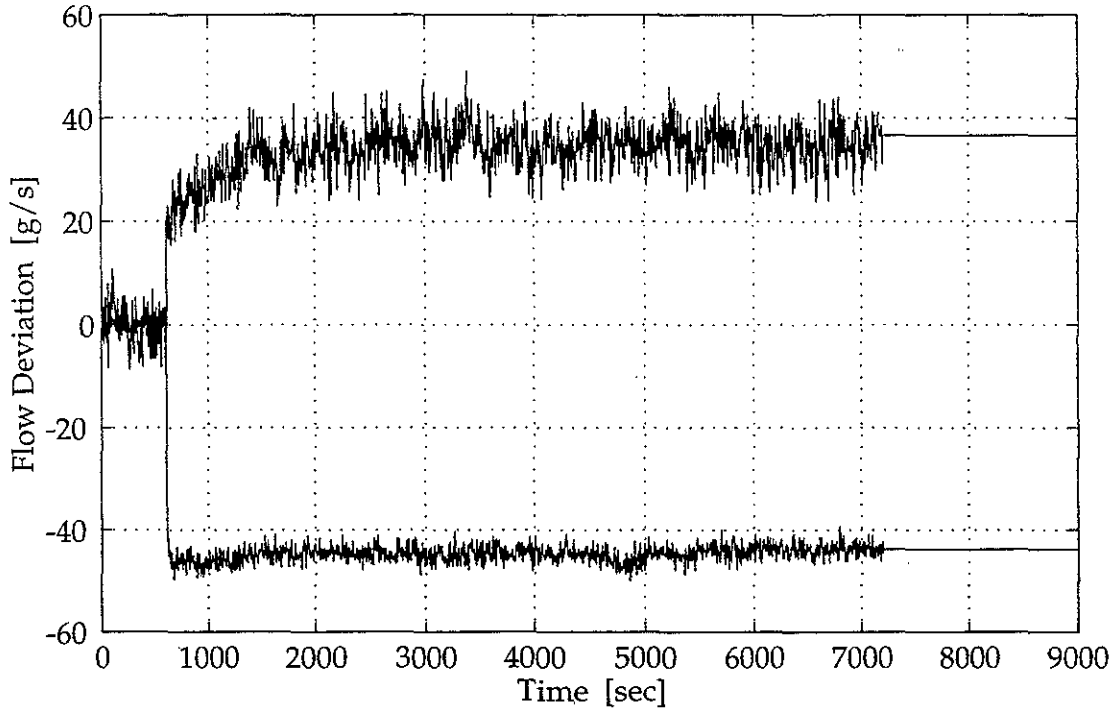

Figure 127. F0 F2 Flow and Secondary Flow from Data Set 25

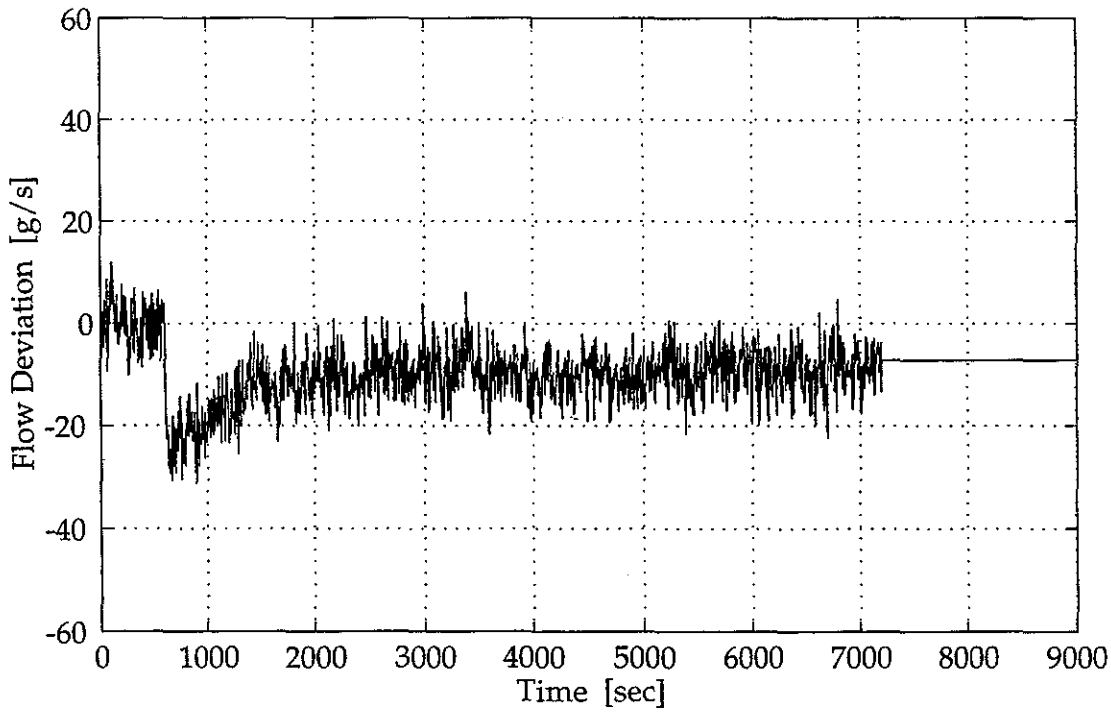

Figure 128. Sum of F0 F2 Flow and Secondary Flow from Data Set 25

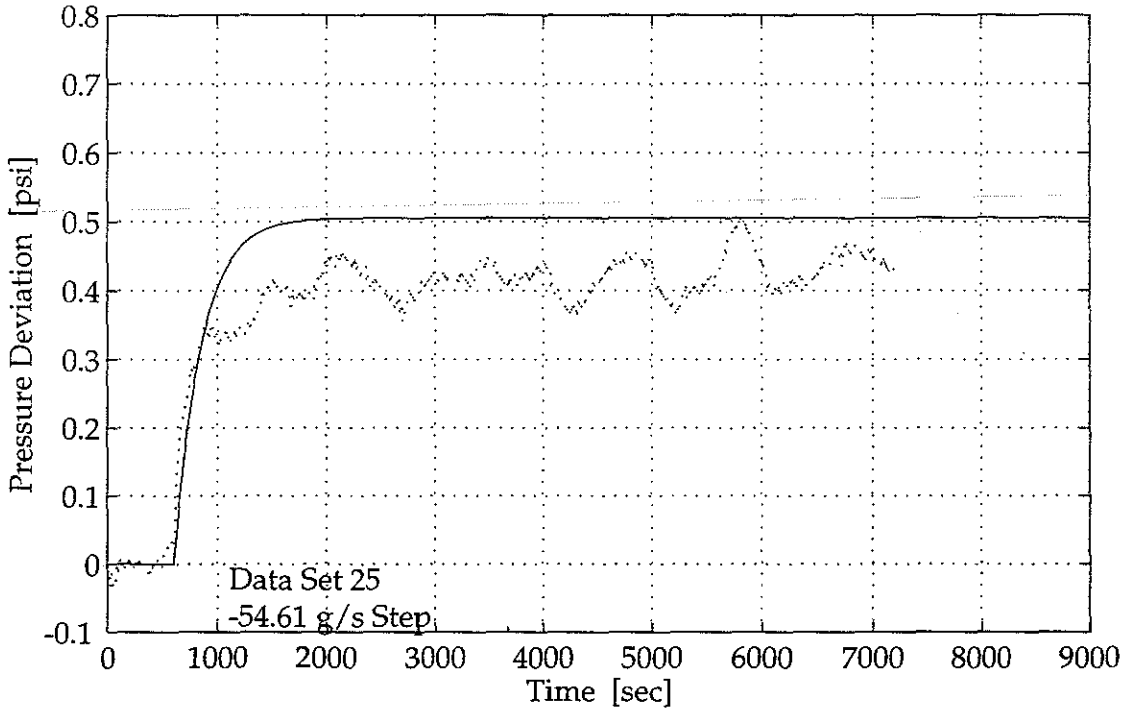

Figure 129. C0 P1 Data and Simulation for Step in F0 F2

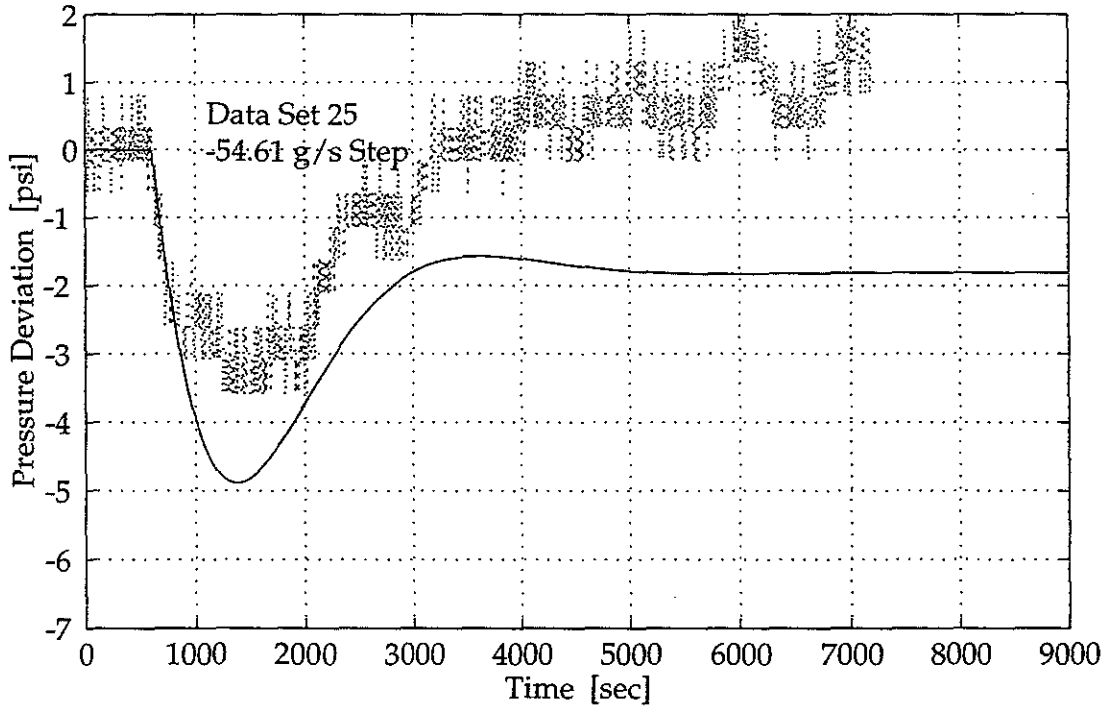

Figure 130. C0 P2 Data and Simulation for Step in F0 F2

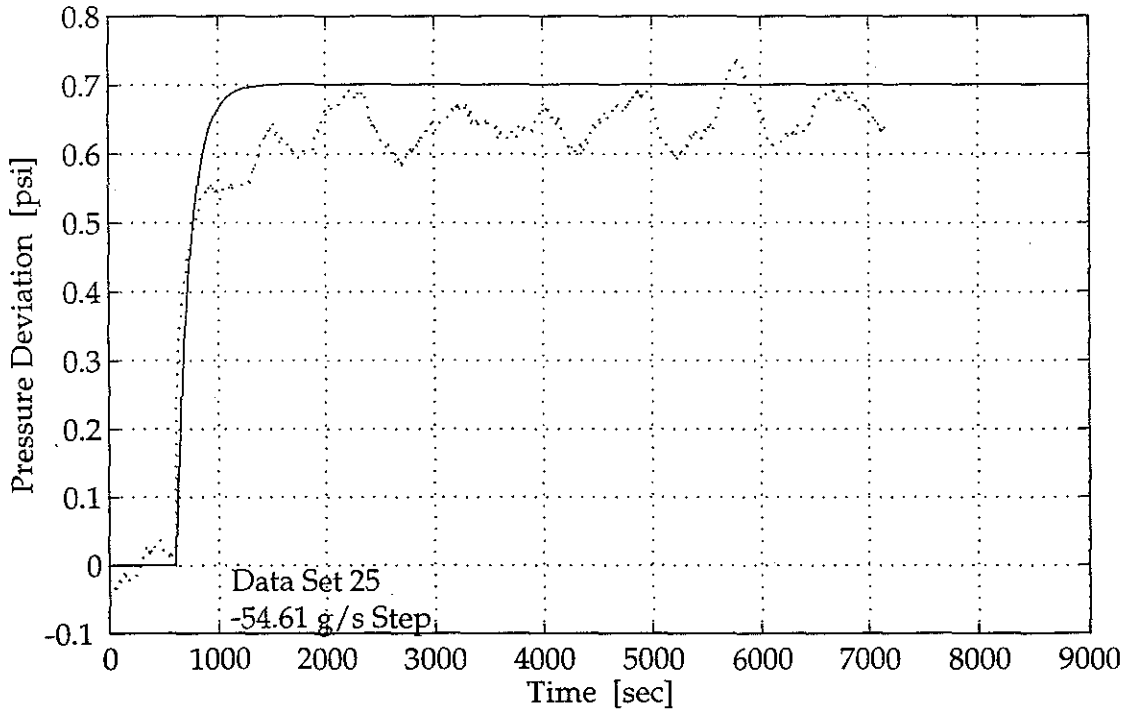

Figure 131. F0 P1 Data and Simulation for Step in F0 F2

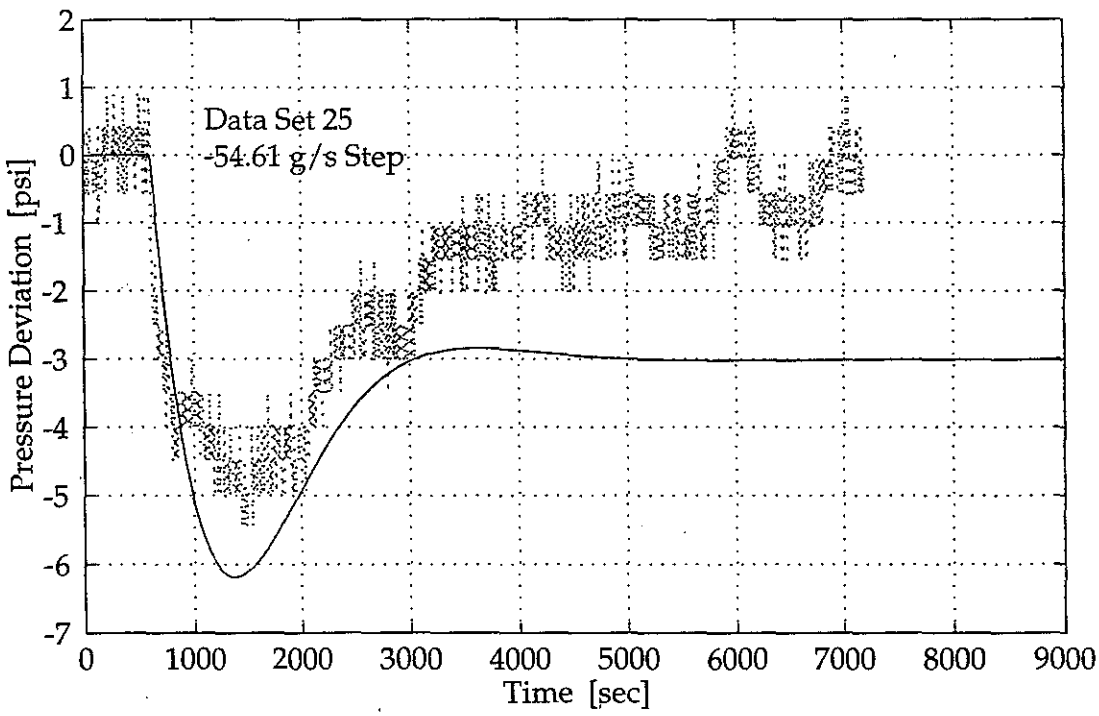

Figure 132. F0 P2 Data and Simulation for Step in F0 F2

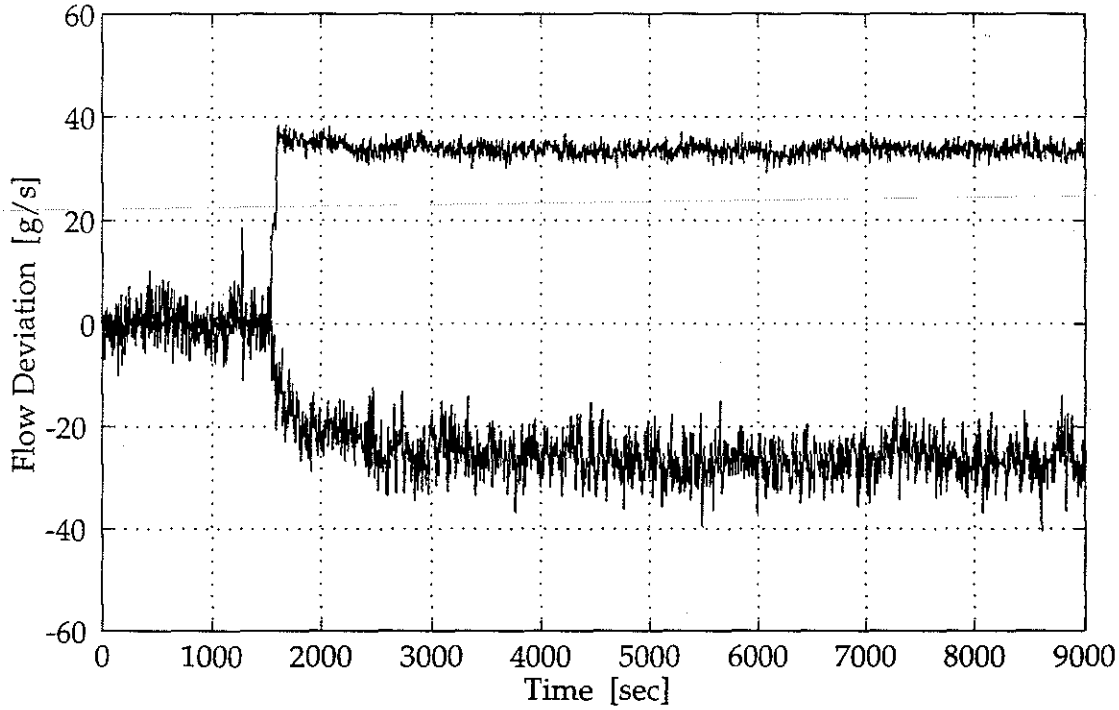

Figure 133. F0 F2 Flow and Secondary Flow from Data Set 26

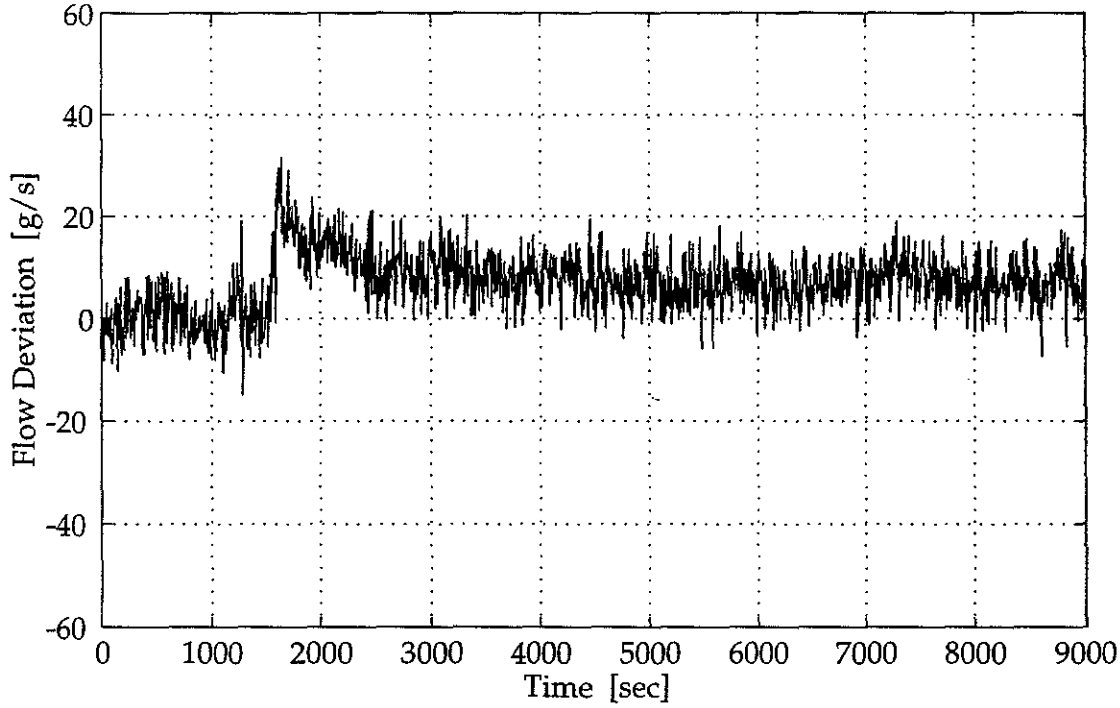

Figure 134. Sum of F0 F2 Flow and Secondary Flow from Data Set 26

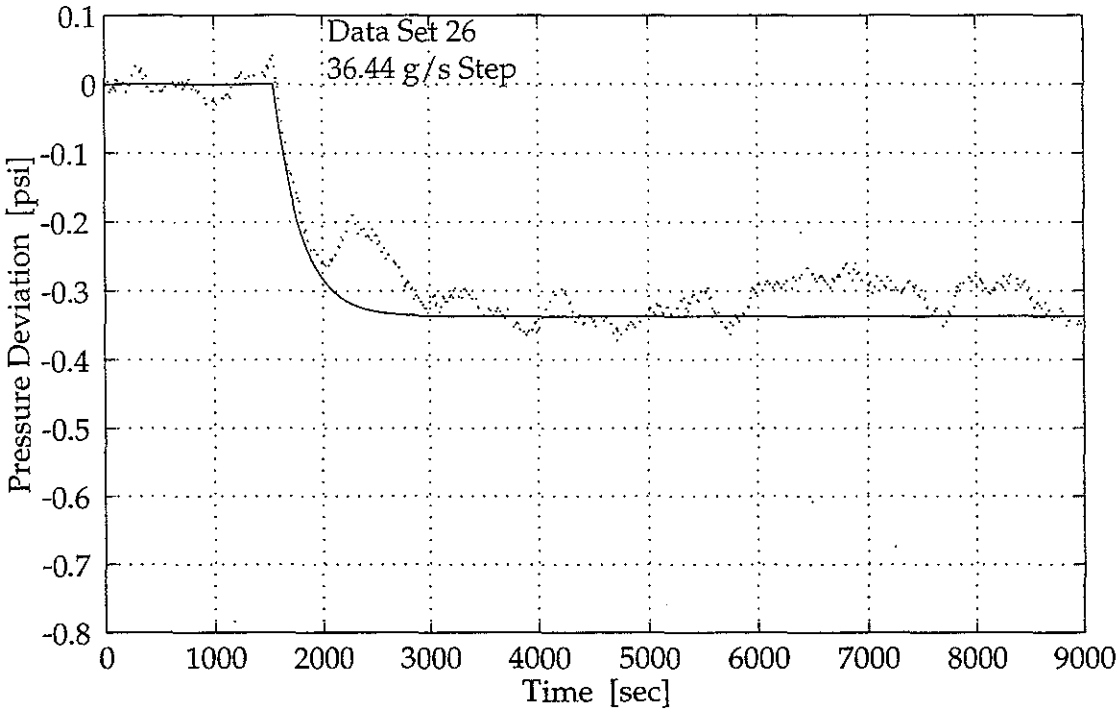

Figure 135. C0 P1 Data and Simulation for Step in F0 F2

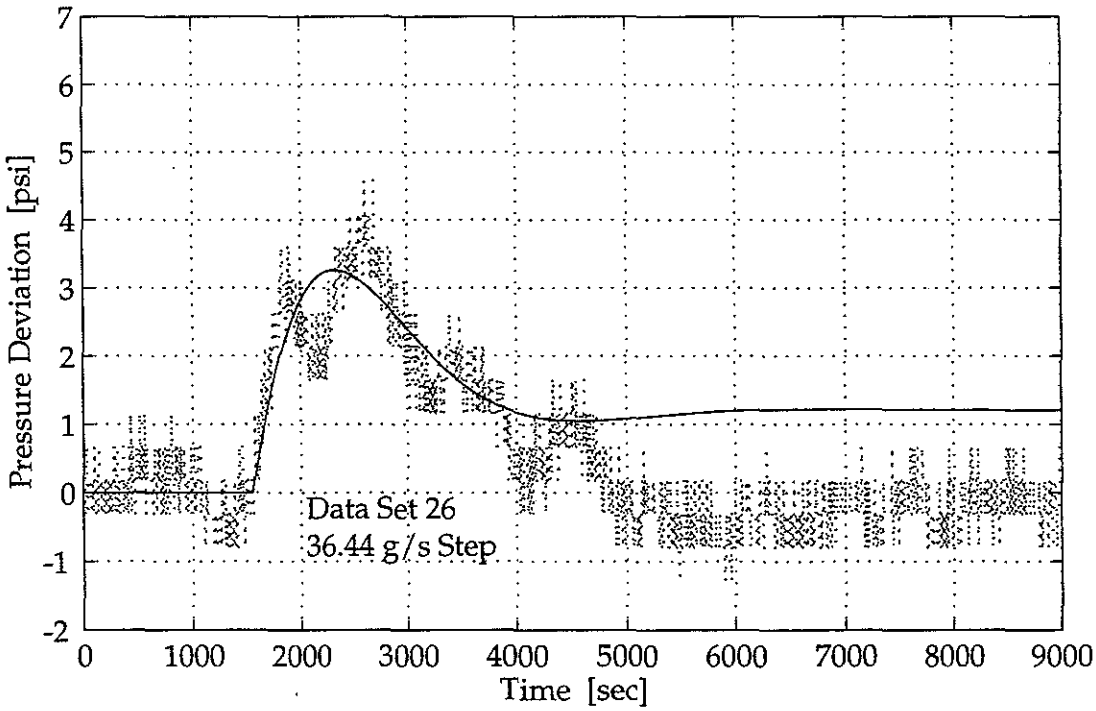

Figure 136. C0 P2 Data and Simulation for Step in F0 F2

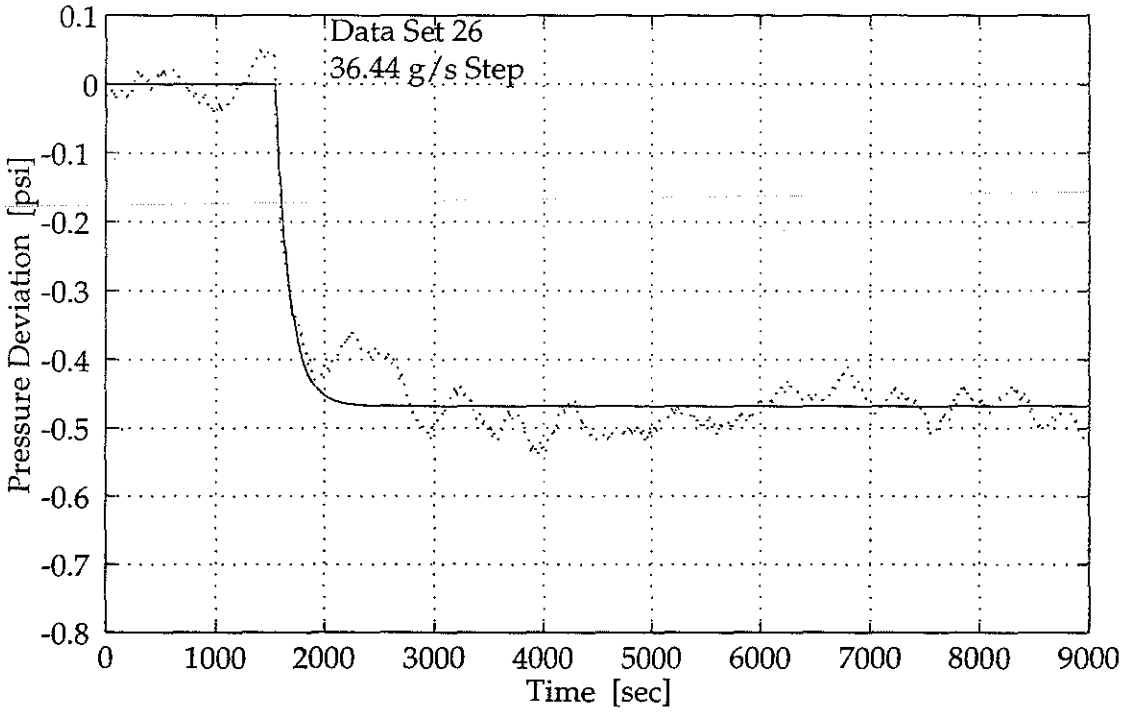

Figure 137. F0 P1 Data and Simulation for Step in F0 F2

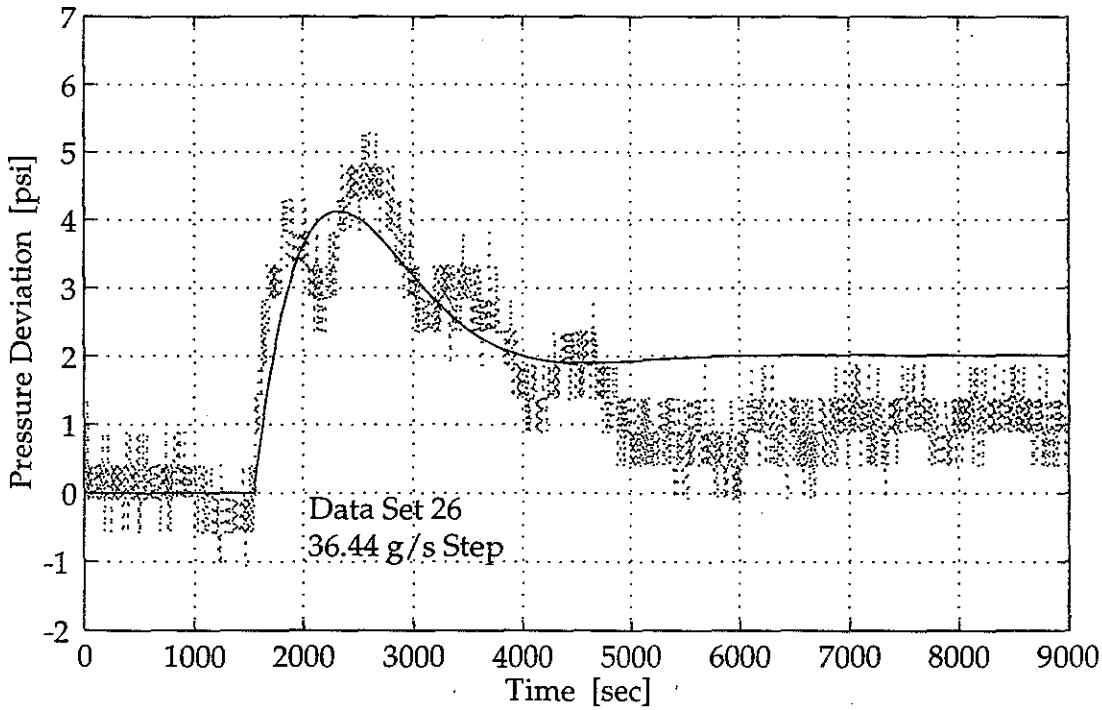

Figure 138. F0 P2 Data and Simulation for Step in F0 F2

responses for this set. The 36 g/s flow step was produced by loading two compressors at the same F0 location, one immediately following the other. The results of this test point to the strengths and weaknesses of the work done on the black box model. P1 is well represented by the models, though the final value are often only good to 25%, and the time constants are not always accurate. P2 is reasonably well modeled for its peak value, but the final value predicted is off, and perhaps the model should be adjusted to reflect this. For both discharge and suction pressures, better models would result from averaging model coefficients from additional longer data sets, if the effects of long term drift can be eliminated.

4.3.4 Validation for Inventory Flow Input. Figures 139 through 146, pertaining to data set 27, demonstrate the use of the inventory flow input model to predict response to a large flow change. Here flow is stepped up by 16 g/s, almost twice the size of the largest step used to create the model. Figure 139 shows this step, while Figure 140 indicates there is a slight decrease in the flow FD due to the reduction in pressure across valve DL. Apparently compressor flows may also be affected slightly by changes in inventory flow, as shown in Figure 141. Here an increase in the sum of compressor flows is seen, which perhaps is an indication of an influence of discharge pressure on compressor performance. As with the modeling data sets, Figures 143 and 145 show that P1 of data set 27 is not influenced by inventory flow change. The discharge pressure model predicts this particular P2 less accurately than for data sets having smaller flow steps. The P2 data does not decrease as much as

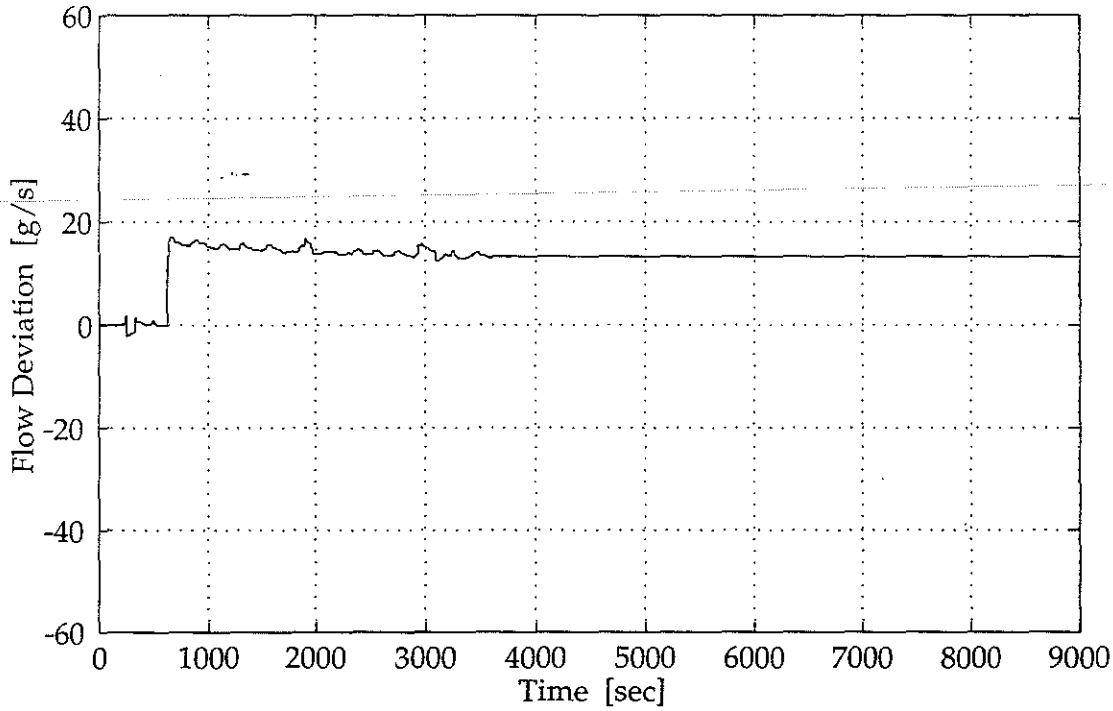

Figure 139. FC Inventory Flow from Data Set 27

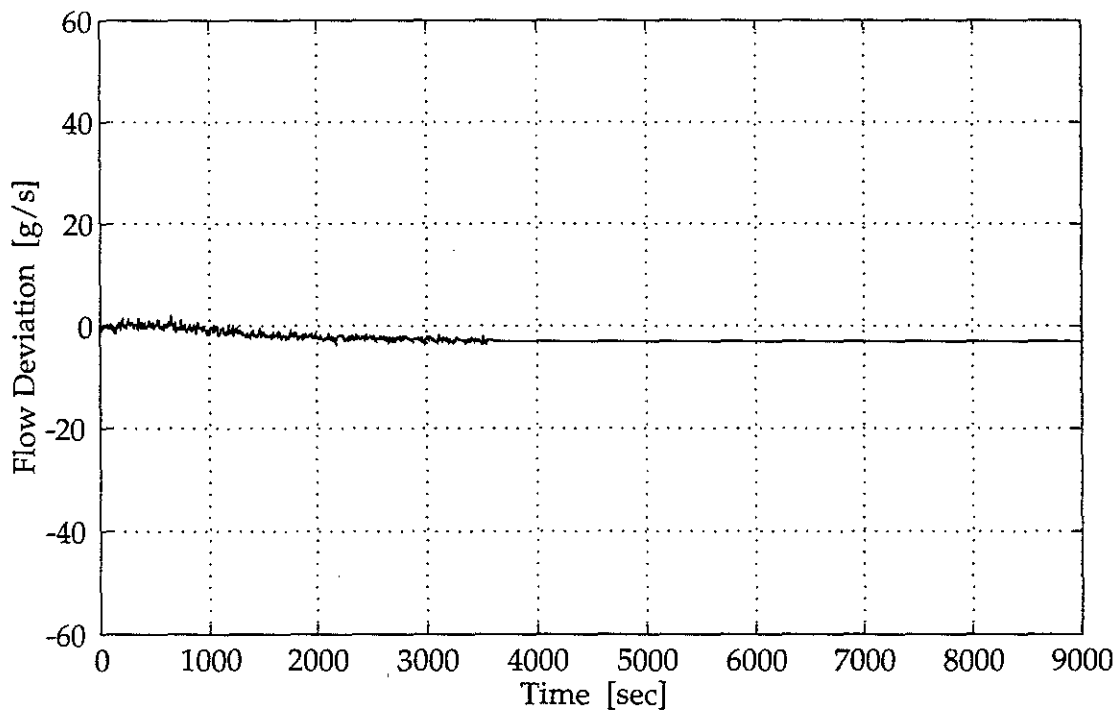

Figure 140. FD Inventory Flow from Data Set 27

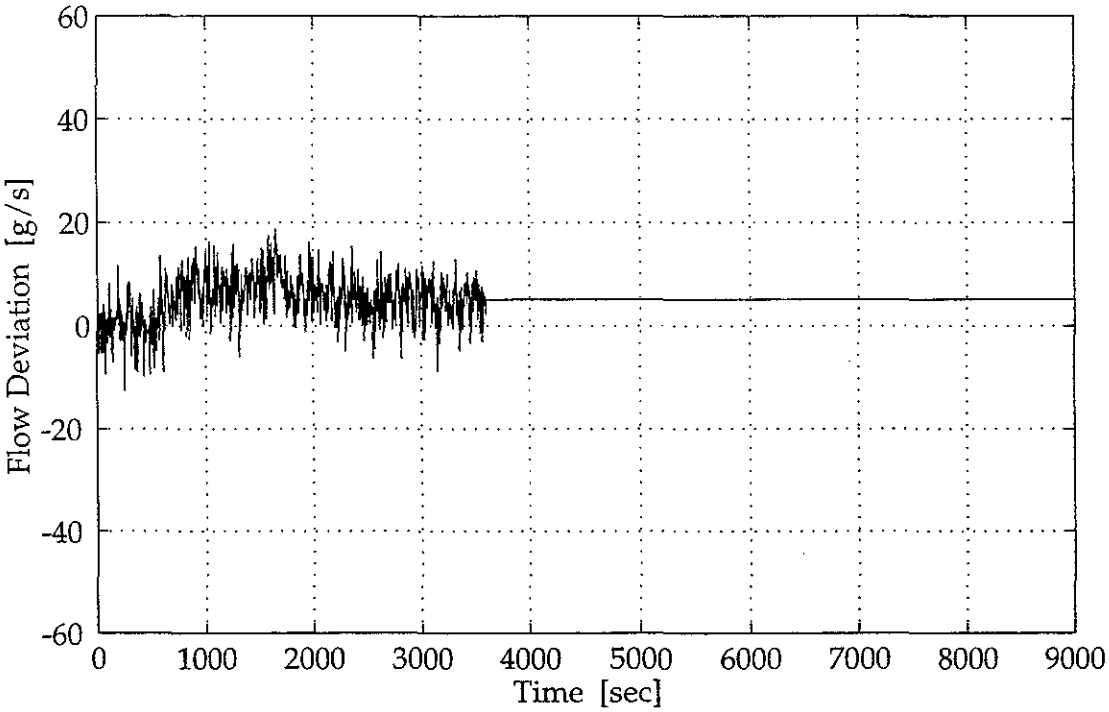

Figure 141. Sum of F2 Flows from Data Set 27

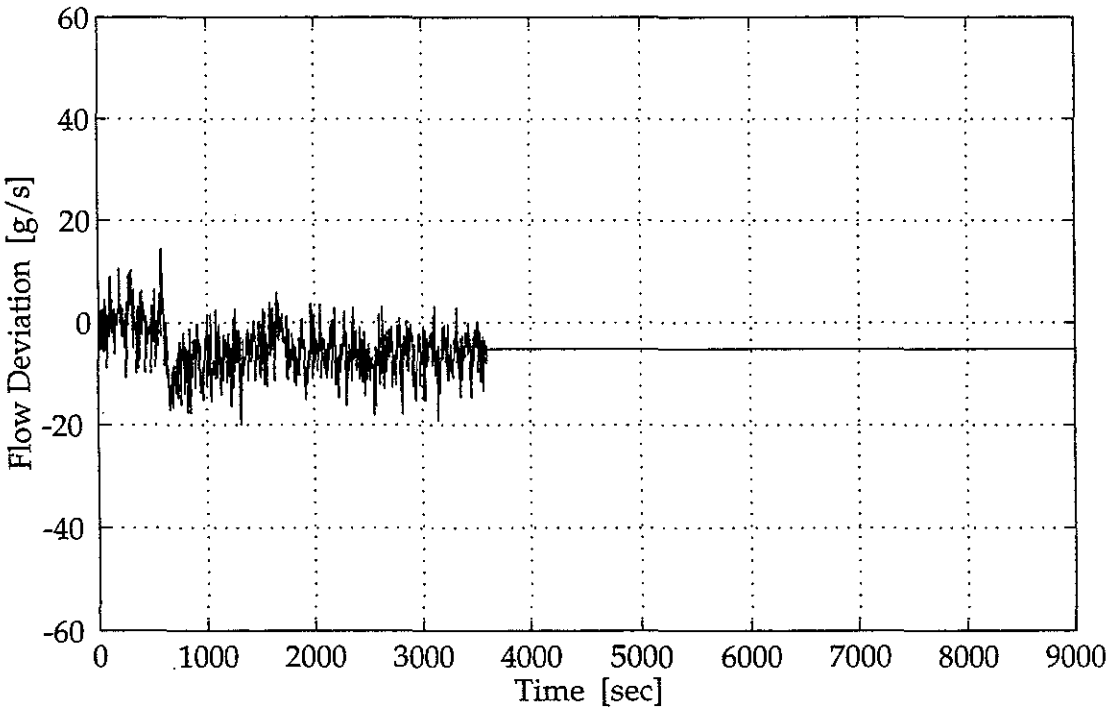

Figure 142. Net Flow Change into Discharge Header from Data Set 27

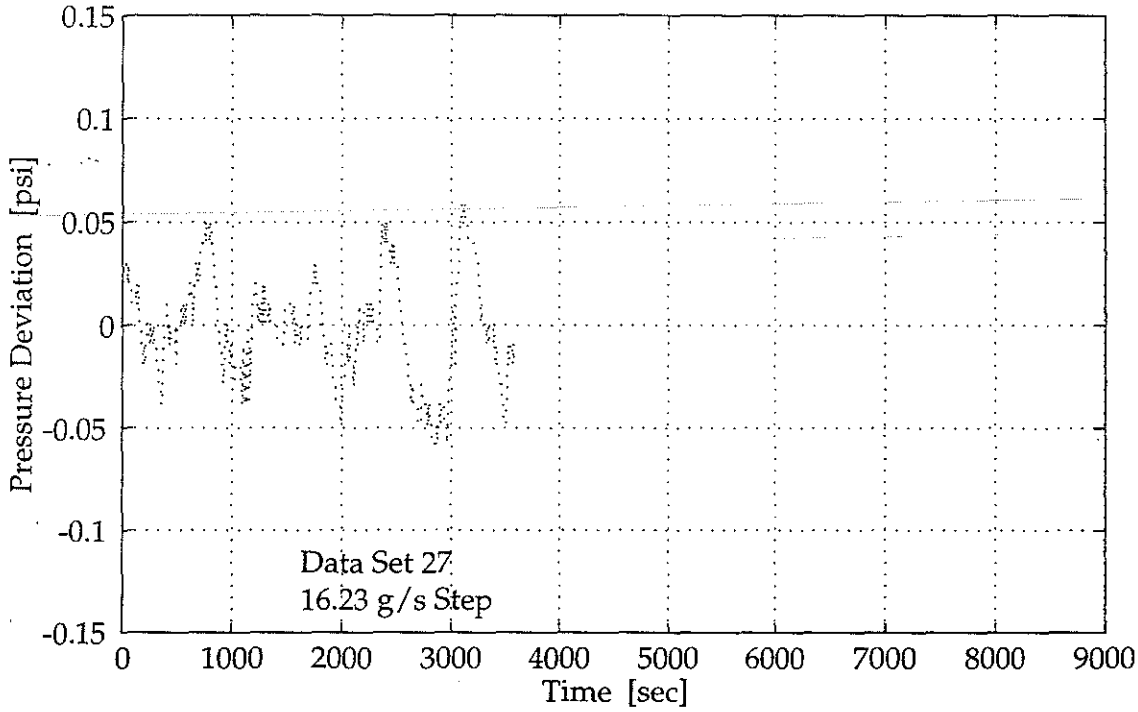

Figure 143. B0 P1 Data for Step in FC

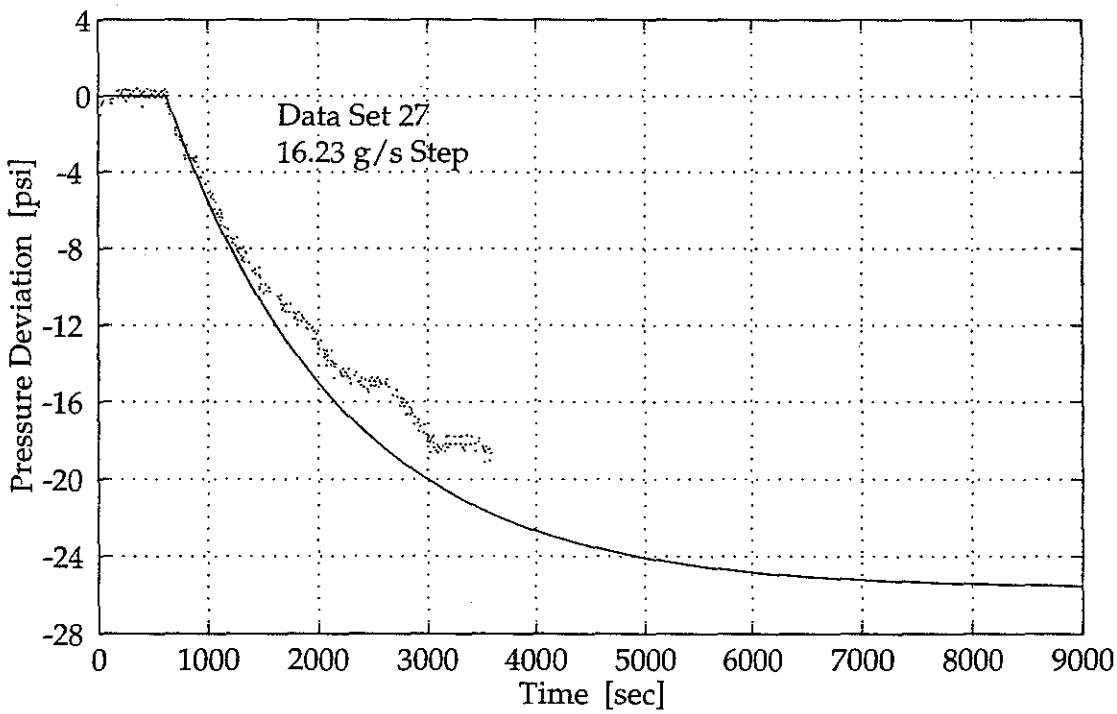

Figure 144. B0 P2 Data and Simulation for Step in FC

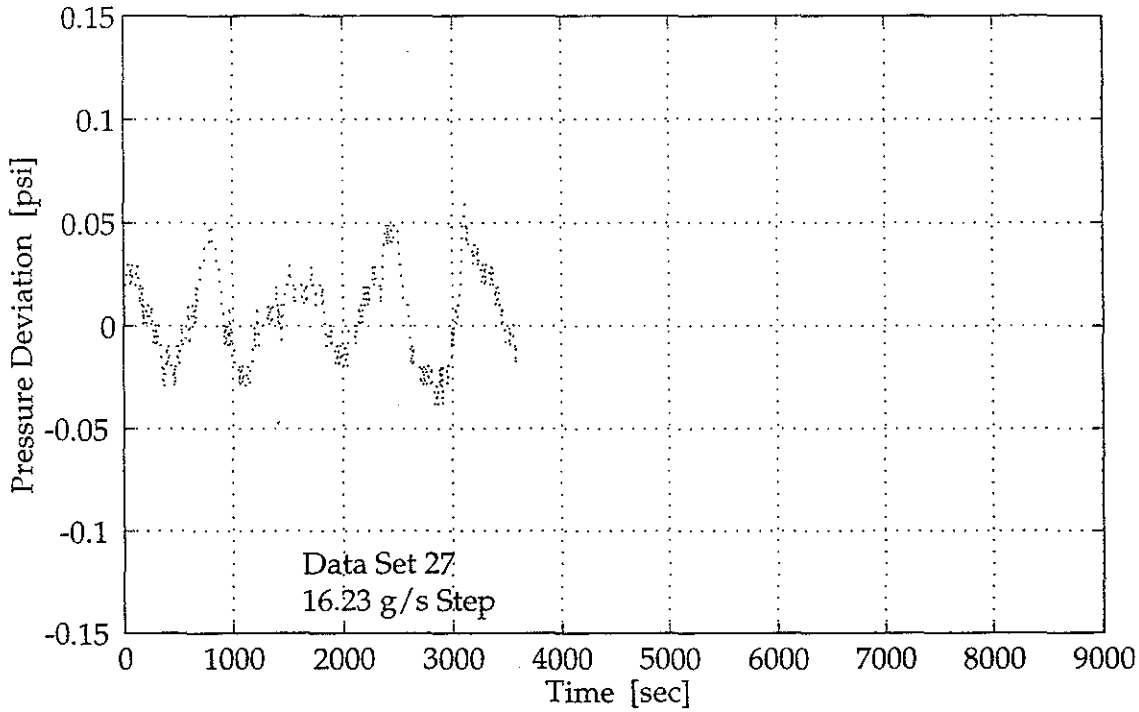

Figure 145. E0 P1 Data for Step in FC

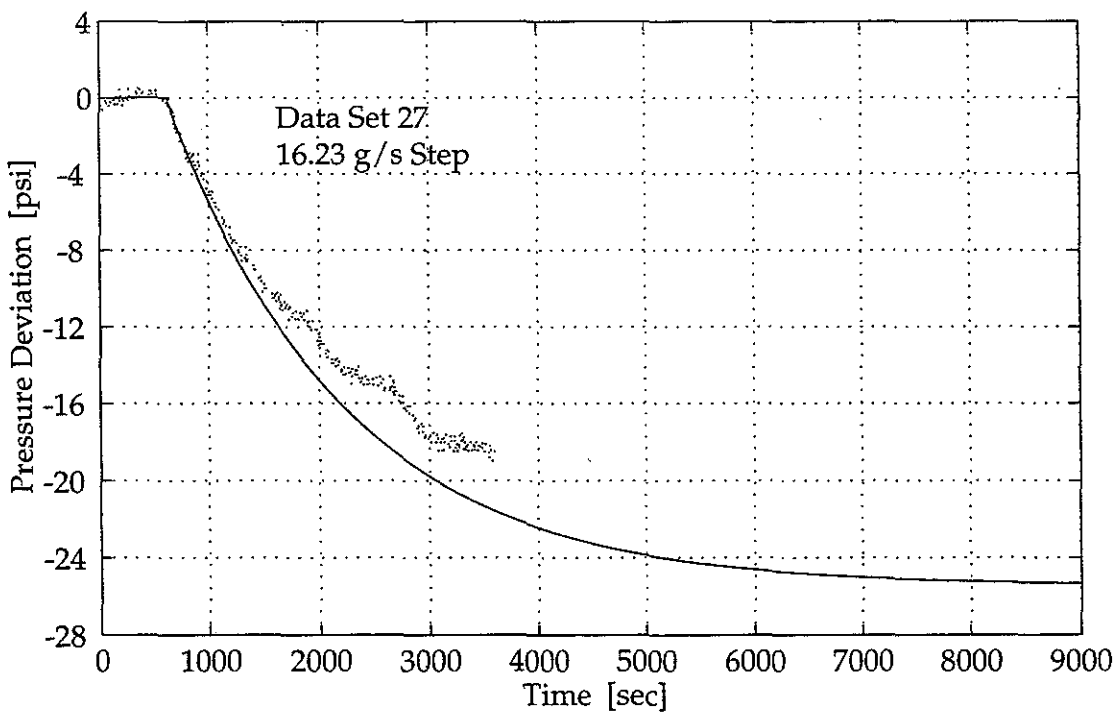

Figure 146. E0 P2 Data and Simulation for Step in FC

the model predicts, and the discrepancy might be explained by the increase seen in compressor flow into the discharge header.

Data set 28 contains a relatively small step in inventory flow. Figures 147 through 154 show flows and pressures for this test along with the model predictions. Now that data set 27 has pointed to a dependence of compressor flow on inventory flow, this can also be seen in Figure 149 for data set 28. For the step reduction in FC, the resulting P2 increase seems to cause a decrease in the compressor flow sum. Thus, decreasing the flow out of the discharge header causes a decrease in flow into the header. This is opposite from the result for data set 27 where increasing the flow out of the discharge header causes an increase in flow into the header. These inlet and outlet flow changes act to diminish the change in net flow into the discharge header, as shown in Figures 142 and 150.

4.3.5 The Black Box System Model. An example block diagram showing the interconnection of averaged individual models to form a black box system model is given in Figure 155. The diagram is simplified to three compressor locations to improve clarity, but it could be extended to include all compressor locations as well as the inventory flow. The black box model requires the initial value of the suction pressure at a stepped compressor, and this is depicted as an input to the non-linear function block along with loader valve position. This non-linear block is modeled with the static portion of (3.24), which consists of (2.6) and its correction for initial suction pressure.

Some explanation of the nomenclature used within other blocks, for example G2BA(q), is warranted. The blocks are designated so that their

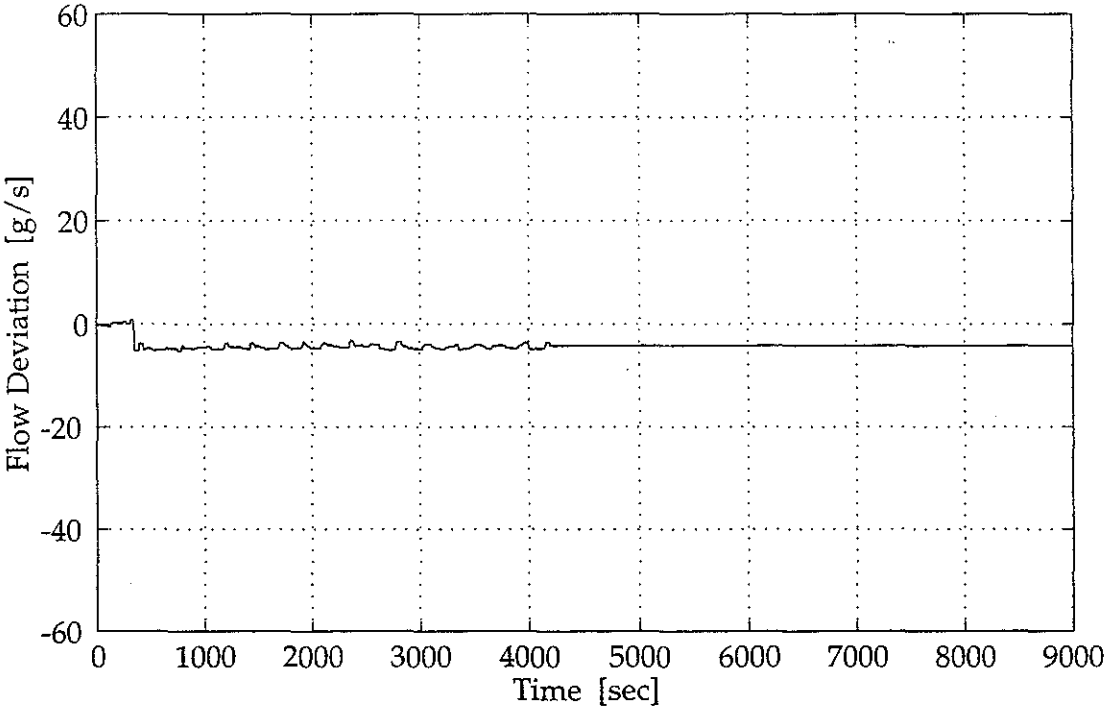

Figure 147. FC Inventory Flow from Data Set 28

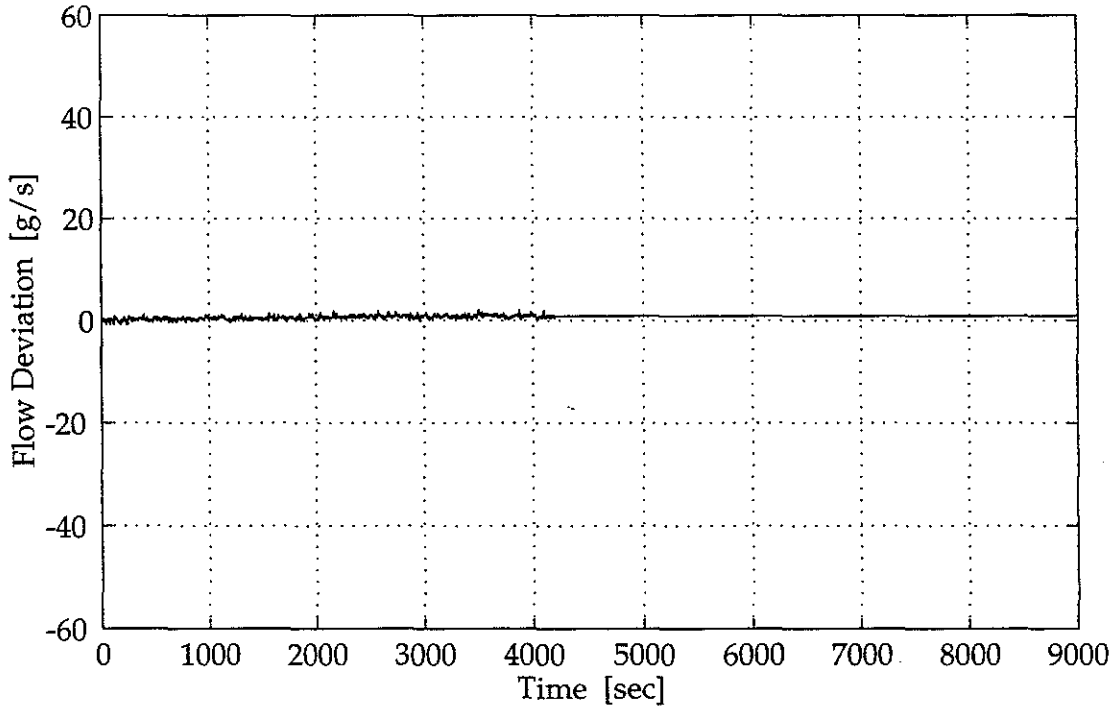

Figure 148. FD Inventory Flow from Data Set 28

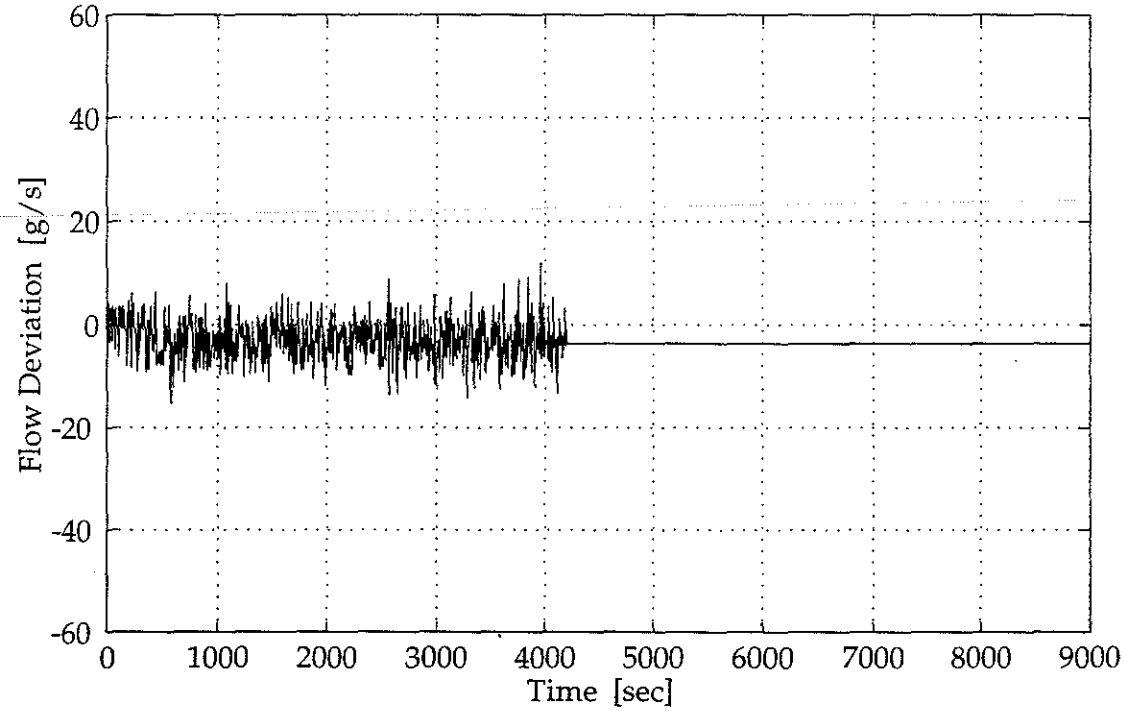

Figure 149. Sum of F2 Flows from Data Set 28

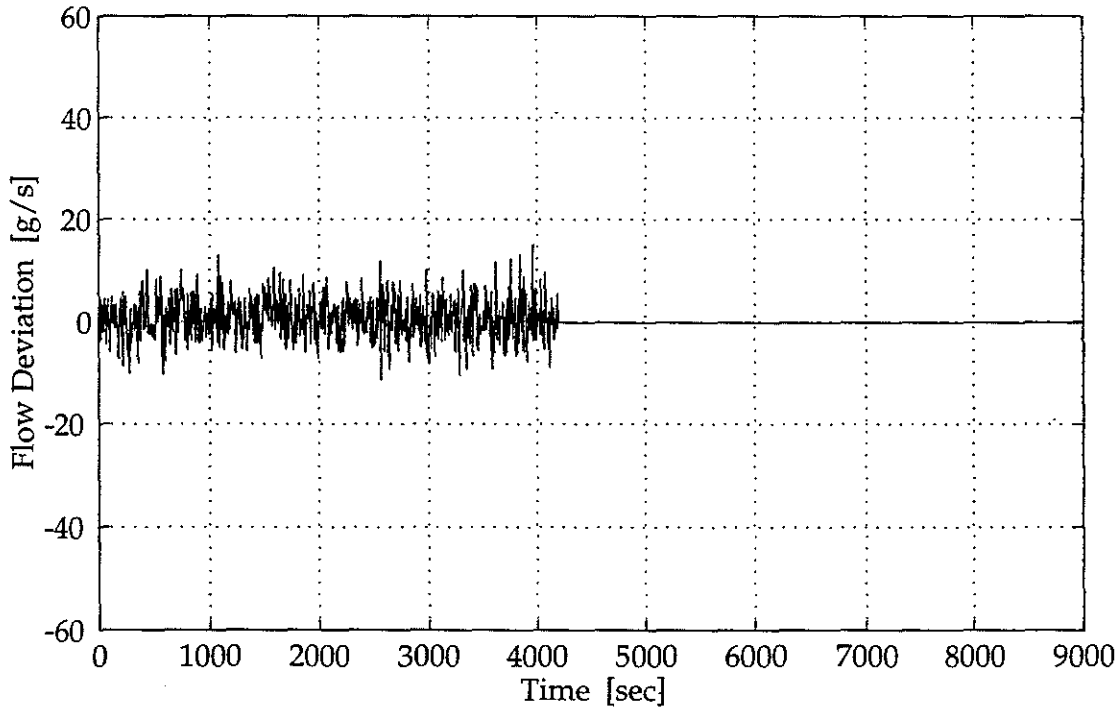

Figure 150. Net Flow Change into Discharge Header from Data Set 28

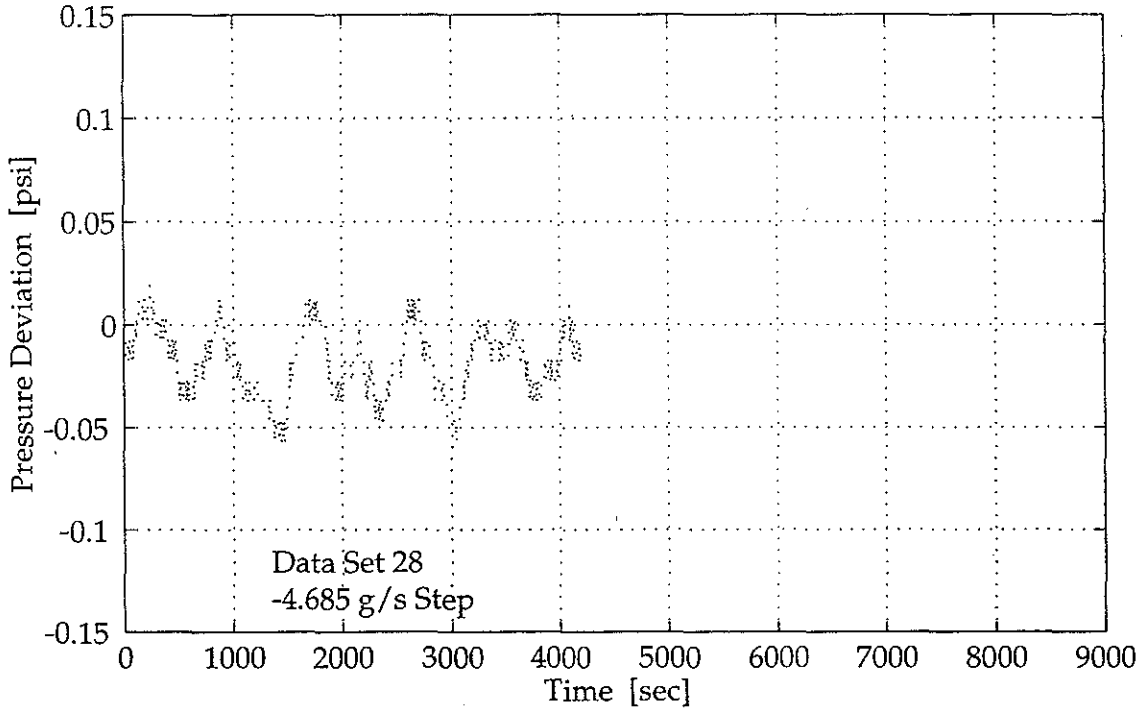

Figure 151. B0 P1 Data for Step in FC

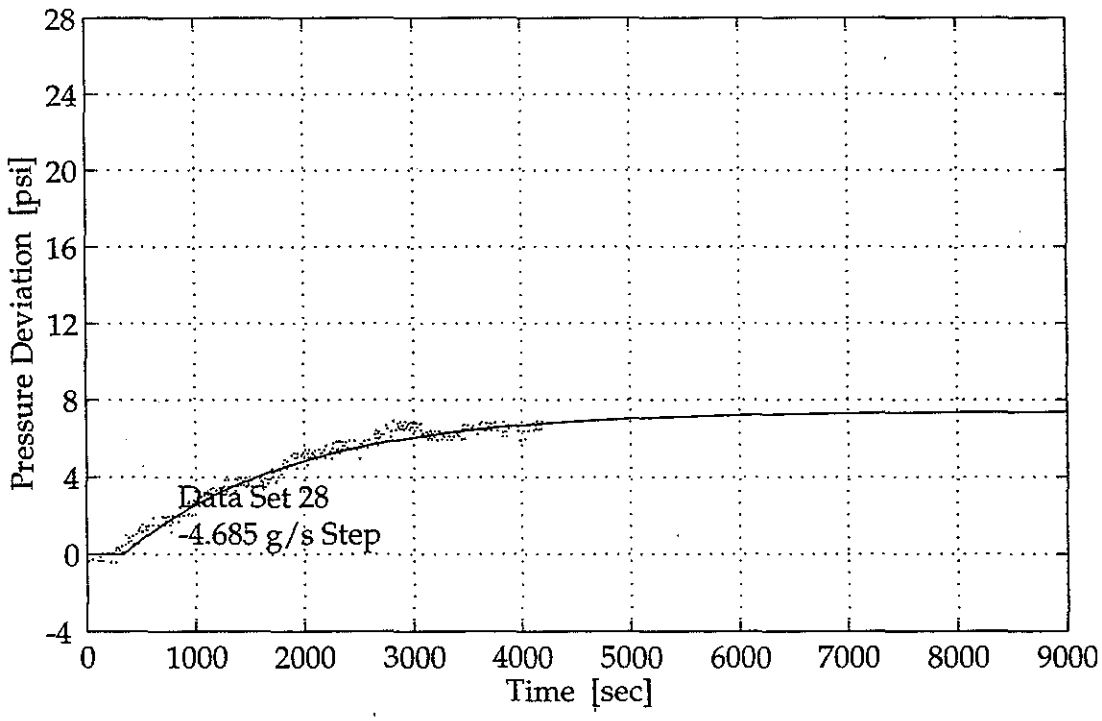

Figure 152. B0 P2 Data and Simulation for Step in FC

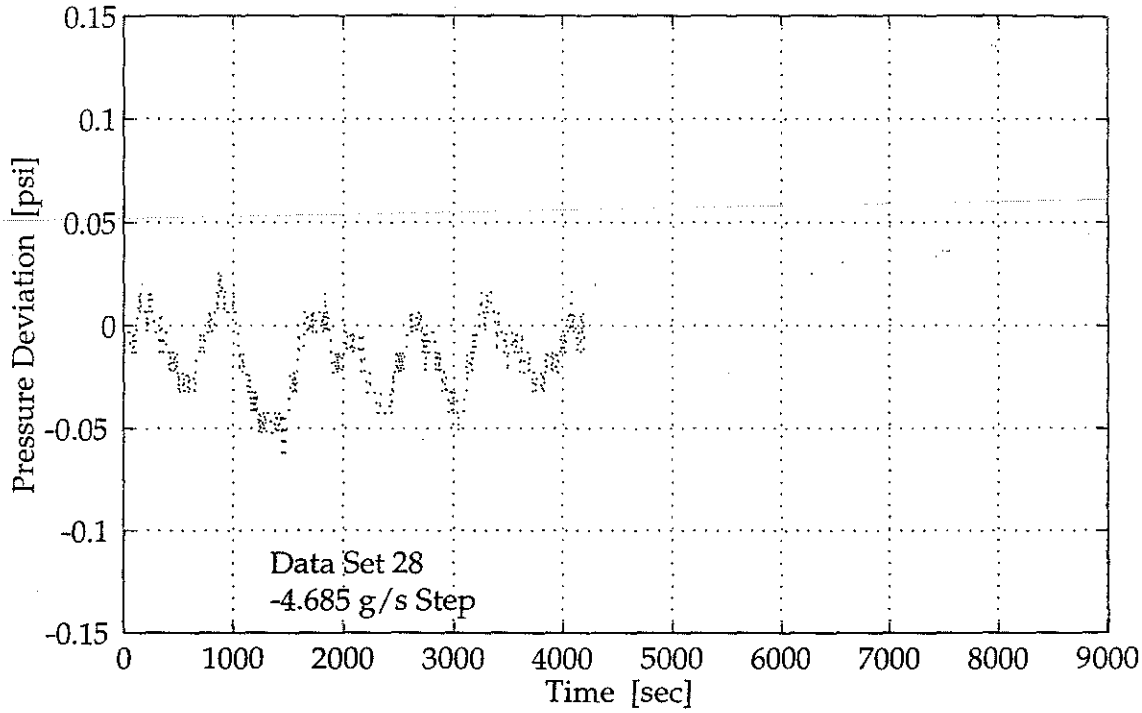

Figure 153. E0 P1 Data for Step in FC

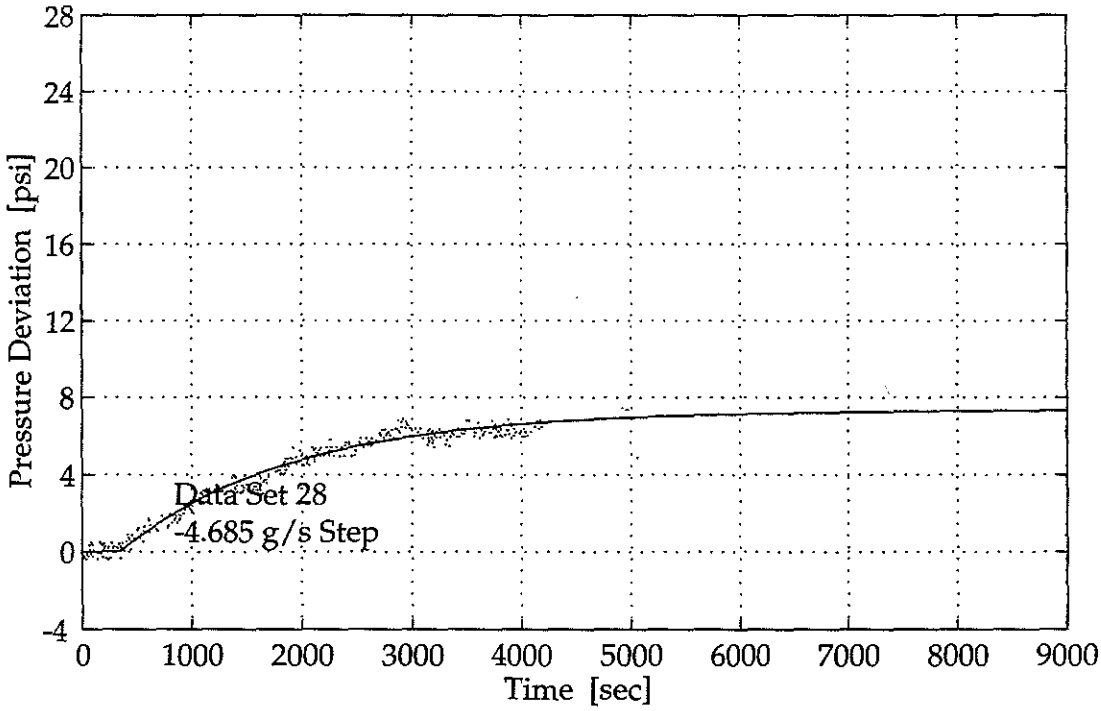

Figure 154. E0 P2 Data and Simulation for Step in FC

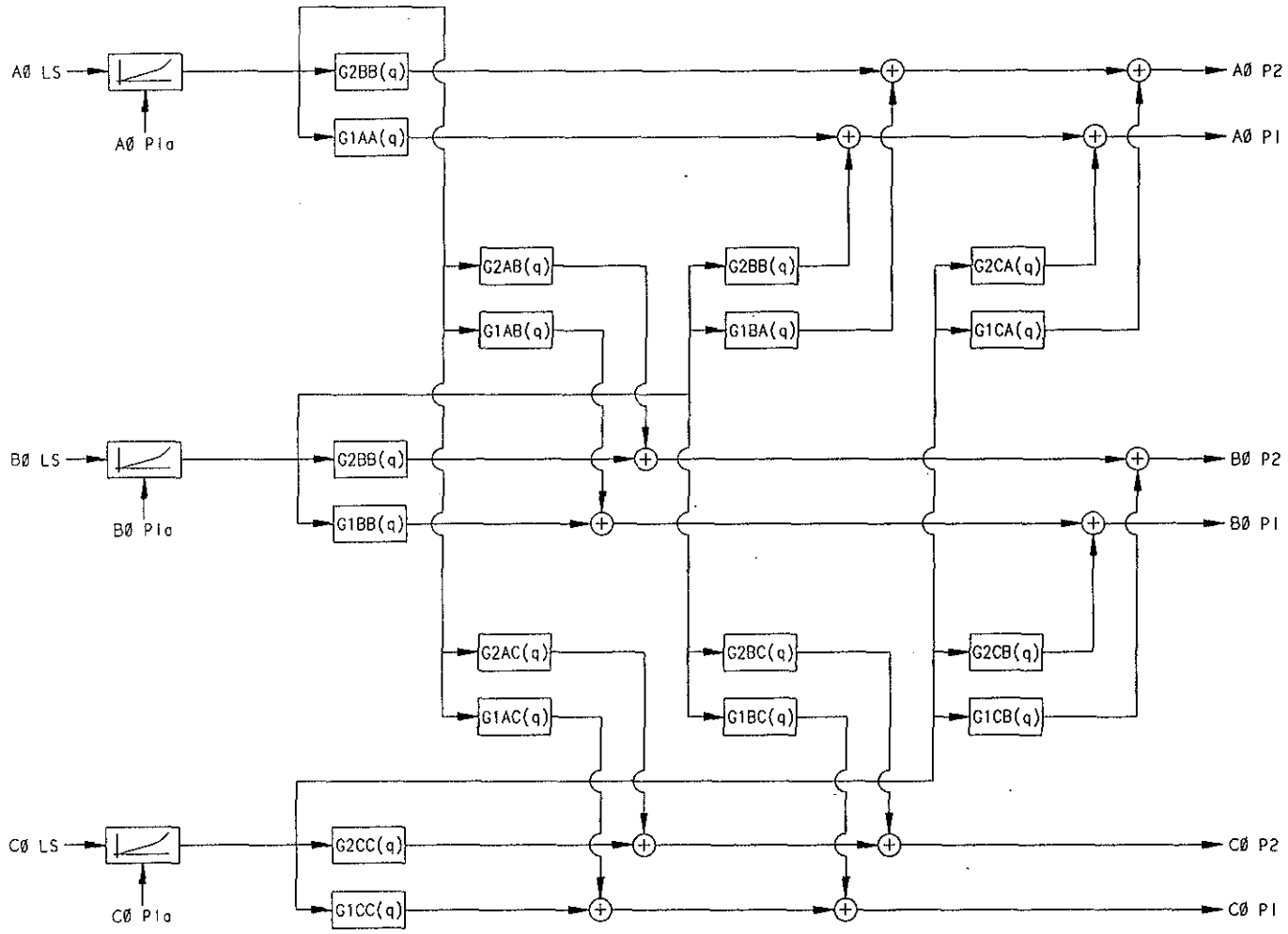

Figure 155. Black Box System Block Diagram for Compressor Flow Inputs

function is encoded in that designation. Thus, information such as whether the pressure involved is suction or discharge, as well as the input and output locations, can be obtained from the block name. The example, G2BA(q), designates a forward path transfer function (G) having a discharge pressure deviation output (2) and relates the flow deviation at the B0 compressor location (B) to the pressure deviation at the A0 location (A). This particular function contains the dynamic portion of (3.24) with four coefficient values given in the third through sixth row of Table 3 for the A0 output location. It represent the response of A0 P2 to a change in B0 F2 only.

4.4 Detailing System Feedback using White Box Models

In Chapter III, the assumption was made that the responses of P1 and P2 can be separated into components that represent the effects of a primary F2 flow and all other F2 flows. This seems a reasonable path to pursue based on a knowledge of the fluid mechanics involved. In practice, what was presented in Figures 56 through 61 is open to considerable interpretation. Compare Figures 56 and 58, for instance, in which the primary components of P1 have different values depending on whether the combined flow or separate flow model is used to generate the plots. It is unclear whether to believe one, the other, or neither version of this P1 component breakout. In fact, early attempts at separating the effects of different flows included a nine input model utilizing the nine F2 flows separately. This fails completely, with some flows being accorded great significance while others are not weighed at all. Under these circumstances, assumptions are made in an effort to extend manipulation on, and usefulness of, the white box model.

4.4.1 Forming a Common Secondary Component. The secondary flow (sum of flows other than the primary flow), is forced to represent an effect that occurs over the entirety of both pressure headers. Extending this idea, one simplifying assumption is that the secondary flow causes equal pressure changes at each compressor location. For P1, the separate flow model approaches this condition more satisfactorily than the combined flow model. The separate flow model also treats the distant SY location differently, perhaps indicating a better representation of the real system. On these somewhat weak arguments, the separate flow model is favored for representing the P1 response. In the case of P2, there exists only the choice of the separate flow model since the combined flow model was not successfully realized.

With the exception of the SY values, coefficients for the separate flow models are averaged to provide a common model for the secondary components of P1 and of P2. The SY results are thrown out since they may not be representative of the rest of the ring. Next, the common secondary component values of both P1 and P2 are calculated, and these are subtracted from the original pressure data to produce new data streams. This manipulated data represents primary components for P1 and for P2 under the assumptions that the separate flow models hold and the secondary flow component is common. New models are then calculated for the primary component based on this manipulated data.

Figures 156 and 157 depict model results based on the above assumptions. Shown are the assumed common secondary component, the

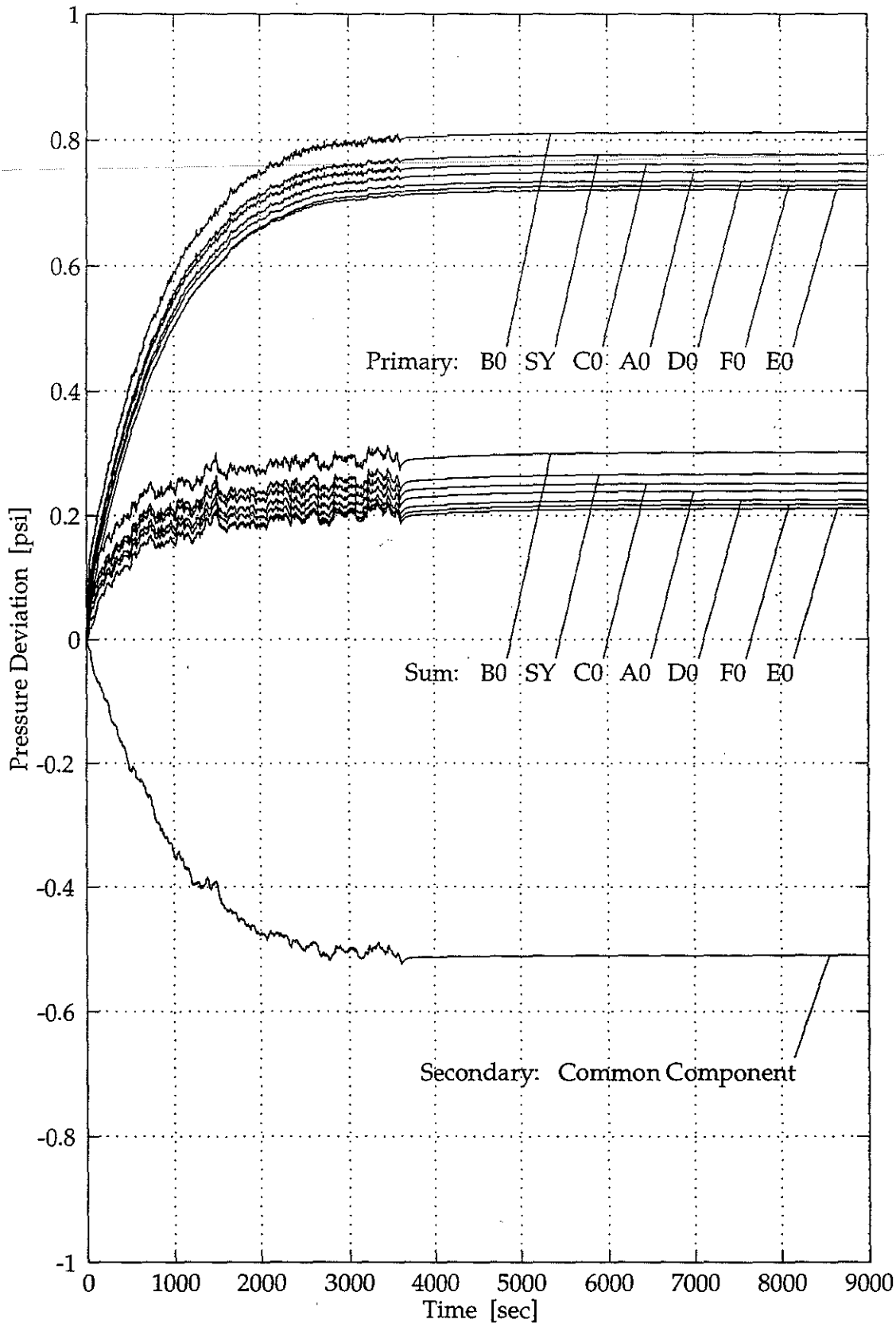

Figure 156. P1 Components for Common Secondary Model from Data Set 5

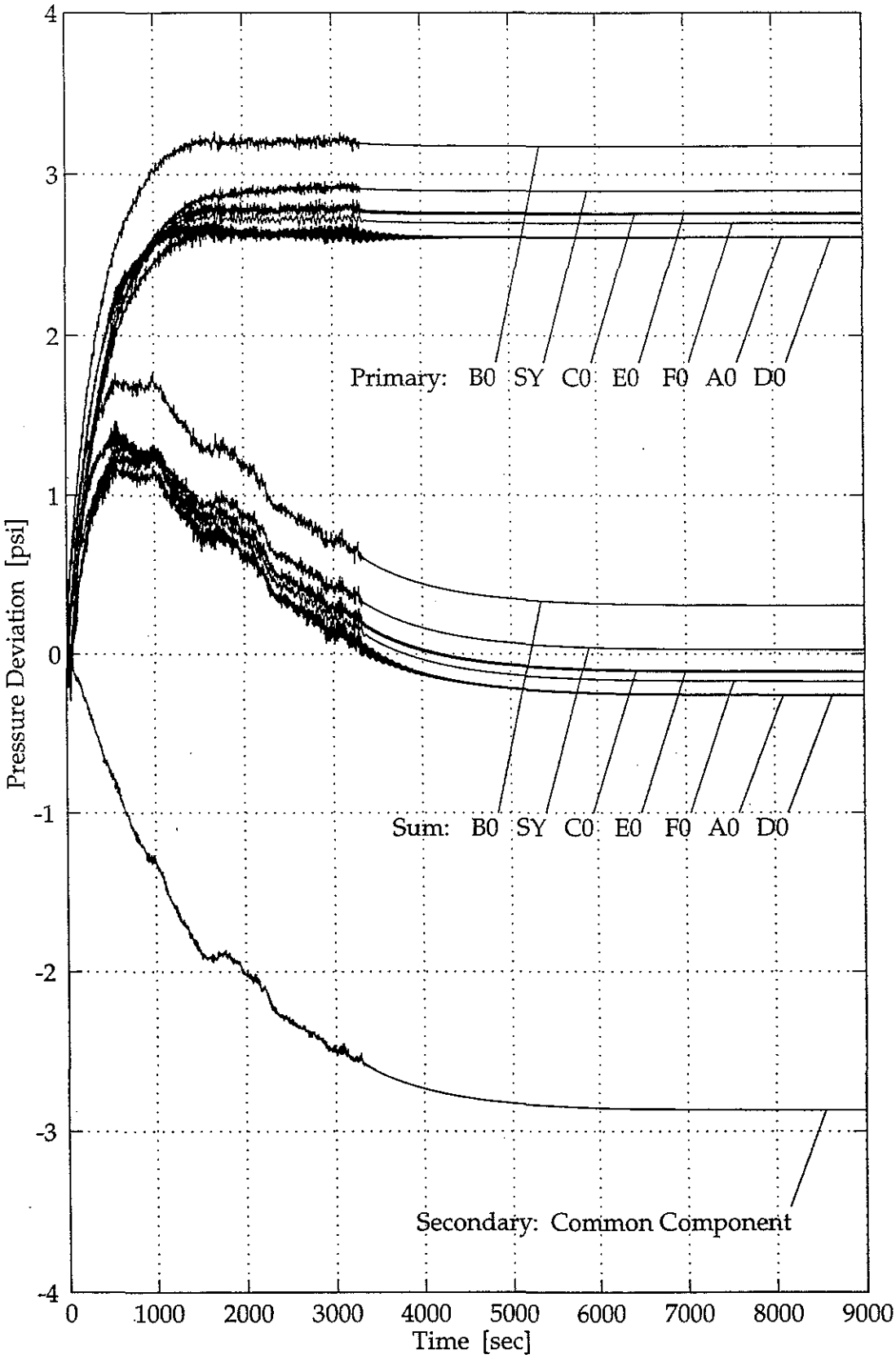

Figure 157. P2 Components for Common Secondary Model from Data Set 3

individually recalculated primary components, and the sum of each primary component with the common primary component. The sum represents the original data to the same degree of accuracy that is typical of Figures 48 through 51. The models retain the forms of (3.24) for P1 and of (3.25) for P2, and have the model coefficient values listed in Tables 4 and 5. The coefficients values for the P1 model appear to have systematic variation from output location to output location, but the same can not be said for values of the P2 model. Converted to a pole-zero representation, the P2 model would show that several output locations contain a pair of complex poles. A potential improvement to the model for simulation purposes might include filtering the data (as was done for the black box model) and recalculating the primary component model, perhaps with lower order polynomials. This might result in less variation in P2 model coefficients with location.

Table 4. P1 and P2 Model Coefficients for Secondary Flow Input

Output Location	P1 Coeff.	Coeff. Value	P2 Coeff.	Coeff. Value
All Locations	b15	-5.3838e-04	b17	-6.1626e-04
	b25	+5.1337e-04	b27	+2.5299e-03
			b37	+1.5394e-03
	f15	-1.8786e+00	f17	+2.3273e-01
	f25	+8.7917e-01	f27	-4.1294e-01
			f37	-8.0306e-01

4.4.2 The White Box System Model. The white box modeling results thus far simply relate flows to pressures, and as they stand are not as practical as the black box model for system simulation. However, the flow to pressure transfer functions determined from white box modeling can also be

Table 5. P1 and P2 Model Coefficients for Primary Flow Input

Output Location	P1 Coeff.	Coeff. Value	P2 Coeff.	Coeff. Value
A0	b14	-1.6034e-03	b16	+8.4651e-03
	b24	+1.5117e-03	b26	-4.0926e-03
			b36	+2.2100e-03
	f14	-1.6202e+00	f16	-2.4458e-01
	f24	+6.2207e-01	f26	+2.4541e-01
			f36	-9.6304e-01
B0	b14	-3.3025e-03	b16	+1.3754e-02
	b24	+3.1774e-03	b26	+1.0479e-02
			b36	-2.0382e-02
	f14	-1.5264e+00	f16	-6.1867e-01
	f24	+5.2880e-01	f26	-1.4611e-01
			f36	-2.1706e-01
C0	b14	-2.4572e-03	b16	-1.3963e-02
	b24	+2.3216e-03	b26	-1.8029e-03
			b36	+2.1792e-02
	f14	-1.4631e+00	f16	+3.7950e-01
	f24	+4.6586e-01	f26	-7.6871e-01
			f36	-5.7813e-01
D0	b14	-1.0368e-03	b16	-1.8212e-02
	b24	+9.7724e-04	b26	+3.5843e-03
			b36	+2.1508e-02
	f14	-1.7493e+00	f16	+7.1252e-01
	f24	+7.5058e-01	f26	-9.7955e-01
			f36	-6.9354e-01
E0	b14	-8.7773e-04	b16	-1.1914e-02
	b24	+8.1101e-04	b26	+1.0642e-03
			b36	+1.4789e-02
	f14	-1.7137e+00	f16	-1.0394e-01
	f24	+7.1508e-01	f26	-9.8611e-01
			f36	+1.1146e-01
F0	b14	-4.6324e-04	b16	+7.3950e-03
	b24	+4.4025e-04	b26	-1.8463e-02
			b36	+1.1632e-02
	f14	-1.8972e+00	f16	-2.4938e+00
	f24	+8.9768e-01	f26	+2.2501e+00
			f36	-7.5319e-01
SY	b14	-2.1800e-03	b16	-2.6162e-03
	b24	+2.0734e-03	b26	+2.1672e-02
			b36	-1.6170e-02
	f14	-1.5782e+00	f16	-7.7352e-01
	f24	+5.8041e-01	f26	+1.5831e-01
			f36	-3.6988e-01

connected in a block diagram to help explain the internal operation of the compressor and header system. Figure 158 is such a diagram, presented in a simplified form showing a system having only three compressor locations. This reduction from nine to three compressor locations allows for a simpler presentation, but the figure could also be expanded to include all locations and inventory flows. Some blocks are taken directly from the white box modeling, and others are derived below. Note that the white box model for the secondary flow is not required in this system model since it's effect is taken into account through the feedback mechanism included in the block diagram.

Referring to Figure 158, blocks are designated so that their function is encoded in that designation. Note that the notation used in Figure 158 is the same as that used for the black box block diagram of Figure 155, but the transfer functions represented in each have very different meanings. Information such as whether the block is in a forward path (having flow input) or feedback path (having pressure input), whether the pressure involved is suction or discharge, and the input and output locations is all obtained from the block name. For example, now the block labeled G2BA(q) designates a forward path transfer function (G) having a discharge pressure deviation output (2) and relates the flow deviation at the B0 compressor location (B) to the pressure deviation at the A0 location (A). This particular function has the form of (3.27), and has coefficient values given in the last column of Table 5 for the A0 output location. It represent the effect on A0 P2 due to a change in B0 F2 only. Theoretically, this function works equally well

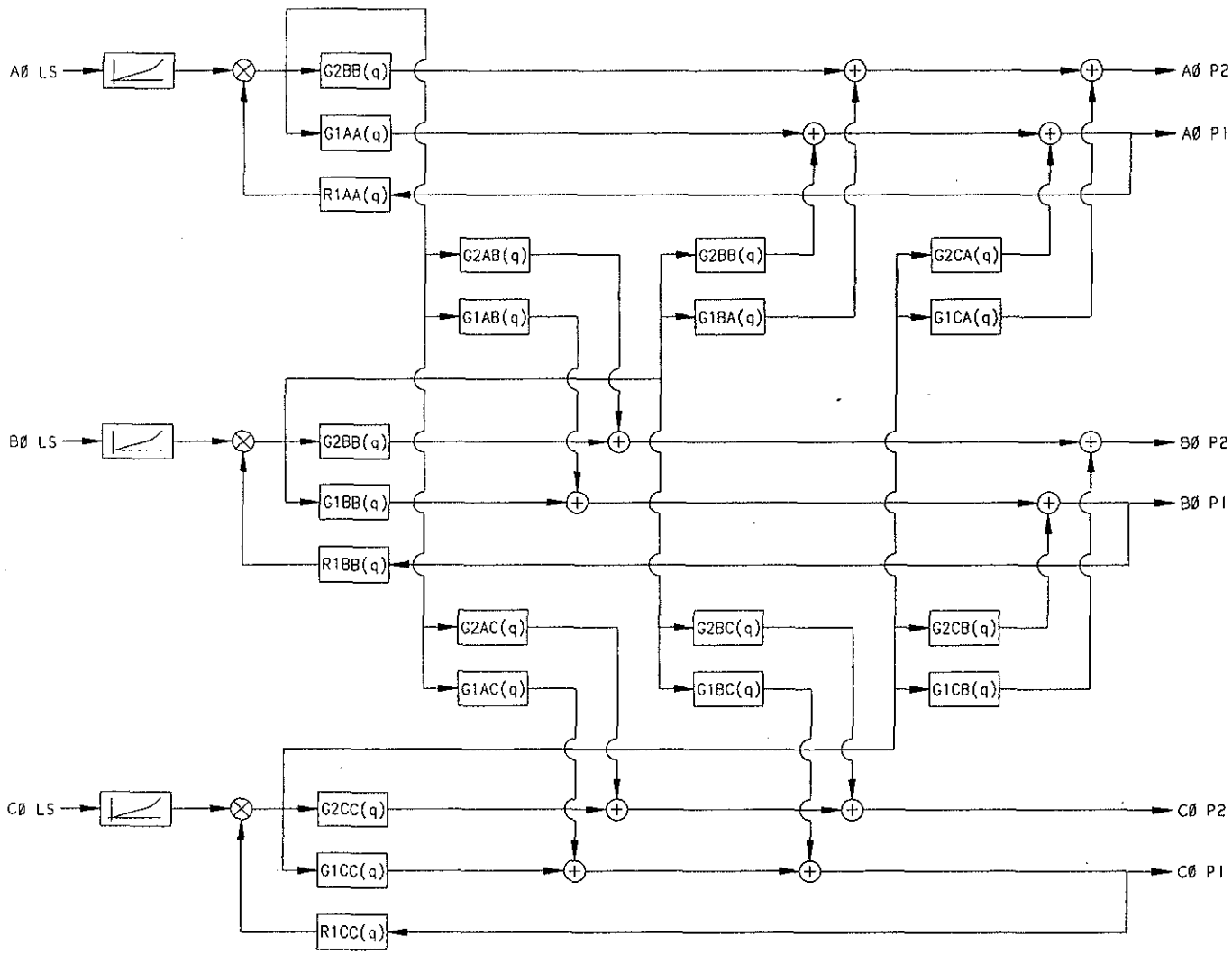

Figure 158. White Box System Block Diagram for Compressor Flow Inputs

for B0 F2 changes caused either by movement of compressor valve B0 LS or by reduction in B0 P1. Recall that B0 P1 influences B0 F2 since its absolute value, B0 P1a, is directly related to the density of gas entering the compressor.

The feedback of P1 influences the F2 flow in a nonlinear manner, and thus Figure 158 does not represent a linear system. While the feedback is negative in the sense that it acts to decrease the effect of the valve movement, it does not enter a summing junction. Instead, the P1a value multiplies the flow step calculated from compressor valve movement using (2.6), to arrive at F2 flow values given by

$$F2(t) = \frac{P1a(t-1)}{Pc} \cdot [f\{LS(t-1) - f\{LS(t-2)\}]. \tag{4.1}$$

Note that this multiplication is also reflected in (3.23) used in the black box modeling to calculate flow step changes. Also, the P1a value is not linearly related to P1 since P1a is the absolute value of the suction header pressure and P1 is the deviation from some base value in that pressure. That is, the two differ by a changing additive value. The feedback functions, such as R1BB(q), are thus not linear either, but could be represented by a first order equation with an appropriate initialization of P1a. This equation takes the form

$$\frac{P1a(t)}{Pc} = \frac{P1a(t-1) + P1(t)}{Pc} = \frac{1}{1-q^{-1}} \cdot \frac{P1(t)}{Pc}. \tag{4.2}$$

Since only one input location was considered when constructing the white box model form, additional work is required to fill other blocks of Figure 158. This could be done by processing additional data sets from other input locations, or by simply assuming all locations behave as the B0 input

location. The later is not unreasonable given the nearly uniform results of the black box modeling. With nonlinear simulation software, for example SIMULINK from the MathWorks, Inc. or with considerable dedicated programming in MATLAB, the blocks of Figure 158 could be connected mathematically. This would produce a more complex system model than that of the black box model, but one providing details of flow changes around the ring for individual compressor loader valve changes. Uses of such a model are discussed in Section 4.6.

4.5 Disturbance Characterizations. A number of disturbances to the system were observed during the experimental investigation. One was the unintended movement of compressor valves, and the model for this needs no further explanation. Another was the disturbance to P1 caused by upsets or oscillation in the refrigerators. This disturbance enters the suction header directly as a change in the F3 flow shown in Figure 1. A final identified disturbance was associated with temperature changes that caused heating and cooling of those portions of the suction and discharge header exposed to ambient conditions. This is a very slow disturbance, but one with significant amplitude. The direct addition of these disturbances at the plant outputs is an appropriate way to account for all of these effects.

Typical disturbances caused by F3 can be seen in Figure 28. (Recall that data set 22 shows the response to inventory valve changes, which perturbs only P2.) The independent disturbance to P1 is roughly sinusoidal with period of 740 seconds and peak amplitude of 0.03 psi, but these characteristics are dependent on the particular data set. The phase of this signal with respect to

inventory or compressor valve changes is arbitrary, as can be seen by comparing different data sets. Modeling this disturbance directly from the F3 flows contained in data set 22 proves to be a difficult task. For one, the signals are small, both of the P1 disturbance and even of the sum of all F3 flows. In addition, the F3 instrumentation is unable to register flow at the low end so that disturbance input signals are sometimes lost. Just as for the F2 flows, the F3 flows are expected to produce responses that vary with location. Ideally they would be treated individually, but changes in individual F3 flows are often hidden by signal noise.

To mimic a refrigerator disturbance, an electric heater internal to the A3 refrigerator was used to generate additional F3 flow, as was described in section 2.6. By evolving gas from an inline reserve of liquid helium residing near the refrigerator outlet, the outlet flow increased affecting P1. P2 also increased due to the increase in fluid density at compressor inlets. For a step in heating rate from 0 to 400 Watts, the F3 and P1 responses at B0 are as shown in Figures 159 and 160. The dotted trace in Figure 160 represents data, while the solid line is the output of the model produced from that data. Note, too, that the time axis is much shorter than other graphs presented in this study. Figure 161 shows the model response at other locations for the 400 Watt step in heater power at A3. One model form found to work, and one that is similar in form to (3.26), is

$$P1(t) = \frac{b18 + b28 \cdot q^{-1}}{1 + f18 \cdot q^{-1} + f28 \cdot q^{-2}} \cdot F3(t-1) + e(t). \tag{4.3}$$

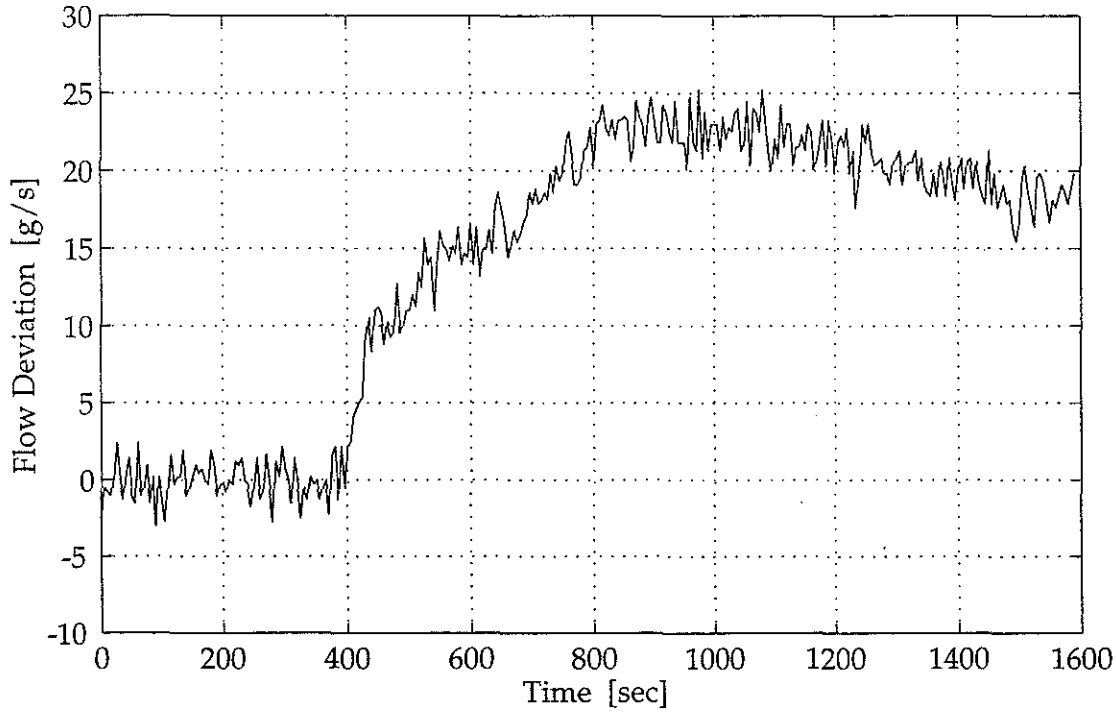

Figure 159. A3 F3 Flow Caused by Heater Step to 400 Watts

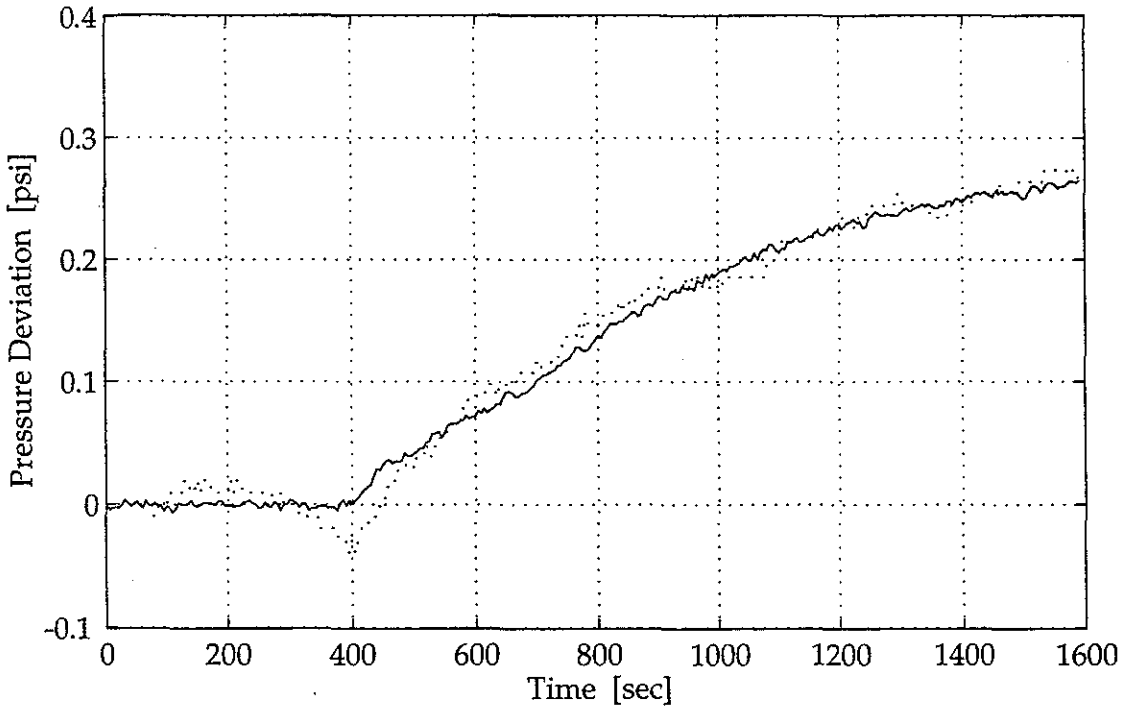

Figure 160. A0 P1 Data and Model for A3 F3 Flow Increase

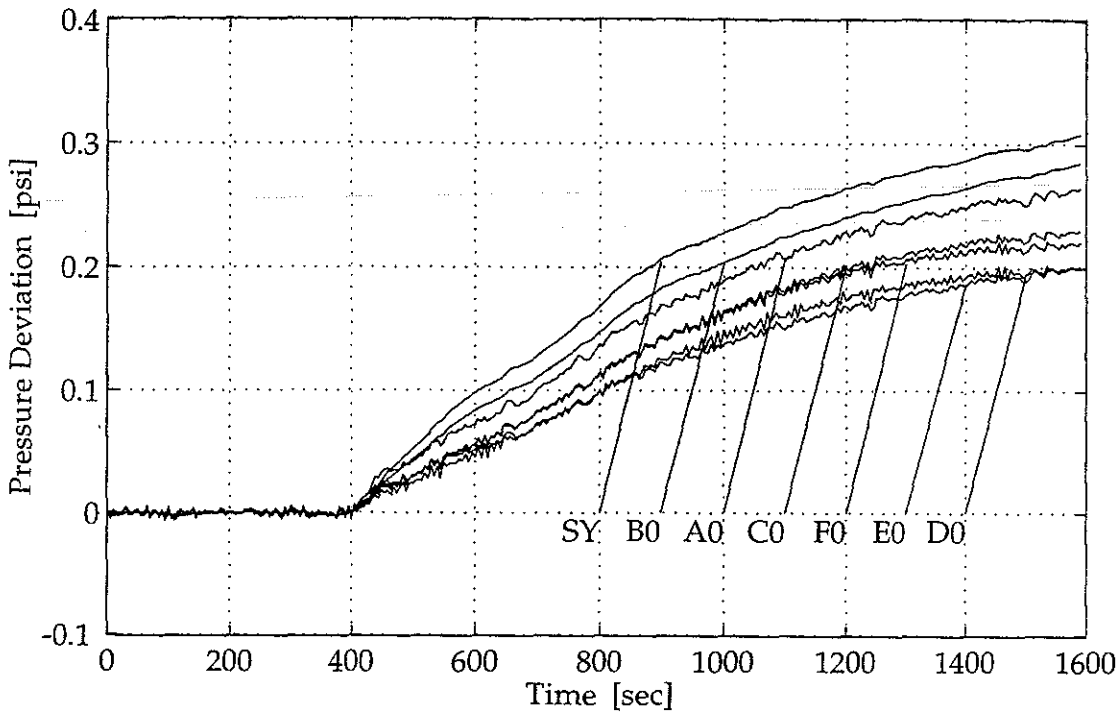

Figure 161. P1 Models for A3 F3 Flow Increase

Table 6. P1 Model Coefficients for A3 F3 Flow Input

Output Location	P1 Coeff.	Coeff. Value	P1 Coeff.	Coeff. Value
A0	b18	+1.8095e-03	f18	-1.3770e+00
	b28	-1.7548e-03	f28	+3.7980e-01
B0	b18	+4.2148e-04	f18	-1.9211e+00
	b28	-4.1726e-04	f28	+9.2119e-01
C0	b18	+1.8221e-03	f18	-8.6812e-01
	b28	-1.7168e-03	f28	-1.2478e-01
D0	b18	+1.8125e-03	f18	-3.3798e-01
	b28	-1.6650e-03	f28	-6.5002e-01
E0	b18	+1.7038e-03	f18	-1.1662e+00
	b28	-1.6541e-03	f28	+1.6916e-01
F0	b18	+2.0129e-03	f18	-5.8186e-01
	b28	-1.8715e-03	f28	-4.0750e-01
SY	b18	+5.1458e-04	f18	-1.9214e+00
	b28	-5.1082e-04	f28	+9.2147e-01

Coefficients for this model are given in Table 6. The size of this disturbance is larger than those seen in Figure 28, but may represent both cases well considering the linearity found during the F2 modeling experience. The heater induced response is much smaller than occurs during a major upset such as a magnet quench, which generates large amounts of gas when superconductivity in the magnet ceases.

For a period of over 24 hour during the data gathering, all regulation of compressors and inventory valves was halted. During this time system pressures followed the ambient pressure quite closely as can be seen in Figures 162 through 164. The affect on P1 for a 20 degree Fahrenheit rise is only about twice the typical steady state disturbances caused by F3. However, since the discharge header is directly exposed to ambient conditions, the effect on P2 is similar in magnitude to turning on a compressor and fully loading it. Note that the data in these figures has been resampled at one minute intervals, and model coefficient calculations assume this as the sample time.

The model form is identical to (4.3) but the coefficients are given different labels. The coefficients based on B0 P1 and B0 P2 data are listed in Table 7. Since ambient temperature is not location dependent, it is assumed that the B0 coefficients model other locations as well. The model does not take into account other atmospheric effects, such as cloud cover, that effect the relationship. Also since only one relatively slow frequency is represented by the input data the model may not hold for fast changes in temperature. It is likely that this model form could be simplified further if data pre-filtering was performed in the modeling process.

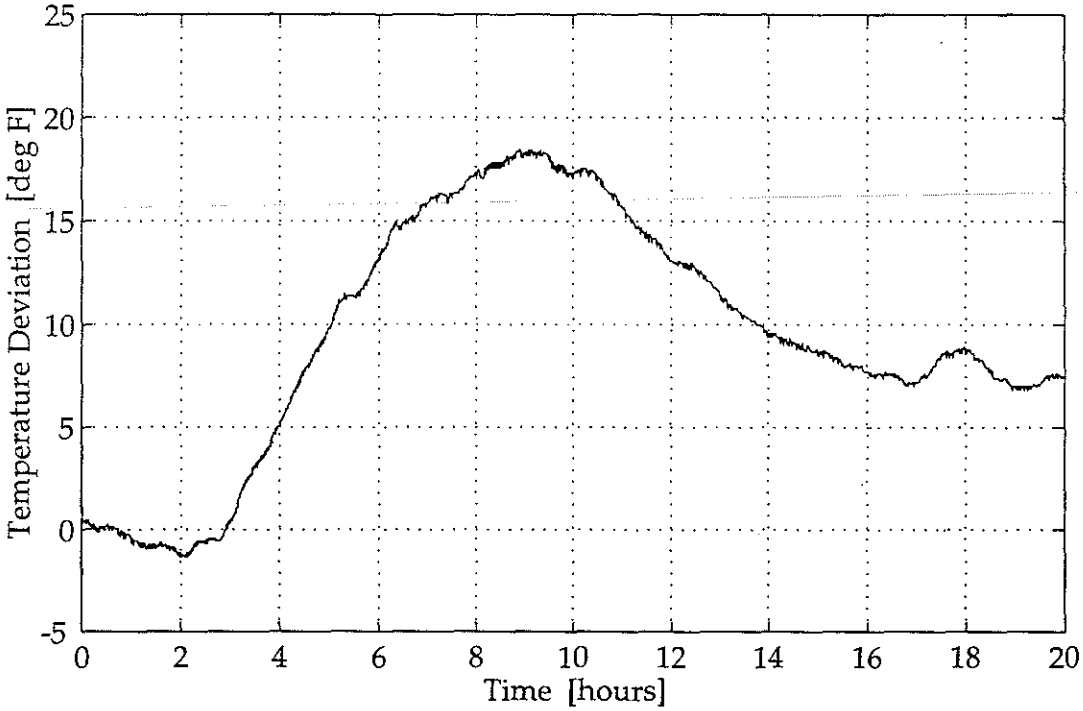

Figure 162. Ambient Temperature Change

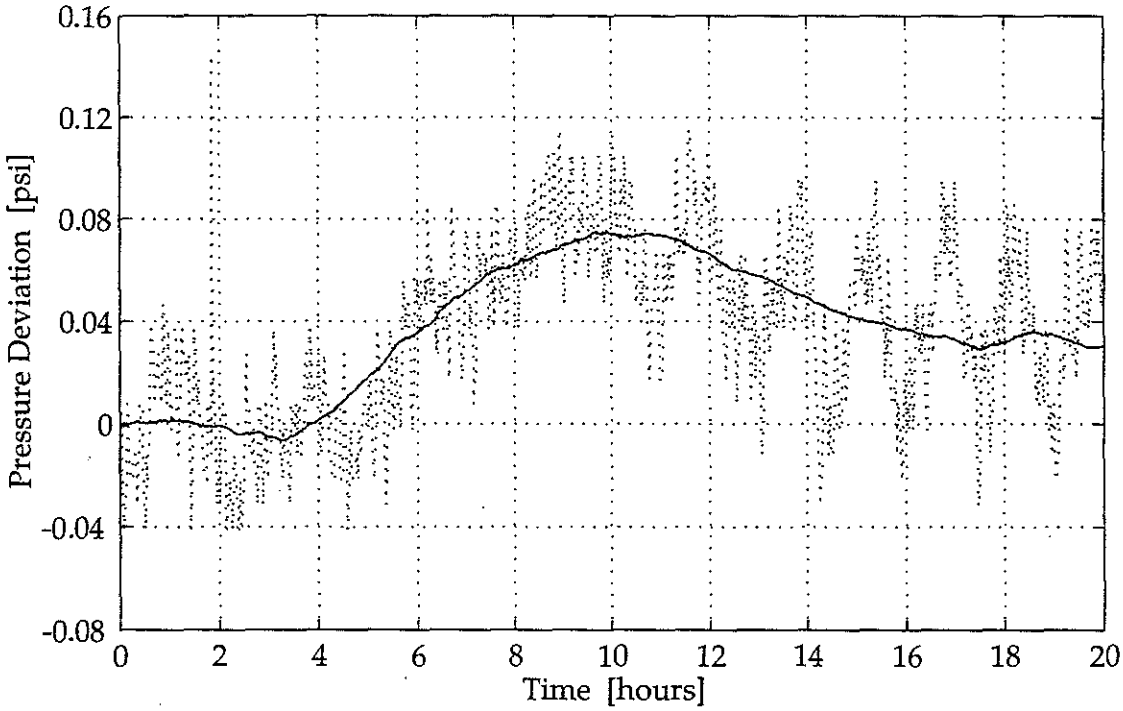

Figure 163. P1 Data and Model for Ambient Temperature Change Input

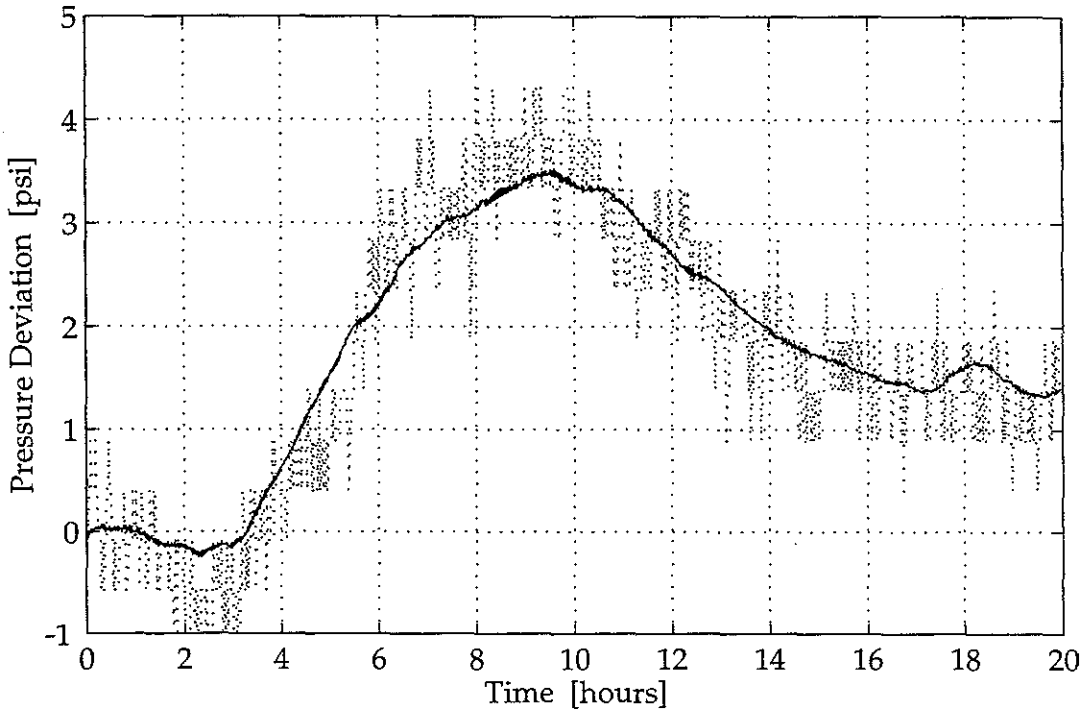

Figure 164. P2 Data and Model for Ambient Temperature Change Input

Table 7. P1 and P2 Model Coefficients for Ambient Temp. Input

Output Location	P1 Coeff.	Coeff. Value	P2 Coeff.	Coeff. Value
All Locations	b19	-2.1785e-03	b10	-1.2421e-01
	b29	+2.2464e-03	b20	+1.4107e-01
	f19	-1.2980e+00	f10	-4.8357e-01
	f29	+3.1446e-01	f20	-4.2796e-01

4.6 Recommendations for Control Improvements

In this section, operation of the control configuration presently in place for compressor and inventory valves is explained, given the insight of this system modeling investigation. Improvements are suggested which require varying degree of effort to implement.

4.6.1 Simple Loop Variable Changes. The key features of the current scheme are regulation of local P2 pressures using local compressor loader valves, and the regulation of the B0 P1 pressure using the inventory valves. Since it was observed that the inventory valves do not directly effect the suction pressure, regulation of P1 occurs indirectly. For instance, once valve B0 CL is stepped open, discharge pressure begins to fall. When it falls sufficiently, the control loops regulating P2s increase one or more compressor valves causing more flow into the discharge header. The equal withdrawal of gas from the suction header causes the value of P1 to drop. To help during the disturbance of a quench, a finite state machine recognizes the quench event and fully opens the inventory valve to quickly transfer to the tank farm the large amount of gas evolved from the magnets.

Due to the large overshoot in P2 at all locations for a change in a single compressor loading, the above scheme causes undesirable interaction among compressor stations. Loading a compressor at one location generally can cause the regulating machines at all other locations to unload. However, when the overshooting P2s come down, these other locations must load up again, causing at best a continuous oscillating behavior. This is indeed seen with P2 varying sinusoidally, typically with a peak amplitude of at least 1.5 psi and often greater. P1 under these conditions is observed to have a peak amplitude of 0.3 psi. The inventory valve, trying to regulate P1, has some effect as well but is too slow to adequately adjust suction pressure. Changing to a configuration in which the compressor valves regulate P1 would eliminate some of this poor behavior since a simpler interaction would result.

Inventory valves would then be used to control discharge pressure in a much more straight forward manner due to the first order relation between the B0 CL and all P2s.

4.6.2 Loop Response Changes. Given the above proposed variable changes in control loop configuration, the system would require tuning to accommodate the new valve functions. With the course compressor valve control presently available, it is unlikely that typical P1 disturbances such as those shown in Figure 28 can be regulated away. Thus, disturbances such as this having amplitude of under 0.1 psi and frequency on the order of 1e-03 hertz, should be ignored. Larger, and perhaps faster, disturbances should be dealt with by the loop, and the quench response finite state machine should be retained. Since the overshoot in P2 with compressor loading is a feature of feedback within the system, it can not be tuned away. Probably the simplest way to deal with these P2 fluctuations is to ignore them. Thus, the inventory valve control loop would not react much to the relatively high frequency pulses associated with compressor loading, but would regulate for more long term changes in the system, such as those resulting from ambient temperature changes.

During certain periods of operation under the current scheme, stability approaches that which might be expected with the above proposed changes. As it happens, when compressors are not regulating because either they are fully loaded and the desired P2 can not be achieved, or they are fully unloaded because too many compressors are turned on, the system is relatively steady. The inventory valve can regulate low frequency disturbances reasonably well

under these conditions, even though it operates on P1 indirectly. With the more direct control proposed, regulation should improve. Either the black box or white box model forms could help accommodate system tuning and simulation, or given sufficient study time, the tuning could be performed on the live system.

4.6.3 Advanced Control Schemes. Many variations in advanced control schemes can be considered when a model is available. Two having significantly different degree of implementation effort are given. A simple scheme for system regulation, which assumes compressors are either on and fully loaded or are off, is likely to provide the most economical operation. It requires global intelligence by the control system in order to schedule compressor turn on based on model predictions of which available compressor will do the most good. The definition of most good might be, for instance, what produces the most level suction pressure profile around the ring. It is most economical from a power standpoint because the compressor motors run at the highest efficiency when fully loaded. In addition, no further attention is needed for compressor valves, other than to assure that they can load upon startup. The drawback is that the control, although very stable, would provide only large flow steps, and the pressure header profiles might be uneven.

Improvement in the hardware that positions compressor valves could be designed to include reliable local mass flow regulation. With this equipment, the mass flow provided by each compressor location would be regulated to a constant value at all but one location, thus eliminating the

effect of suction pressure on compressor throughput. At the last location, the compressor would tightly regulate P1, which would also produce only small compressor influenced changes in P2. In exchange for the small variations in pressures with time resulting from this scheme, relatively large variation of pressure with location might develop. The locational pressure variations could be lessened by choosing appropriate mass flow set points at each of the flow regulated compressor locations. The sum of the flows for the flow regulated locations would presumably be held constant, but the proportioning of that flow sum among locations could be scheduled using the white box model result.

CHAPTER V

SUMMARY AND CONCLUSION

5.1 Summary of Investigation Goals and Techniques

This study has sought to characterize the slowest pressure and flow dynamics associated with the operation of a large distributed refrigeration system operating near its steady state conditions. Limiting the investigation in this way, it was possible to address this primary goal on several levels of detail. Secondary goals included: gauging the robustness of any models developed, deciphering variation in the dynamics with location around the refrigeration complex, obtaining a model for noise in the various signals measured, characterizing disturbances to the system, and providing suggestions for improvement in control of the system.

First principles of fluid mechanics were reviewed in order to gain confidence that available linear modeling techniques could satisfactorily describe the dynamics in a mathematical form. That review suggested that not only would time invariant linear modeling based on data be appropriate, but due to the nature of the piping configuration, it might be the only way to obtain a useful model. The use of first principles alone might not satisfactorily handle the mixed pipeline and reservoir functions of the suction and discharge headers. Experiments to obtain the modeling data were designed to make what was thought to be the best use of the data acquisition system and study time available. Flow inputs were chosen to meet practical concerns about accurate valve positioning and realistic magnitude changes.

The inputs utilized addressed input directional influences as well as locational differences.

5.2 Responses to Controllable Inputs

The study goals were met with varying degrees of success. On the most basic level, a good understanding of the system operation was secured through simple visual review of the experimental data. The feedback of suction pressure resulting in changes to mass flow through compressors has a strong influence on the system, which was not previously recognized. The large overshoot in discharge pressure associated with a step in compressor flow had, in the past, been attributed to oscillations typical of a poorly tuned higher order system. The fluctuation in discharge pressure around the ring, it was discovered, is limited to one excursion followed by a return to nearly the original value at all location except where flow is purposely changed.

Even though the system's internal suction pressure feedback is believed non-linear, linear models were found to satisfactorily describe the overall affect of changing compressor flows. Fitting a black box model to the suction and discharge, however, came after considerable evaluation of what can be practically expected from the routines contained in the System Identification Toolbox. Models for both pressure outputs were achieved only after filtering out all but the lowest component of the flow and pressure signal. Once this was performed, the model forms given in equations (3.21), (3.22), and (3.23) were found to describe all the black box relationships between compressor and inventory flow inputs and pressure outputs for all of the data sets analyzed.

Though the response shapes are the same between data sets using compressor flow inputs, considerable variation exists in specific attributes, such as peak value, settling time, and final value. Averaging model coefficients found for like input locations provides a model that reasonably predicts the response observed for independent validation data sets. One attribute that is not characterized well is the final value of discharge pressure. This is due to the experimental design flaw of insufficient test duration, rather than an inability to model the data. Inventory flow input models agree very well among different data sets.

From the averaged coefficient black box model, a number of relevant observations are drawn. First, there are definite locational trends in pressure response to compressor flow. Pressure measurements located near to the flow change show the most deviation from their original value and those furthest away show the least deviation. This is clear only from the model simulation as these locational differences are typically smaller than the noise in the data signals. Another locational variation appears, with equal compressor flow steps producing slightly greater pressure responses when applied at B0 than when applied at E0 on the opposite side of the ring. For equal step inputs in between these locations the responses are intermediate in magnitude.

Second, the model for compressor flow inputs works equally well for positive and negative flow steps. For inventory flow inputs there may be a difference in the response to negative and positive flow steps, but it is slight and a single model may be used for typical flow changes. It appears that when changing inventory flow by large steps, which changes discharge pressure

significantly, there is a small effect on compressor flow. This compressor flow change acts to produce a discharge pressure lower than the model would predict for decreasing inventory flow, and it acts to produce a pressure higher than predicted for increasing inventory flow. Third, and related to the second observation, is that the model performed well when validated on data sets having inputs greater than those used to construct the model.

The white box model represents a third level of data analysis. Though only limited data sets were investigated in this detail, the results quantify system characteristics that are important to predicting the performance of advanced control schemes. This modeling was premised on producing direct flow to pressure transfer functions, and the ability to separate the pressure response due to the manipulated compressor flow from that due to flow changes caused by suction pressure feedback. Extending this analysis by repeating the modeling on other data sets, or by simply utilizing the symmetry of the system, would result in a very detailed model. The suction pressure feedback introduces non-linear elements to the an overall system description, and further progress with this model form therefore requires the availability of non-linear simulation tools.

5.3 Uses for the Modeling Results

The black box model may be used as is for design and simulation of feedback control. The design may be as simple as optimizing controller coefficients for the current control scheme, or as complex as proposing a multivariable control requiring extensive modification to the control computer systems and software. As noted above, the black box model has not

capture all the influences on the header pressures, and will predict responses from some data sets better than others. The unidentified influences may be random, as is suggested by improvements gained by averaging of model coefficients produced from several data sets. If this is true, there is little that can be done to improve the model short of averaging in additional results. But feedback control by its very nature is forgiving of errors in the model, so properly identifying the nature of the response is perhaps more important than getting a precise model expression.

The secondary goals of identifying and characterizing disturbances and noise originated out of a desire to improve the fit of models for controllable inputs. It became a useful exercise in identifying these signals in their own right, for use in modeling a feedback control response to disturbances and noise. Because it was necessary to filter noise and disturbances during the black box modeling, no noise or disturbance models evolved directly from that process. Computational efforts to identify a noise model during white box modeling resulted only in a characterization of the noise as random. This result is probably consistent with the nature of the input and output signals collected. Identified disturbances are limited to the effects of changes in flows exiting refrigerators, ambient temperature changes, and unintentional changes in compressor flow caused by aging equipment.

The white box modeling suggests that a tighter control of both suction and discharge pressure might result from adding mass flow control to the system. The proposed scheme can be investigated either through simulation using an augmented white box model, or directly on the refrigeration system.

The simulation method has the obvious advantage that any signal connection is possible without breaking real hardware. It has the disadvantage that the model as currently constructed includes several key assumptions that have not been verified experimentally. Thus, the amplitude of the pressure response to the primary flow is not well known prior to performing an experiment to measure and model it directly. Ideally developing such a control becomes an iterative process containing steps of modeling, experimental validation, and model correction.

APPENDIX A

PROGRAMS FOR PRODUCING MODELS AND FIGURES

A.1 Matlab Program ModF2.m

```
% A program to find model of P1 and P2 responding to an initial
% step change in compressor flow via a change in slider valve
% position. For the P1 model, narx1=[na nb nk] is [1 1 1] for arx
% model with one input. For the P2 model, narx2=[na nb nk] is
% [2 2 1] for arx model with one input. Modeling and simulation
% are done with data manipulated to give zero value at first time
% stamp. To achive this, data up to first time stamp is taken as
%the average of values up to that point, and this average is then
% subtracted from all data. The last values of the data are
% averaged and used to extend the data. Program expects data to
% be in a standard format in matrix D.
%
% Following information characteristic of an individual data set
% is saved at end of program as the 19 elements of the 21 element
% vector IND.
data=18;            % data set number
hs=8;               % loader house number (A0=1,...F0=8,SY=9)
LSi=100;            % initial LS loader position in percent
LSf=92.296;         % final LS loader position in percent
Patm=14.503;        % atmospheric pressure in psia
P1o=1.3874;         % initial P1 pressure at stepped loader
ko=90;              % point prior to step input
P=3;                % order of the Butterworth data filter
C1=0.004;           % gives P1 filter cutoff frequency of 0.002 hz
ki1=500;            % initial sample to build extention for P1
kf1=600;            % final sample to build extension for P1
C2=0.004;           % gives P2 filter cutoff frequency of 0.002 hz
ki2=500;            % initial sample to build extention for P2
kf2=600;            % final sample to build extension for P2
L=2400;             % total length of data record created
kx=L-size(D,1);     % length of extension of data points
H=1800;             % number of points in final plot
kn=size(D,1);       % length of data original data record
T=5;                % sample time in seconds
% Find and save P2 pressure at hs, time averaged from data start
% to point ko.
P2o=mean(D(1:ko,9+hs));          % average up to step
%
% Plotting, identification, and result information.
IND=zeros(21,1)                  % storage of input parameters
TH1=zeros(9,13);                 % storage of results for P1
TH2=zeros(9,21);                 % storage of results for P2
time=0:T:(L-1)*T;                % time in seconds
time=time';                      % time as a column vector
```

```
HOUSE=['A0';'B0';'BA';'C0';'D0';'E0';'EA';'F0';'SY';'IN'];
%
% For reference, plot flows, manipulating the data similarly to
% that used for the modeling. Indices must be changed each data
% set. This data is not filtered. Note that inventory flows are
% subtracted because their positive change value effect on P2 is
% opposite to that of positive compressor flow changes.
      flo=zeros(L,3);                 % initialze to zero
      flo(1:L-kx,1:2)=[D(:,hs+18) ...
            (D(:,19)+D(:,20)+D(:,21)+D(:,22)+D(:,23)+ ...
            D(:,24)+D(:,25)+D(:,26)+D(:,27)-D(:,hs+18)-D(:,28)- ...
            D(:,29))];                % raw data
      floo(1,1:2)=mean(flo(1:ko,1:2));   % average up to step
      flo(L-kx+1:L,1)=ones(kx,1)*mean(flo(L-kx-20:L-kx,1));
                                        % build an extention
      flo(L-kx+1:L,2)=ones(kx,1)*mean(flo(L-kx-20:L-kx,2));
      flo(:,1)=flo(:,1)-floo(1,1);      % subtract initial input value
      flo(:,2)=flo(:,2)-floo(1,2);
      flo(:,3)=flo(:,1)+flo(:,2);       % add the two components
%
% Plot the flow data.
pno=input('Enter flow page number: ','s')
figure(1)
set(gcf,'DefaultTextFontName','Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
plot(time(1:H), flo(1:H,1), 'y', time(1:H), flo(1:H,2), 'm')
      ylabel('Flow Deviation  [g/s]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
            axis([0 (H+1)*T -20 20])
            set(gca, 'FontName', 'Palatino')
            fig=input('Enter separate flow figure number: ','s')
            h=text(4500, -28.6, ['Figure ', fig, '. ', HOUSE(hs,1), ...
                  HOUSE(hs,2), ' F2 Flow and FS from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      grid
      h=text(9270, 26.2, pno);
      set(h, 'HorizontalAlignment', 'right')
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
```

```
plot(time(1:H), flo(1:H,3), 'c')
      ylabel('Flow Deviation  [g/s]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
            axis([0 (H+1)*T -20 20])
            set(gca, 'FontName', 'Palatino')
            fig=input('Enter combined flow figure number: ','s')
            h=text(4500, -28.6, ['Figure ', fig, '. Sum of ', ...
                  HOUSE(hs,1), HOUSE(hs,2), ...
                  ' F2 Flow and FS Flow from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      grid
print -dps2
pause
% Flow step size calculation for loader changes.
Pc=15.9;                  % base pressure for loader/flow calibration
C(1:6)=[0.0015928689 -0.40467791 34.59480361 -958.9842469 ...
      5.371698113 -482.3767296];
% for initial loader position
      if LSi < 98
            F2i=C(1)*LSi^3+C(2)*LSi^2+C(3)*LSi+C(4);
            F2i=(F2i*(P1o+Patm)/Pc);
      else
            F2i=C(5)*LSi+C(6);
            F2i=(F2i*(P1o+Patm)/Pc);
      end
% for final loader position
      if LSf < 98
            F2f=C(1)*LSf^3+C(2)*LSf^2+C(3)*LSf+C(4);
            F2f=(F2f*(P1o+Patm)/Pc);
      else
            F2f=C(5)*LSf+C(6);
            F2f=(F2f*(P1o+Patm)/Pc);
      end
step=F2f-F2i                        % initial flow step size
%
% Set up the input. Flow step size taken directly from the data.
      u=zeros(L,1);
      u(ko+1:L,1)=ones(L-ko,1)*step;   % build the step starting at
                                       % point ko+1
% Also build a unit step input to simulate all data sets on an
% equal basis.
```

```
      uu=zeros(L,1);
      uu(ko+1:L,1)=ones(L-ko,1);
%
% Perform model calculations for P1 and P2.
for i=1:9                          % i is an index for location.
% Set up the P1 output.
      y1=zeros(L,1);
      y1(1:L-kx,1)=D(:,i);         % raw data
      y1o=mean(y1(1:ko,1));        % average up to step
      y1(1:ko,1)=ones(ko,1)*y1o;   % set all values up to step
                                   % to equal yo
      y1(kf1+1:L,1)=ones(L-kf1:L,1)*mean(y1(ki1:kf1,1));
                                   % build the extension
      y1=y1-y1o;              % subtract initial value from output
%
% Set up the input/output pair.
      z1=[y1(1:L) u];         % output and input, as defined above
      z1f=idfilt(z1(:,:),P,C1);    % non-causal filter of z
%
% Compute the P1 model and find the sum of the squared errors.
      narx1=[1 1 1];               % [na nb nk] for arx
      tharx1=arx(z1f, narx1, [], T);       % calculate using filtered z
      ysim1(:,i)=idsim([z1(:,2)],tharx1); % simulate using input
      searx1=(ysim1(1:L-kx,i)-z1(1:L-kx,1)).^2;
                                   % sum of squared errors between
      ssearx1=sum(searx1);         % model and unfiltered output
      searxf1=(ysim1(1:L-kx,i)-z1f(1:L-kx,1)).^2;
                                   % sum of squared errors between
      ssearxf1=sum(searxf1);       % model and filtered output
% Also simulate a unit step input.
      ysim1(:,i+9)=idsim(uu(:,1),tharx1);      % uu is a unit step
%
% Look at the zero-pole structure.
[zepo1, K1]=th2zp(tharx1);         % pole and gain
KP1=K1(2,1);                       % gain
[ze1, po1]=getzp(zepo1,1,1);             % pole
fvarx1=(tharx1(3,2))/(1+tharx1(3,1));    % final value
%
% Save 13 items that characterize the results of the P1 modeling
in matrix TH1.
% First save the parameter values and corresponding gain, zero,
and poles.
TH1(i,1)=tharx1(3,1);              % a1
TH1(i,2)=tharx1(3,2);              % b1
TH1(i,3)=K1(2,1);                  % K
TH1(i,4)=zepo1(2,3);               % p1
```

```
% Second, save the standard deviation of above values.
TH1(i,5)=tharx1(4,1);                % a1
TH1(i,6)=tharx1(5,2);                % b1
TH1(i,7)=K1(3,1);                    % K
TH1(i,8)=zepo1(2,4);                 % p1
% Third, save miscellaneous values.
TH1(i,9)=tharx1(1,1);                % variance of the noise
TH1(i,10)=tharx1(2,1);               % Akaike's final prediction error
TH1(i,11)=ssearx1;             % sum of squared errors, unfiltered
TH1(i,12)=ssearxf1;            % sum of the squared errors, filtered
TH1(i,13)=fvarx1;              % final value to unit step input
%
% Set up the P2 output.
      y2=zeros(L,1);
      y2(1:L-kx,1)=D(:,9+i);         % raw data
      y2o=mean(y2(1:ko,1));          % average up to step
      y2(1:ko,1)=ones(ko,1)*y2o;     % set all values up to step to
                                     % equal yo
      y2(L-kx+1:L,1)=ones(kx,1)*mean(y2(ki2:kf2,1));
                                     % build the extension
      y2=y2-y2o;             % subtract initial value from output
%
% Set up the input/output pair.
      z2=[y2(1:L) u];        % output and input, z, defined above
      z2f=idfilt(z2(:,:),P,C2);      % non-causal filter of z
%
% Compute the model and find the sum of the squared errors.
      narx2=[2 2 1];         % [na nb nk] for arx with one input
      tharx2=arx(z2f, narx2, [], T);       % calculate using filtered z
                                           % w/extension
      ysim2(:,i)=idsim([z2(:,2)],tharx2); % simulate using input
      searx2=(ysim2(1:L-kx,i)-z2(1:L-kx,1)).^2;
                                     % sum of squared errors between
      ssearx2=sum(searx2);           % model and unfiltered output
      searxf2=(ysim2(1:L-kx,i)-z2f(1:L-kx,1)).^2;
                                     % sum of squared errors between
      ssearxf2=sum(searxf2);         % model and filtered output
% Also simulate a unit step input.
      ysim2(:,i+9)=idsim(uu(:,1),tharx2);      % uu is a unit step
%
% Look at the zero-pole structure.
[zepo2, K2]=th2zp(tharx2);           % zero, poles, and gain
KP2=K2(2,1);                         % gain
[ze2, po2]=getzp(zepo2,1,1);         % zero and poles
fvarx2=(tharx2(3,3)+tharx2(3,4))/(1+tharx2(3,1)+tharx2(3,2));
                                     % final value
```

```
%
% Save 21 items that characterize the results of the modeling in
% matrix TH2. First save the parameter values and corresponding
% gain, zero, and poles.
TH2(i,1)=tharx2(3,1);                % a1
TH2(i,2)=tharx2(3,2);                % a2
TH2(i,3)=tharx2(3,3);                % b1
TH2(i,4)=tharx2(3,4);                % b2
TH2(i,5)=K2(2,1);                    % K
TH2(i,6)=zepo2(2,1);                 % z1
TH2(i,7)=zepo2(2,3);                 % p1
TH2(i,8)=zepo2(3,3);                 % p2
% Second, save the standard deviation of above values.
TH2(i,9)=tharx2(4,1);                % a1
TH2(i,10)=tharx2(5,2);               % a2
TH2(i,11)=tharx2(6,3);               % b1
TH2(i,12)=tharx2(7,4);               % b2
TH2(i,13)=K2(3,1);                   % K
TH2(i,14)=zepo2(2,2);                % z1
TH2(i,15)=zepo2(2,4);                % p1
TH2(i,16)=zepo2(3,4);                % p2
% Third, save miscellaneous values.
TH2(i,17)=tharx2(1,1);               % variance of the noise
TH2(i,18)=tharx2(2,1);               % Akaike's final prediction error
TH2(i,19)=ssearx2;             % sum of squared errors, unfiltered
TH2(i,20)=ssearxf2;            % sum of the squared errors, filtered
TH2(i,21)=fvarx2;              % final value to unit step input
%
% Plot P1 and P2 data and result at individual locations for the
% flow step.
pno=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ' pg number: '], 's')
figure(2)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
% Plot P1 result.
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
      plot(time(1:H,1), ysim1(1:H,i), 'y-', time(1:H,1), z1f(1:H,1), ...
            'm--', time(1:ko,1), D(1:ko,i)-y1o, ...
            'g:', time(ko+1:L-kx,1), z1(ko+1:L-kx,1), 'g:')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
```

```
        if step > 0                          % for F2 increases
              axis([0 (H+1)*T -0.3001 0.0501])
              set(gca, 'FontName', 'Palatino')
              text(ko*5+500, 0.025, [num2str(step),' g/s Step'], ...
                    'VerticalAlignment','top', ...
                    'HorizontalAlignment','left')
              text(ko*5+500, 0.025, ['Data Set ', num2str(data)], ...
                    'VerticalAlignment','bottom', ...
                    'HorizontalAlignment','left')
              fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                    ' P1 figure number: '],'s')
                    h=text(4500, -0.376, ['Figure ', fig,'.  ', ...
                     HOUSE(i,1), HOUSE(i,2), ...
              ' P1 Data, Filtered Data, and Model for Step in ', ...
                    HOUSE(hs,1), HOUSE(hs,2), ' F2']);
              set(h, 'HorizontalAlignment', 'center')
        h=text(9270, 0.105, pno);
        set(h, 'HorizontalAlignment', 'right')
        else
              axis([0 (H+1)*T -0.0501 0.3001])    % for F2 decreases
              set(gca, 'FontName', 'Palatino')
              text(ko*5+500, -0.025, [num2str(step),' g/s Step'], ...
                    'VerticalAlignment', 'top', ...
                    'HorizontalAlignment', 'left')
              text(ko*5+500, -0.025, ['Data Set ', num2str(data)], ...
                    'VerticalAlignment','bottom', ...
                    'HorizontalAlignment','left')
              fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                    ' P1 figure number: '],'s')
              h=text(4500, -0.126, ['Figure ', fig, '.  ', ...
                    HOUSE(i,1), HOUSE(i,2), ...
              ' P1 Data, Filtered Data, and Model for Step in ', ...
                    HOUSE(hs,1), HOUSE(hs,2), ' F2']);
              set(h, 'HorizontalAlignment', 'center')
        h=text(9270, 0.355, pno);
        set(h, 'HorizontalAlignment', 'right')
        end
        grid
% Plot P2 result.
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
        plot(time(1:H,1), ysim2(1:H,i), 'y-', time(1:H,1), z2f(1:H,1), ...
              'm--', time(1:ko,1), D(1:ko,9+i)-y2o, ...
              'g:', time(ko+1:L-kx,1), z2(ko+1:L-kx,1), 'g:')
        ylabel('Pressure Deviation  [psi]')
        hy=get(gca, 'ylabel');
```

```
        set(hy,'FontName','Palatino')
        xlabel('Time [sec]')
        hx=get(gca,'xlabel');
        set(hx,'FontName','Palatino')
        if step > 0                % for F2 increases
              axis([0 (H+1)*T -0.5 2.5])
              set(gca,'FontName','Palatino')
              text(ko*5+500, -0.25, [num2str(step),' g/s Step'], ...
                    'VerticalAlignment','top', ...
                    'HorizontalAlignment','left')
              text(ko*5+500, -0.25, ['Data Set ', num2str(data)], ...
                    'VerticalAlignment','bottom', ...
                    'HorizontalAlignment','left')
              fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                    ' P2 figure number: '],'s')
              h=text(4500, -1.145, ['Figure ', fig, '. ', ...
                    HOUSE(i,1), HOUSE(i,2), ...
              ' P2 Data, Filtered Data, and Model for Step in ', ...
                    HOUSE(hs,1), HOUSE(hs,2), ' F2']);
              set(h,'HorizontalAlignment','center')
        else
              axis([0 (H+1)*T -2.5 0.5])      % for F2 decreases
              set(gca,'FontName','Palatino')
              text(ko*5+500, 0.25, [num2str(step),' g/s Step'], ...
                    'VerticalAlignment','top', ...
                    'HorizontalAlignment','left')
              text(ko*5+500, 0.25, ['Data Set ', num2str(data)], ...
                    'VerticalAlignment','bottom', ...
                    'HorizontalAlignment','left')
              fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                    ' P2 figure number: '],'s')
              h=text(4500, -3.145, ['Figure ', fig, '. ', ...
                    HOUSE(i,1), HOUSE(i,2), ...
              ' P2 Data, Filtered Data, and Model for Step in ', ...
                    HOUSE(hs,1), HOUSE(hs,2), ' F2']);
              set(h,'HorizontalAlignment','center')
        end
        grid
print -dps2
pause                                % required to see result
end
%
% Plot all the P1 simulations for the step input calculated from
% loader data.
% Sort house names on basis of last value in plot.
[s1,d1]=sort(ysim1(L-kx-ko,1:9));  % d1 has sorted house index
```

```
                                        % based on last data; 1=smallest
pno=input('Enter all models page number: ','s')
figure(3)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
plot(time(1:H,1),ysim1(ko:ko+H-1,1),'y-', ...
        time(1:H,1),ysim1(ko:ko+H-1,2),'m-', ...
        time(1:H,1),ysim1(ko:ko+H-1,3),'c-', ...
        time(1:H,1),ysim1(ko:ko+H-1,4),'r-', ...
        time(1:H,1),ysim1(ko:ko+H-1,5),'g-', ...
        time(1:H,1),ysim1(ko:ko+H-1,6),'b-', ...
        time(1:H,1),ysim1(ko:ko+H-1,7),'w-', ...
        time(1:H,1),ysim1(ko:ko+H-1,8),'r-', ...
        time(1:H,1),ysim1(ko:ko+H-1,9),'g-')
    ylabel('Pressure Deviation  [psi]')
    hy=get(gca, 'ylabel');
    set(hy, 'FontName', 'Palatino')
    xlabel('Time  [sec]')
    hx=get(gca, 'xlabel');
    set(hx, 'FontName', 'Palatino')
    if step > 0              % for F2 increases
        axis([0 (H+1)*T -0.3001 0.0501])
        set(gca, 'FontName', 'Palatino')
        md=(min(ysim1(:,d1(9))))/2;
        for j=1:1:9
            text(3200+(j*550), md, ...
                [HOUSE(d1(j),1), HOUSE(d1(j),2)], ...
                'VerticalAlignment','bottom', ...
                'HorizontalAlignment','right')
            hold on
            plot([3200+(j*550) 3700+(j*550)], ...
                [md ysim1(ko+(3700+j*550)/T, d1(j))])
            text(200, 0.025, ...
                ['Flow Step = ', num2str(step),' g/s'], ...
                'VerticalAlignment','middle', ...
                'HorizontalAlignment','left')
        end
        fig=input('Enter P1 model output figure number: ','s')
        h=text(4500, -0.376, ['Figure ', fig, ...
            '.  P1 Model for Step in ', ...
            HOUSE(hs,1), HOUSE(hs,2), ...
            ' F2 from Data Set ', num2str(data)]);
        set(h, 'HorizontalAlignment', 'center')
    h=text(9270, 0.105, pno);
```

```
        set(h, 'HorizontalAlignment', 'right')
        else
                axis([0 (H+1)*T -0.0501 0.3001])    % for F2 decreases
                set(gca, 'FontName', 'Palatino')
                md=(max(ysim1(:,d1(1))))/2;
                for j=1:1:9
                        text(3200+(j*550), md, ...
                                [HOUSE(d1(10-j),1), HOUSE(d1(10-j),2)], ...
                                'VerticalAlignment','top', ...
                                'HorizontalAlignment','right')
                        hold on
                        plot([3200+(j*550) 3700+(j*550)], ...
                                [md ysim1(ko+(3700+j*550)/T, d1(10-j))])
                        text(200, -0.025, ...
                                ['Flow Step = ', num2str(step),' g/s'], ...
                                'VerticalAlignment','middle', ...
                                'HorizontalAlignment','left')
                end
                fig=input('Enter P1 model output figure number: ','s')
                h=text(4500, -0.126, ...
                        ['Figure ', fig, '.  P1 Model for Step in ', ...
                        HOUSE(hs,1), HOUSE(hs,2), ...
                        ' F2 from Data Set ', num2str(data)]);
                set(h, 'HorizontalAlignment', 'center')
        h=text(9270, 0.355, pno);
        set(h, 'HorizontalAlignment', 'right')
        end
        hold off
        grid
%
% Plot all the P2 models for the step input calculated from loader
% data.
% Sort house names on basis of last value in plot.
[s2,d2]=sort(ysim2(L-kx-ko,1:9));  % d2 has sorted house index
                                % based on time of last data; 1=smallest
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
plot(time(1:H,1),ysim2(ko:ko+H-1,1),'y-', ...
                time(1:H,1),ysim2(ko:ko+H-1,2),'m-', ...
                time(1:H,1),ysim2(ko:ko+H-1,3),'c-', ...
                time(1:H,1),ysim2(ko:ko+H-1,4),'r-', ...
                time(1:H,1),ysim2(ko:ko+H-1,5),'g-', ...
                time(1:H,1),ysim2(ko:ko+H-1,6),'b-', ...
                time(1:H,1),ysim2(ko:ko+H-1,7),'w-', ...
                time(1:H,1),ysim2(ko:ko+H-1,8),'r-', ...
                time(1:H,1),ysim2(ko:ko+H-1,9),'g-')
```

```
ylabel('Pressure Deviation  [psi]')
hy=get(gca, 'ylabel');
set(hy, 'FontName', 'Palatino')
xlabel('Time  [sec]')
hx=get(gca, 'xlabel');
set(hx, 'FontName', 'Palatino')
if step > 0                % for F2 increases
      axis([0 (H+1)*T -0.5 2.5])
      set(gca, 'FontName', 'Palatino')
      md=(max(ysim2(:,d2(9)))+ysim2(1600,d2(9)))/2;
      for j=1:1:9
            text(3200+(j*550), md, ...
                  [HOUSE(d2(10-j),1), HOUSE(d2(10-j),2)], ...
                  'VerticalAlignment','bottom', ...
                  'HorizontalAlignment','right')
            hold on
            plot([3200+(j*550) 3700+(j*550)], ...
                  [md ysim2(ko+(3700+j*550)/T, d2(10-j))])
            text(200, -0.25, ...
                  ['Flow Step = ', num2str(step),' g/s'], ...
                  'VerticalAlignment','middle', ...
                  'HorizontalAlignment','left')
      end
      fig=input('Enter P2 model output figure number: ','s')
      h=text(4500, -1.145, ...
            ['Figure ', fig, '.  P2 Model for Step in ', ...
            HOUSE(hs,1), HOUSE(hs,2), ...
            ' F2 from Data Set ', num2str(data)]);
      set(h, 'HorizontalAlignment', 'center')
else
      axis([0 (H+1)*T -2.5 0.5])     % for F2 decreases
      set(gca, 'FontName', 'Palatino')
      md=(min(ysim2(:,d2(1)))+ysim2(1600,d2(1)))/2;
      for j=1:1:9
            text(3200+(j*550), md , ...
                  [HOUSE(d2(j),1), HOUSE(d2(j),2)], ...
                  'VerticalAlignment','top', ...
                  'HorizontalAlignment','right')
            hold on
            plot([3200+(j*550) 3700+(j*550)], ...
                  [md ysim2(ko+(3700+j*550)/T, d2(j))])
            text(200, 0.25, ...
                  ['Flow Step = ', num2str(step),' g/s'], ...
                  'VerticalAlignment', 'middle', ...
                  'HorizontalAlignment','left')
      end
```

```
            fig=input('Enter P2 model output figure number: ','s')
            h=text(4500, -3.145, ...
                  ['Figure ', fig, '.  P2 Model for Step in ', ...
                  HOUSE(hs,1), HOUSE(hs,2), ...
                  ' F2 from Data Set ', num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      end
      hold off
      grid
print -dps2
pause
%
% Plot all the P1 models for the unit step input.
% Sort house names on basis of last value in plot.
[s1,d1]=sort(ysim1(L-kx-ko,10:18));       % d1 has sorted house
                                          % index
pno=input('Enter all unit step page number: ','s')
figure(4)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
set(gca,'YtickLabelMode','manual')
plot(time(1:H,1),ysim1(ko:ko+H-1,10),'y-', ...
            time(1:H,1),ysim1(ko:ko+H-1,11),'m-', ...
            time(1:H,1),ysim1(ko:ko+H-1,12),'c-', ...
            time(1:H,1),ysim1(ko:ko+H-1,13),'r-', ...
            time(1:H,1),ysim1(ko:ko+H-1,14),'g-', ...
            time(1:H,1),ysim1(ko:ko+H-1,15),'b-', ...
            time(1:H,1),ysim1(ko:ko+H-1,16),'w-', ...
            time(1:H,1),ysim1(ko:ko+H-1,17),'r-', ...
            time(1:H,1),ysim1(ko:ko+H-1,18),'g-')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
            axis([0 (H+1)*T -0.02 0.005])        % F2 increases only
            set(gca, 'FontName', 'Palatino')
            set(gca,'YTickLabels',[-0.020 -0.015 -0.010 -0.005 0 0.005])
            md=(min(ysim1(:,d1(9)+9)))/2;
            for j=1:1:9
                  text(3200+(j*550), md, ...
                        [HOUSE(d1(j),1), HOUSE(d1(j),2)], ...
                        'VerticalAlignment','bottom', ...
```

```
                    'HorizontalAlignment','right')
                hold on
                plot([3200+(j*550) 3700+(j*550)], ...
                    [md ysim1(ko+(3700+j*550)/T, d1(j)+9)])
            end
        hold off
        fig=input('Enter P1 model output figure number: ','s')
        h=text(4500, -0.0255, ['Figure ', fig, ...
            '. P1 Model for Unit Step in ', ...
            HOUSE(hs,1), HOUSE(hs,2), ...
            ' F2 based on Data Set ', num2str(data)]);
        set(h, 'HorizontalAlignment', 'center')
        h=text(9270, 0.0089, pno);
        set(h, 'HorizontalAlignment', 'right')
        grid
%
% Plot all the P2 simulations for the unit step input.
% Sort house names on basis of last value in plot.
[s2,d2]=sort(ysim2(L-kx-ko,10:18));      % d2 has sorted house
                                         % index
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
plot(time(1:H,1),ysim2(ko:ko+H-1,10),'y-', ...
            time(1:H,1),ysim2(ko:ko+H-1,11),'m-', ...
            time(1:H,1),ysim2(ko:ko+H-1,12),'c-', ...
            time(1:H,1),ysim2(ko:ko+H-1,13),'r-', ...
            time(1:H,1),ysim2(ko:ko+H-1,14),'g-', ...
            time(1:H,1),ysim2(ko:ko+H-1,15),'b-', ...
            time(1:H,1),ysim2(ko:ko+H-1,16),'w-', ...
            time(1:H,1),ysim2(ko:ko+H-1,17),'r-', ...
            time(1:H,1),ysim2(ko:ko+H-1,18),'g-')
        ylabel('Pressure Deviation  [psi]')
        hy=get(gca, 'ylabel');
        set(hy, 'FontName', 'Palatino')
        xlabel('Time  [sec]')
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
            axis([0 (H+1)*T -0.04 0.16])          % F2 increases only
            set(gca, 'FontName', 'Palatino')
            set(gca,'YTick',[-0.040 0 0.040 0.080 0.120 0.160])
            set(gca,'YTickLabels',[-0.040 0 0.040 0.080 0.120 0.160])
            md=(max(ysim2(:,d2(9)+9))+ysim2(1600,d2(9)+9))/2;
            for j=1:1:9
                text(3200+(j*550), md, ...
                    [HOUSE(d2(10-j),1), HOUSE(d2(10-j),2)], ...
                    'VerticalAlignment','bottom', ...
```

```
                    'HorizontalAlignment','right')
                hold on
                plot([3200+(j*550) 3700+(j*550)], ...
                    [md ysim2(ko+(3700+j*550)/T, d2(10-j)+9)])
            end
        hold off
        fig=input('Enter P2 model output figure number: ','s')
        h=text(4500, -0.084, ['Figure ', fig, ...
            '. P2 Model for Unit Step in ', ...
            HOUSE(hs,1), HOUSE(hs,2), ...
            ' F2 based on Data Set ', num2str(data)]);
        set(h, 'HorizontalAlignment', 'center')
        grid
print -dps2
%
TH1;                            % display results summary
TH2;                            % display results summary
% Save information used to set up calculations for INDividual
% data set.
IND(1)=data;                % data set number
IND(2)=hs;              % loader house number (A0=1,...F0=8,SY=9)
IND(3)=LSi;                 % initial LS loader position in percent
IND(4)=LSf;                 % final LS loader position in percent
IND(5)=Patm;                % atmospheric pressure in psia
IND(6)=P1o;                 % initial P1 pressure at stepped loader
IND(7)=ko;                  % point prior to step input
IND(8)=P;                   % order of the Butterworth data filter
IND(9)=C1;              % filter cutoff frequency of 0.002 hz for P1
IND(10)=ki1;            % initial sample to build extention for P1
IND(11)=kf1;            % final sample to build extension for P1
IND(12)=C2;             % filter cutoff frequency of 0.002 hz for P2
IND(13)=ki2;            % initial sample to build extention for P2
IND(14)=kf2;            % final sample to build extension for P2
IND(15)=L;                  % total length of data record created
IND(16)=kx;                 % length of extension of data points
IND(17)=H;                  % number of points in final plot
IND(18)=kn;                 % length of original data record
IND(19)=T;                  % sample time in seconds
IND(20)=step;               % flow step size
IND(21)=P2o;                % averaged initial P2 pressure at hs
```

A.2 Matlab Program ModFC.m

```
% A program to find model of P2 responding to an initial step
% change in inventory flow via a change in valve position.
% For the P2 model, narx1=[na nb nk] is [1 1 1] for arx model with
% one input. Modeling and simulation are done with data
% manipulated to give zero value at first time stamp. To achive
% this, data up to first time stamp is taken as the average of
% values up to that point, and this average is then subtracted
% from all data. The last values of the data are averaged and used
% to extend the data tail.
%
% Following information characteristic of an individual data set
% is saved later as the 16 element vector IND. (element 15 holds
% step size calculated below, element 16 has initial P1 pressure)
data=21;                    % data set number
hs=10;                      % house number (always 2 for IN)
EVi=56.951;                 % initial valve position in percent
EVf=61.143;                 % final valve position in percent
Patm=14.380;                % atmospheric pressure in psia
P2o=292.13;                 % initial P2 pressure at valve (B0 P2)
ko=59;                      % point prior to step input
P=3;                        % order of the Butterworth data filter
C2=0.004;                   % P2 filter cutoff frequency 0.002 hz
L=2200;                     % total length of data record created
kx=L-size(D,1);             % length of extension of data points
H=1800;                     % number of points in final plot
kn=size(D,1);               % length of data original data record
T=5;                        % sample time in seconds
%
% Find and save P1 pressure at hs, time averaged from data start
% to ko.
P1o=mean(D(1:ko,2));        % average up to step at B0
%
% Plotting, identification, and result information.
IND=zeros(16,1);            % storage of input parameters
TH2=zeros(9,13);            % storage of results for P2
time=0:T:(L-1)*T;           % time in seconds
time=time';                 % time as a column vector
HOUSE=['A0';'B0';'BA';'C0';'D0';'E0';'EA';'F0';'SY';'IN'];
%
% For reference, plot flows, manipulating the data similarly to
% that used for the modeling. Indices must be changed each data
% set. This data is not filtered.
      flo=zeros(L,4);                    % initialze to zero
      flo(1:L-kx,1:3)=[D(:,28) ...
```

```
            D(:,29) (D(:,19)+D(:,20)+D(:,21)+D(:,22) ...
            +D(:,23)+D(:,24)+D(:,25)+D(:,26)+D(:,27))]; % raw data
      floo(1,1:3)=mean(flo(1:ko,1:3));   % average up to step
      flo(L-kx+1:L,1)=ones(kx,1)*mean(flo(L-kx-20:L-kx,1));
                                         % build an extention
      flo(L-kx+1:L,2)=ones(kx,1)*mean(flo(L-kx-20:L-kx,2));
      flo(L-kx+1:L,3)=ones(kx,1)*mean(flo(L-kx-20:L-kx,3));
      flo(:,1)=flo(:,1)-floo(1,1);       % subtract initial input
                                         % value from all, FC
      flo(:,2)=flo(:,2)-floo(1,2);       % and for FD
      flo(:,3)=flo(:,3)-floo(1,3);       % sum of F2
      flo(:,4)=flo(:,3)-flo(:,1)-flo(:,2);  % combine three
                                   % components for plotting only
%
% Plot the flow data.
pno=input('Enter first flow page number: ','s')
figure(1)
set(gcf,'DefaultTextFontName','Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
plot(time(1:H), flo(1:H,1), 'y')
      ylabel('Flow Deviation  [g/s]')
      hy=get(gca,'ylabel');
      set(hy,'FontName','Palatino')
      xlabel('Time  [sec]')
      hx=get(gca,'xlabel');
      set(hx,'FontName','Palatino')
            axis([0 (H+1)*T -20 20])
            set(gca,'FontName','Palatino')
            fig=input('Enter seperate FC flow figure number: ','s')
            h=text(4500, -28.6, ['Figure ', fig, '.  ', ...
                  ' FC Inventory Flow from Data Set ', ...
                  num2str(data)]);
            set(h,'HorizontalAlignment','center')
      h=text(9270, 26.2, pno);
      set(h,'HorizontalAlignment','right')
      grid
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
plot(time(1:H), flo(1:H,2), 'b')
      ylabel('Flow Deviation  [g/s]')
      hy=get(gca,'ylabel');
      set(hy,'FontName','Palatino')
      xlabel('Time  [sec]')
      hx=get(gca,'xlabel');
```

```
        set(hx, 'FontName', 'Palatino')
              axis([0 (H+1)*T -20 20])
              set(gca, 'FontName', 'Palatino')
              fig=input('Enter seperate FD flow figure number: ','s')
              h=text(4500, -28.6, ['Figure ', fig, '. ', ...
                    ' FD Inventory Flow from Data Set ', ...
                    num2str(data)]);
              set(h, 'HorizontalAlignment', 'center')
        grid
print -dps2
pause
%
pno=input('Enter second flow page number: ','s')
figure(2)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
plot(time(1:H), flo(1:H,3), 'r')
        ylabel('Flow Deviation  [g/s]')
        hy=get(gca, 'ylabel');
        set(hy, 'FontName', 'Palatino')
        xlabel('Time  [sec]')
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
              axis([0 (H+1)*T -20 20])
              set(gca, 'FontName', 'Palatino')
              fig=input( ...
                    'Enter combined F2 flow figure number: ','s')
              h=text(4500, -28.6, ['Figure ', fig, '. ', ...
                    ' Sum of F2 Flows from Data Set ', ...
                    num2str(data)]);
              set(h, 'HorizontalAlignment', 'center')
        h=text(9270, 26.2, pno);
        set(h, 'HorizontalAlignment', 'right')
        grid
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
plot(time(1:H), flo(1:H,4), 'm')
        ylabel('Flow Deviation  [g/s]')
        hy=get(gca, 'ylabel');
        set(hy, 'FontName', 'Palatino')
        xlabel('Time  [sec]')
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
              axis([0 (H+1)*T -20 20])
```

```
            set(gca, 'FontName', 'Palatino')
            fig=input( ...
                  'Enter combined all flow figure number: ','s')
            h=text(4500, -28.6, ['Figure ', fig, '. ', ...
      ' Net Flow Change into Discharge Header from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      grid
print -dps2
pause
%
% Calculate step size directly from data. Average values up to step
% for initial flow, and skip five sample times for step to settle
% prior to evaluating final flow as an average of 25 values.
      step=mean(D(ko+5:ko+30,28))-mean(D(1:ko,28));
% Set up the input. Flow step size taken directly from the data.
      u=zeros(L,1);
      u(ko+1:L,1)=ones(L-ko,1)*step;   % build the step starting at
                                       % point ko+1
% Also build a unit step input to simulate all data sets on an
% equal basis.
      uu=zeros(L,1);
      uu(ko+1:L,1)=ones(L-ko,1);
%
% Perform model calculations for P2.
ysim2(:,:)=zeros(L,18);
for i=1:9                              % i is an index for location.
% Set up the P1 output to plot as reference later.
      y1=zeros(L-kx,1);
      y1(1:L-kx,1)=D(:,i);             % raw data
      y1o=mean(y1(1:ko,1));            % average up to step
      y1(1:ko,1)=ones(ko,1)*y1o;       % set all values up to step
                                       % to equal yo
      y1=y1-y1o;        % subtract initial value from all output
% Set up the P2 output.
      y2=zeros(L-kx,1);
      y2(1:L-kx,1)=D(:,i+9);           % raw data
      y2o=mean(y2(1:ko,1));            % average up to step
      y2(1:ko,1)=ones(ko,1)*y2o;       % set all values up to step
                                       % to equal yo
      y2=y2-y2o;        % subtract initial value from all output
%
% Set up the input/output pair.
      z2=[y2(:) u(1:L-kx)];            % output and input, above
      z2f=idfilt(z2(:,:),P,C2);        % non-causal filter of z
%
```

```
% Compute the P2 model and find the sum of the squared errors.
      narx2=[1 1 1];                          % [na nb nk] for arx
      tharx2=arx(z2f, narx2, [], T);          % calculate w/o extension
      ysim2(:,i)=idsim(u,tharx2);             % simulate using input
      searx2=(ysim2(1:L-kx,i)-z2(:,1)).^2;        % sum of squared
                                              % errors between model
      ssearx2=sum(searx2);                    % and unfiltered output
      searxf2=(ysim2(1:L-kx,i)-z2f(:,1)).^2;      % sum of squared
                                              % errors between model
      ssearxf2=sum(searxf2);                  % and filtered output
% Also simulate a unit step input.
      ysim2(:,i+9)=idsim(uu(:,1),tharx2);         % uu is a unit step
%
% Look at the zero-pole structure.
[zepo2, K2]=th2zp(tharx2);                    % pole and gain
KP2=K2(2,1);                                  % gain
[ze2, po2]=getzp(zepo2,1,1);                  % pole
fvarx2=(tharx2(3,2))/(1+tharx2(3,1));         % final value
%
% Save 13 items that characterize the results of the P1 modeling
in matrix TH1.
% First save the parameter values and corresponding gain, zero,
and poles.
TH2(i,1)=tharx2(3,1);                 % a1
TH2(i,2)=tharx2(3,2);                 % b1
TH2(i,3)=K2(2,1);                     % K
TH2(i,4)=zepo2(2,3);                  % p1
% Second, save the standard deviation of above values.
TH2(i,5)=tharx2(4,1);                 % a1
TH2(i,6)=tharx2(5,2);                 % b1
TH2(i,7)=K2(3,1);                     % K
TH2(i,8)=zepo2(2,4);                  % p1
% Third, save miscellaneous values.
TH2(i,9)=tharx2(1,1);                 % variance of the noise
TH2(i,10)=tharx2(2,1);                % Akaike's final prediction error
TH2(i,11)=ssearx2;      % sum of the squared errors, unfiltered
TH2(i,12)=ssearxf2;           % sum of the squared errors, filtered
TH2(i,13)=fvarx2;                     % final value to unit step input
%
% Plot P1 data just for reference.
pno=input( ...
      ['Enter ', HOUSE(i,1), HOUSE(i,2), ' page number: '] ,'s')
figure(3)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
subplot(2,1,1)
```

```
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
      plot(time(1:ko,1), D(1:ko,i)-y1o, 'g:', time(ko+1:L-kx,1), ...
            y1(ko+1:L-kx,1), 'g:')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
            axis([0 (H+1)*T -0.1501 0.1501])
            set(gca, 'FontName', 'Palatino')
            text(ko*5+1000, -0.125, [num2str(step),' g/s Step'], ...
                  'VerticalAlignment','top', ...
                  'HorizontalAlignment','left')
            text(ko*5+1000, -0.125, ['Data Set ', num2str(data)], ...
                  'VerticalAlignment','bottom', ...
                  'HorizontalAlignment','left')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                  ' P1 figure number: '],'s')
            h=text(4500, -0.215, ['Figure ', fig, '.  ', ...
                  HOUSE(i,1), HOUSE(i,2), ...
                  ' P1 Data for Step in FC']);
            set(h, 'HorizontalAlignment', 'center')
      h=text(9270, 0.197, pno);
      set(h, 'HorizontalAlignment', 'right')
      grid
% Plot P2 data and result at individual locations for the flow step.
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
      plot(time(1:H,1), ysim2(1:H,i), 'y-', time(1:L-kx,1), ...
            z2f(1:L-kx,1), 'm--', time(1:ko,1), D(1:ko,9+i)-y2o, ...
            'g:', time(ko+1:L-kx,1), z2(ko+1:L-kx,1), 'g:')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
      if step < 0                % for FC decreases
            axis([0 (H+1)*T -2 16])
            set(gca, 'FontName', 'Palatino')
            set(gca,'YTick',[-2 0 2 4 6 8 10 12 14 16])
            set(gca,'YTickLabels',[-2 0 2 4 6 8 10 12 14 16])
            text(ko*5+1000, 1.0, [num2str(step),' g/s Step'], ...
                  'VerticalAlignment','top', ...
                  'HorizontalAlignment','left')
```

```
            text(ko*5+1000, 1.0, ['Data Set ', num2str(data)], ...
                  'VerticalAlignment','bottom', ...
                  'HorizontalAlignment','left')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                  ' P2 figure number: '],'s')
            h=text(4500, -5.90, ['Figure ', fig, '. ', ...
                  HOUSE(i,1), HOUSE(i,2), ...
            ' P2 Data, Filtered Data, and Model for Step in FC']);
            set(h, 'HorizontalAlignment', 'center')
      else
            axis([0 (H+1)*T -16 2])        % for FC increases
            set(gca, 'FontName', 'Palatino')
            set(gca,'YTick',[-16 -14 -12 -10 -8 -6 -4 -2 0 2])
            set(gca,'YTickLabels',[-16 -14 -12 -10 -8 -6 -4 -2 0 2])
            text(ko*5+1000, -1.0, [num2str(step),' g/s Step'], ...
                  'VerticalAlignment','top', ...
                  'HorizontalAlignment','left')
            text(ko*5+1000, -1.0, ['Data Set ', num2str(data)], ...
                  'VerticalAlignment','bottom', ...
                  'HorizontalAlignment','left')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                  ' P2 figure number: '],'s')
            h=text(4500, -19.90, ['Figure ', fig, '. ', ...
                  HOUSE(i,1), HOUSE(i,2), ...
            ' P2 Data, Filtered Data, and Model for Step in FC']);
            set(h, 'HorizontalAlignment', 'center')
      end
      grid
print -dps2
pause                                  % required to see result
end
%
% Plot all the P2 simulations for step input calculated from data.
% Sort house names on basis of last value in plot.
[s2,d2]=sort(ysim2(ko+H-1,1:9));
pno=input('Enter all P2 models and unit step page number: ','s')
figure(4)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
plot(time(1:H,1),ysim2(ko:ko+H-1,1),'y-', ...
            time(1:H,1),ysim2(ko:ko+H-1,2),'m-', ...
            time(1:H,1),ysim2(ko:ko+H-1,3),'c-', ...
            time(1:H,1),ysim2(ko:ko+H-1,4),'r-', ...
            time(1:H,1),ysim2(ko:ko+H-1,5),'g-', ...
```

```
        time(1:H,1),ysim2(ko:ko+H-1,6),'b-', ...
        time(1:H,1),ysim2(ko:ko+H-1,7),'w-', ...
        time(1:H,1),ysim2(ko:ko+H-1,8),'r-', ...
        time(1:H,1),ysim2(ko:ko+H-1,9),'g-')
ylabel('Pressure Deviation  [psi]')
hy=get(gca,'ylabel');
set(hy,'FontName','Palatino')
xlabel('Time  [sec]')
hx=get(gca,'xlabel');
set(hx,'FontName','Palatino')
if step < 0                 % for FC decreases
        axis([0 (H+1)*T -2 16])
        set(gca,'FontName','Palatino')
        set(gca,'YTick',[-2 0 2 4 6 8 10 12 14 16])
        set(gca,'YTickLabels',[-2 0 2 4 6 8 10 12 14 16])
        md=(max(ysim2(:,d2(9))))/2;
        for j=1:1:9
                text(3200+(j*550), md , ...
                        [HOUSE(d2(10-j),1), HOUSE(d2(10-j),2)], ...
                        'VerticalAlignment','top', ...
                        'HorizontalAlignment','right')
                hold on
                plot([3200+(j*550) 3700+(j*550)], ...
                        [md ysim2(ko+(3700+j*550)/T, d2(10-j))])
                text(200, -1, ...
                        ['Flow Step = ', num2str(step),' g/s'], ...
                        'VerticalAlignment','middle', ...
                        'HorizontalAlignment','left')
        end
        fig=input('Enter P2 model output figure number: ','s')
        h=text(4500, -5.90, ['Figure ', fig, ...
                '.  P2 Model for Step in FC from Data Set ', ...
                num2str(data)]);
        set(h,'HorizontalAlignment','center')
h=text(9270, 18.8, pno);
set(h,'HorizontalAlignment','right')
else
        axis([0 (H+1)*T -16 2])         % for FC increases
        set(gca,'FontName','Palatino')
        set(gca,'YTick',[-16 -14 -12 -10 -8 -6 -4 -2 0 2])
        set(gca,'YTickLabels',[-16 -14 -12 -10 -8 -6 -4 -2 0 2])
        md=(min(ysim2(:,d2(9))))/2;
        for j=1:1:9
                text(3200+(j*550), md, ...
                        [HOUSE(d2(j),1), HOUSE(d2(j),2)], ...
                        'VerticalAlignment','bottom', ...
```

```
                    'HorizontalAlignment','right')
                hold on
                plot([3200+(j*550) 3700+(j*550)], ...
                    [md ysim2(ko+(3700+j*550)/T, d2(j))])
                text(200, 1, ...
                    ['Flow Step = ', num2str(step),' g/s'], ...
                    'VerticalAlignment','middle', ...
                    'HorizontalAlignment','left')
            end
            fig=input('Enter P2 model output figure number: ','s')
            h=text(4500, -19.90, ['Figure ', fig, ...
                '. P2 Model for Step in FC from Data Set ', ...
                num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
        h=text(9270, 4.8, pno);
        set(h, 'HorizontalAlignment', 'right')
        end
        hold off
        grid
%
% Plot all the P2 simulations for the unit step input.
% Sort house names on basis of last value in plot.
[s2,d2]=sort(ysim2(ko+H-1,10:18));
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
plot(time(1:H,1),ysim2(ko:ko+H-1,10),'y-', ...
            time(1:H,1),ysim2(ko:ko+H-1,11),'m-', ...
            time(1:H,1),ysim2(ko:ko+H-1,12),'c-', ...
            time(1:H,1),ysim2(ko:ko+H-1,13),'r-', ...
            time(1:H,1),ysim2(ko:ko+H-1,14),'g-', ...
            time(1:H,1),ysim2(ko:ko+H-1,15),'b-', ...
            time(1:H,1),ysim2(ko:ko+H-1,16),'w-', ...
            time(1:H,1),ysim2(ko:ko+H-1,17),'r-', ...
            time(1:H,1),ysim2(ko:ko+H-1,18),'g-')
        ylabel('Pressure Deviation  [psi]')
        hy=get(gca, 'ylabel');
        set(hy, 'FontName', 'Palatino')
        xlabel('Time  [sec]')
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
            axis([0 (H+1)*T -2 0.4])        % for FC increases
            set(gca, 'FontName', 'Palatino')
            set(gca,'YTick',[-2.0 -1.60 -1.20 -0.80 -0.40 0 0.40])
            set(gca,'YTickLabels',[-2.0 -1.60 -1.2 -0.80 -0.40 0 0.40])
            md=(min(ysim2(:,d2(9)+9)))/2;
            for j=1:1:9
```

```
                text(3200+(j*550), md, ...
                      [HOUSE(d2(j),1), HOUSE(d2(j),2)], ...
                      'VerticalAlignment','bottom', ...
                      'HorizontalAlignment','right')
                hold on
                plot([3200+(j*550) 3700+(j*550)], ...
                      [md ysim2(ko+(3700+j*550)/T, d2(j)+9)])
            end
            fig=input('Enter P2 model output figure number: ','s')
            h=text(4500, -2.53, ['Figure ', fig, ...
            '. P2 Model for Unit Step in FC Based on Data Set ', ...
                num2str(data)]);
            set(h,'HorizontalAlignment','center')
        hold off
        grid
print -dps2
%
IND(1)=data;                % data set number
IND(2)=hs;                  % valve house number (always IN=10)
IND(3)=EVi;                 % initial valve position in percent
IND(4)=EVf;                 % final valve position in percent
IND(5)=Patm;                % atmospheric pressure in psia
IND(6)=P2o;                 % initial P2 pressure at valve (B0 P2)
IND(7)=ko;                  % point prior to step input
IND(8)=P;                   % order of the Butterworth data filter
IND(9)=C2;                  % P2 filter cutoff frequency 0.002 hz
IND(10)=L;                  % total length of data record created
IND(11)=kx;                 % length of extension of data points
IND(12)=H;                  % number of points in final plot
IND(13)=kn;                 % length of data original data record
IND(14)=T;                  % sample time in seconds
IND(15)=step;               % flow step size calculated from data
IND(16)=P1o;                % averaged initial P1 pressure at B0
```

A.3 Matlab Program SimF2.m

```
% A program to simulate system response using models
% developed for compressor flow changes. A data set is used for
% comparision, with an initial flow step size being the only item
% calculated from that data set. Model coefficients in matrix C are
% assumed to be loaded into the session, and the P1 and P2 data
% come from matrix D.
%
data=25;                    % data set number
hs=8;                       % loader house no (A0=1,...F0=8,SY=9)
LSi=100;                    % initial LS loader position in percent
LSf=0;                      % final LS loader position in percent
Patm=14.468;                % atmospheric pressure in psia
P1o=1.3776;                 % initial P1 pressure at stepped loader
ko=121;                     % point prior to step input
L=2400;                     % total length of data record created
kx=L-size(D,1);             % length of simulation beyond data
H=1800;                     % number of points in final plot
kn=size(D,1);               % length of data original data record
T=5;                        % sample time in seconds
% Find and save P2 pressure at hs, time averaged from data start
% to ko.
P2o=mean(D(1:ko,9+hs));          % average up to step
%
% Plotting, identification, and result information.
IND=zeros(14,1);            % storage of input parameters and sse
time=0:5:(L-1)*5;           % time in seconds
time=time';                 % time as a column vector
SSE=zeros(9,2);             % storage of sum of squared errors
HOUSE=['A0';'B0';'BA';'C0';'D0';'E0';'EA';'F0';'SY';'IN'];
%
% Map BA and EA locations to use the model coefficients for B0
% and E0.
% SY is not modeled as an input house.
if hs == 1 inloc=1;
elseif hs == 2 iloc=2;
elseif hs == 3 iloc=2;
elseif hs == 4 iloc=3;
elseif hs == 5 iloc=4;
elseif hs == 6 iloc=5;
elseif hs == 7 iloc=5;
elseif hs == 8 iloc=6;
end
%
% For reference, plot flows, manipulating the data similarly to
```

```
% that used for the modeling. Indices must be changed each data
% set. This data is not filtered. Note that inventory flows are
% subtracted because their positive change value effect on P2 is
% opposite to that of positive compressor flow changes.
      flo=zeros(L,3);                          % initialze to zero
      flo(1:L-kx,1:2)=[D(:,hs+18) ...
            (D(:,19)+D(:,20)+D(:,21)+D(:,22)+D(:,23)+ ...
            D(:,24)+D(:,25)+D(:,26)+D(:,27)-D(:,hs+18)-D(:,28)- ...
            D(:,29))];                         % raw data
      floo(1,1:2)=mean(flo(1:ko,1:2));         % average up to step
      flo(L-kx+1:L,1)=ones(kx,1)*mean(flo(L-kx-20:L-kx,1));
                                               % build an extention
      flo(L-kx+1:L,2)=ones(kx,1)*mean(flo(L-kx-20:L-kx,2));
      flo(:,1)=flo(:,1)-floo(1,1);             % subtract initial input
                                               %value from all
      flo(:,2)=flo(:,2)-floo(1,2);
      flo(:,3)=flo(:,1)+flo(:,2);              % add the two components
                                               % for plotting only
%
% Plot the flow data.
pno=input('Enter flow page number: ','s')
figure(1)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
plot(time(1:H), flo(1:H,1), 'y', time(1:H), flo(1:H,2), 'm')
      ylabel('Flow Deviation  [g/s]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
            axis([0 (H+1)*T -60 60])
            set(gca, 'FontName', 'Palatino')
            fig=input('Enter seperate flow figure number: ','s')
            h=text(4500, -85.8, ['Figure ', fig, '.  ', ...
                  HOUSE(hs,1), HOUSE(hs,2), ...
                  ' F2 Flow and Secondary Flow from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      h=text(9270, 78.6, pno);
      set(h, 'HorizontalAlignment', 'right')
      grid
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
```

```
plot(time(1:H), flo(1:H,3), 'c')
      ylabel('Flow Deviation  [g/s]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
            axis([0 (H+1)*T -60 60])
            set(gca, 'FontName', 'Palatino')
            fig=input('Enter combined flow figure number: ','s')
            h=text(4500, -85.8, ['Figure ', fig, '.  Sum of ', ...
                  HOUSE(hs,1), HOUSE(hs,2), ...
                  ' F2 Flow and Secondary Flow from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      grid
print -dps2
pause
%
% Flow step size calculation used for loader changes.
Pc=15.9;
a=0.0015928689;
b=-0.40467791;
c=34.59480361;
d=-958.9842469;
e=5.371698113;
f=-482.3767296;
% for initial loader position:
      if LSi < 80                    % if loader is less than 80
            F2i=0;                   % assume compressor turn on
      elseif LSi < 98
            F2i=a*LSi^3+b*LSi^2+c*LSi+d;
            F2i=(F2i*(P1o+Patm)/Pc);
      else
            F2i=e*LSi+f;
            F2i=(F2i*(P1o+Patm)/Pc);
      end
% for final loader position:
      if LSf < 80                    % if loader is less than 80%
            F2f=0;                   % assume compressor turn off
      elseif LSf < 98
            F2f=a*LSf^3+b*LSf^2+c*LSf+d;
            F2f=(F2f*(P1o+Patm)/Pc);
      else
            F2f=e*LSf+f;
            F2f=(F2f*(P1o+Patm)/Pc);
```

```
    end
simstep=F2f-F2i                 % initial step size in flow,
u=zeros(L, 1);                  % set up the input by building
u(ko+1:L,1)=ones(L-ko,1)*simstep;     % the step starting at ko+1
%
% Create model theta from previously determined coefficients
% residing in matrix C.
ysimP1=zeros(L,1);
ysimP2=zeros(L,1);
for i=1:9
    P1a1=C(6*(iloc-1)+1,i);
    P1b1=C(6*(iloc-1)+2,i);
    P2a1=C(6*(iloc-1)+3,i);
    P2a2=C(6*(iloc-1)+4,i);
    P2b1=C(6*(iloc-1)+5,i);
    P2b2=C(6*(iloc-1)+6,i);
    A1=[1 P1a1];
    B1=[0 P1b1];
    A2=[1 P2a1 P2a2];
    B2=[0 P2b1 P2b2];
    thP1=poly2th(A1,B1);
    thP2=poly2th(A2,B2);
% Detrend pressure to the average of values prior to flow step.
    yP1o=mean(D(1:ko,i));       % average up to step for P1
    yP1(:,i)=D(:,i)-yP1o;    % subtract initial value from output
    yP2o=mean(D(1:ko,i+9));     % average up to step for P2
    yP2(:,i)=D(:,i+9)-yP2o; % subtract initial value from output
% Simulate the P1 response to a step.
    ysimP1(:,i)=idsim(u,thP1);  % simulate P1 using input
    seP1=(ysimP1(1:L-kx,i)-yP1(1:L-kx,i)).^2;      % sum of
                                % squared errors between model
    SSE(i,1)=sum(seP1);         % and unfiltered input, P1
% Simulate the P2 response to a step.
    ysimP2(:,i)=idsim(u,thP2);  % simulate P2 using input
    seP2=(ysimP2(1:L-kx,i)-yP2(1:L-kx,i)).^2;      % sum of
                                % squared errors between model
    SSE(i,2)=sum(seP2);         % and unfiltered output, P2
%
% Plot P1 and P2 data and simulation at individual locations for
% the flow step.
pno=input( ...
    ['Enter ', HOUSE(i,1), HOUSE(i,2), ' page number: '] ,'s')
figure(2)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
% Plot P1 result.
```

```
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
      plot(time(1:H,1), ysimP1(1:H,i), 'y-', ...
            time(1:L-kx,1), D(1:L-kx,i)-yP1o, 'g:')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
      if simstep > 0
            axis([0 (H+1)*T -0.800 0.100])       % for F2 increases
            set(gca, 'FontName', 'Palatino')
            set(gca,'YTick', ...
                  [-0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0.0 0.1])
      set(gca,'YTickLabels', ...
                  [-0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0.0 0.1])
            text(ko*5+500, 0.05, [num2str(simstep),' g/s Step'], ...
                  'VerticalAlignment','top',
'HorizontalAlignment','left')
            text(ko*5+500, 0.05, ['Data Set ', num2str(data)], ...
                  'VerticalAlignment','bottom', ...
                  'HorizontalAlignment','left')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                  ' P1 figure number: '],'s')
            h=text(4500, -0.995, ['Figure ', fig, '.  ', ...
                  HOUSE(i,1), HOUSE(i,2), ...
                  ' P1 Data and Simulation for Step in ', ...
                  HOUSE(hs,1), HOUSE(hs,2), ' F2']);
            set(h, 'HorizontalAlignment', 'center')
      h=text(9270, 0.241, pno);
      set(h, 'HorizontalAlignment', 'right')
      else
            axis([0 (H+1)*T -0.100 0.800])       % for F2 decreases
            set(gca, 'FontName', 'Palatino')
            set(gca,'YTick', ...
                  [-0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8])
            set(gca,'YTickLabels', ...
                  [-0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8])
            text(ko*5+500, -0.05, [num2str(simstep),' g/s Step'], ...
                  'VerticalAlignment','top', ...
                  'HorizontalAlignment','left')
            text(ko*5+500, -0.05, ['Data Set ', num2str(data)], ...
                  'VerticalAlignment','bottom', ...
                  'HorizontalAlignment','left')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
```

```
                ' P1 figure number: '],'s')
            h=text(4500, -0.290, ['Figure ', fig, '. ', ...
                HOUSE(i,1), HOUSE(i,2), ...
                ' P1 Data and Simulation for Step in ', ...
                HOUSE(hs,1), HOUSE(hs,2), ' F2']);
            set(h, 'HorizontalAlignment', 'center')
        h=text(9270, 0.941, pno);
        set(h, 'HorizontalAlignment', 'right')
        end
        grid
% Plot P2 result.
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
        plot(time(1:H,1), ysimP2(1:H,i), 'y-', ...
            time(1:L-kx,1), D(1:L-kx,9+i)-yP2o, 'g:')
        ylabel('Pressure Deviation  [psi]')
        hy=get(gca, 'ylabel');
        set(hy, 'FontName', 'Palatino')
        xlabel('Time  [sec]')
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
        if simstep > 0
            axis([0 (H+1)*T -2 7])              % for F2 increases
            set(gca, 'FontName', 'Palatino')
            set(gca,'YTick',[-2 -1 0 1 2 3 4 5 6 7])
            set(gca,'YTickLabels',[-2 -1 0 1 2 3 4 5 6 7])
            text(ko*5+500, -0.5, [num2str(simstep),' g/s Step'], ...
                'VerticalAlignment','top', ...
                'HorizontalAlignment','left')
            text(ko*5+500, -0.5, ['Data Set ', num2str(data)], ...
                'VerticalAlignment','bottom', ...
                'HorizontalAlignment','left')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                ' P2 figure number: '],'s')
            h=text(4500, -3.95, ['Figure ', fig, '. ', ...
                HOUSE(i,1), HOUSE(i,2), ...
                ' P2 Data and Simulation for Step in ', ...
                HOUSE(hs,1), HOUSE(hs,2), ' F2']);
            set(h, 'HorizontalAlignment', 'center')
        else
            axis([0 (H+1)*T -7 2])              % for F2 decreases
            set(gca, 'FontName', 'Palatino')
            set(gca,'YTick',[-7 -6 -5 -4 -3 -2 -1 0 1 2])
            set(gca,'YTickLabels',[-7 -6 -5 -4 -3 -2 -1 0 1 2])
            text(ko*5+500, 0.25, [num2str(simstep),' g/s Step'], ...
                'VerticalAlignment','top', ...
```

```
                'HorizontalAlignment','left')
            text(ko*5+500, 0.25, ['Data Set ', num2str(data)], ...
                'VerticalAlignment','bottom', ...
                'HorizontalAlignment','left')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                ' P2 figure number: '],'s')
            h=text(4500, -8.90, ['Figure ', fig, '. ', ...
                HOUSE(i,1), HOUSE(i,2), ...
                ' P2 Data and Simulation for Step in ', ...
                HOUSE(hs,1), HOUSE(hs,2), ' F2']);
            set(h, 'HorizontalAlignment', 'center')
        end
        grid
pause                                   % required to see result
print -dps2
end
% Plot all the P1 simulations for the step input calculated from
% loader data. Sort house names on basis of last value in plot.
[s1,d1]=sort(ysimP1(L-kx-ko,1:9));      % d1 has sorted house
                                % index based on last data; 1=smallest
pno=input('Enter all models page number: ','s')
figure(3)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
plot(time(1:H,1),ysimP1(ko:ko+H-1,1),'y-', ...
        time(1:H,1),ysimP1(ko:ko+H-1,2),'m-', ...
        time(1:H,1),ysimP1(ko:ko+H-1,3),'c-', ...
        time(1:H,1),ysimP1(ko:ko+H-1,4),'r-', ...
        time(1:H,1),ysimP1(ko:ko+H-1,5),'g-', ...
        time(1:H,1),ysimP1(ko:ko+H-1,6),'b-', ...
        time(1:H,1),ysimP1(ko:ko+H-1,7),'w-', ...
        time(1:H,1),ysimP1(ko:ko+H-1,8),'r-', ...
        time(1:H,1),ysimP1(ko:ko+H-1,9),'g-')
        ylabel('Pressure Deviation  [psi]')
        hy=get(gca, 'ylabel');
        set(hy, 'FontName', 'Palatino')
        xlabel('Time  [sec]')
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
        if simstep > 0
            axis([0 (H+1)*T -0.800 0.100])      % for F2 increases
            set(gca, 'FontName', 'Palatino')
            set(gca,'YTick', ...
                [-0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0.0 0.1])
```

```
        set(gca,'YTickLabels', ...
                [-0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0.0 0.1])
        md=(min(ysimP1(:,d1(9))))/2;
        for j=1:1:9
                text(3200+(j*550), md, ...
                        [HOUSE(d1(j),1), HOUSE(d1(j),2)], ...
                        'VerticalAlignment','bottom', ...
                        'HorizontalAlignment','right')
                hold on
                plot([3200+(j*550) 3700+(j*550)], ...
                        [md ysimP1(ko+(3700+j*550)/T, d1(j))])
                text(200, 0.05, ...
                        ['Flow Step = ', num2str(simstep),' g/s'], ...
                        'VerticalAlignment','middle', ...
                        'HorizontalAlignment','left')
        end
        fig=input('Enter P1 model output figure number: ','s')
        h=text(4500, -0.995, ...
                ['Figure ', fig, '.  P1 Simulation for Step in ', ...
                HOUSE(hs,1), HOUSE(hs,2), ...
                'F2 from Data Set ', num2str(data)]);
        set(h, 'HorizontalAlignment', 'center')
h=text(9270, 0.241, pno);
set(h, 'HorizontalAlignment', 'right')
else
        axis([0 (H+1)*T -0.100 0.800])        % for F2 decreases
        set(gca, 'FontName', 'Palatino')
        set(gca,'YTick', ...
                [-0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8])
        set(gca,'YTickLabels', ...
                [-0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8])
        md=(max(ysimP1(:,d1(1))))/2;
        for j=1:1:9
                text(3200+(j*550), md, ...
                        [HOUSE(d1(10-j),1), HOUSE(d1(10-j),2)], ...
                        'VerticalAlignment','top', ...
                        'HorizontalAlignment','right')
                hold on
                plot([3200+(j*550) 3700+(j*550)], ...
                        [md ysimP1(ko+(3700+j*550)/T, d1(10-j))])
                text(200, -0.05, ...
                        ['Flow Step = ', num2str(simstep),' g/s'], ...
                        'VerticalAlignment','middle', ...
                        'HorizontalAlignment','left')
        end
        fig=input('Enter P1 model output figure number: ','s')
```

```
            h=text(4500, -0.395, ...
                  ['Figure ', fig, '.  P1 Simulation for Step in ', ...
                  HOUSE(hs,1), HOUSE(hs,2), ...
                  ' F2 from Data Set ', num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      h=text(9270, 0.941, pno);
      set(h, 'HorizontalAlignment', 'right')
      end
      hold off
      grid
%
% Plot all the P2 simulations for the step input calculated from
loader data.
% Sort house names on basis of last value in plot.
[s2,d2]=sort(ysimP2(L-kx-ko,1:9));        % d2 has sorted house
                  % index based on time of last data; 1=smallest
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
plot(time(1:H,1),ysimP2(ko:ko+H-1,1),'y-', ...
      time(1:H,1),ysimP2(ko:ko+H-1,2),'m-', ...
      time(1:H,1),ysimP2(ko:ko+H-1,3),'c-', ...
      time(1:H,1),ysimP2(ko:ko+H-1,4),'r-', ...
      time(1:H,1),ysimP2(ko:ko+H-1,5),'g-', ...
      time(1:H,1),ysimP2(ko:ko+H-1,6),'b-', ...
      time(1:H,1),ysimP2(ko:ko+H-1,7),'w-', ...
      time(1:H,1),ysimP2(ko:ko+H-1,8),'r-', ...
      time(1:H,1),ysimP2(ko:ko+H-1,9),'g-')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
      if simstep > 0
            axis([0 (H+1)*T -2 7])          % for F2 increases
            set(gca, 'FontName', 'Palatino')
            set(gca,'YTick',[-2 -1 0 1 2 3 4 5 6 7])
            set(gca,'YTickLabels',[-2 -1 0 1 2 3 4 5 6 7])
            md=(max(ysimP2(:,d2(9)))+ysimP2(1600,d2(9)))/2;
            for j=1:1:9
                  text(3200+(j*550), md, ...
                        [HOUSE(d2(10-j),1), HOUSE(d2(10-j),2)], ...
                        'VerticalAlignment','bottom', ...
                        'HorizontalAlignment','right')
                  hold on
                  plot([3200+(j*550) 3700+(j*550)], ...
```

```
                    [md ysimP2(ko+(4000+j*600)/T, d2(10-j))])
                text(200, -0.50, ...
                    ['Flow Step = ', num2str(simstep),' g/s'], ...
                    'VerticalAlignment','middle', ...
                    'HorizontalAlignment','left')
            end
            fig=input('Enter P2 model output figure number: ','s')
            h=text(4500, -3.95, ['Figure ', fig, ...
                '. P2 Simulation for Step in ', ...
                HOUSE(hs,1), HOUSE(hs,2), ...
                ' F2 from Data Set ', num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
        else
            axis([0 (H+1)*T -7 2])          % for F2 decreases
            set(gca, 'FontName', 'Palatino')
            set(gca,'YTick',[-7 -6 -5 -4 -3 -2 -1 0 1 2])
            set(gca,'YTickLabels',[-7 -6 -5 -4 -3 -2 -1 0 1 2])
            md=(min(ysimP2(:,d2(1)))+ysimP2(1600,d2(1)))/2;
            for j=1:1:9
                text(3200+(j*550), md , ...
                    [HOUSE(d2(j),1), HOUSE(d2(j),2)], ...
                    'VerticalAlignment','top', ...
                    'HorizontalAlignment','right')
                hold on
                plot([3200+(j*550) 3700+(j*550)], ...
                    [md ysimP2(ko+(3700+j*550)/T, d2(j))])
                text(200, 0.50, ...
                    ['Flow Step = ', num2str(simstep),' g/s'], ...
                    'VerticalAlignment','middle', ...
                    'HorizontalAlignment','left')
            end
            fig=input('Enter P2 model output figure number: ','s')
            h=text(4500, -9.95, ['Figure ', fig, ...
                '. P2 Simulation for Step in ', ...
                HOUSE(hs,1), HOUSE(hs,2), ...
                ' F2 from Data Set ', num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
        end
        hold off
        grid
print -dps2
```

A.4 Matlab Program SimFC.m

```
% An program to simulate system response using models
% developed for inventory flow changes. A data set is used for
% comparision, with an initial flow step size being the only item
% taken from that data set. Model coefficients in matrix C are
% assumed to be loaded into the session, and the P2 data come
% from matrix D.
%
data=28;                    % data set number
hs=10;                      % loader house no (always 10 for IN)
EVi=59.335;                 % initial valve position in percent
EVf=56.997;                 % final valve position in percent
Patm=14.438;                % atmospheric pressure in psia
P2o=285.25;                 % initial P2 pressure at valve (B0 P2)
ko=68;                      % point prior to step input
L=2400;                     % total length of data record created
kx=L-size(D,1);             % length of simulation beyond data
kn=size(D,1);               % length of data original data record
H=1800;                     % number of points in final plot
T=5;                        % sample time in seconds
%
% Find and save P1 pressure at hs, time averaged from data start
% to ko.
P1o=mean(D(1:ko,2));        % average up to step at B0
%
% Plotting, identification, and result information.
IND=zeros(14,1);            % storage of input parameters and sse
time=0:5:(L-1)*5;           % time in seconds
time=time';                 % time as a column vector
SSE=zeros(9,2);             % storage of sum of squared errors
HOUSE=['A0';'B0';'BA';'C0';'D0';'E0';'EA';'F0';'SY';'IN'];
%
% For reference, plot flows, manipulating the data similarly to
% that used for the modeling. Indices must be changed each data
% set. This data is not filtered.
    flo=zeros(L,4);         % initialze to zero
    flo(1:L-kx,1:3)=[D(:,28) ...
        D(:,29) (D(:,19)+D(:,20)+D(:,21)+D(:,22)+D(:,23) ...
        +D(:,24)+D(:,25)+D(:,26)+D(:,27))]; % raw data
    floo(1,1:3)=mean(flo(1:ko,1:3));   % average up to step
    flo(L-kx+1:L,1)=ones(kx,1)*mean(flo(L-kx-20:L-kx,1));
                            % build an extention
    flo(L-kx+1:L,2)=ones(kx,1)*mean(flo(L-kx-20:L-kx,2));
    flo(L-kx+1:L,3)=ones(kx,1)*mean(flo(L-kx-20:L-kx,3));
    flo(:,1)=flo(:,1)-floo(1,1);     % subtract initial value from FC
```

```
      flo(:,2)=flo(:,2)-floo(1,2);          % and from FD
      flo(:,3)=flo(:,3)-floo(1,3);          % sum of F2
      flo(:,4)=flo(:,3)-flo(:,2)-flo(:,1);  % net flow INTO discharge
%
% Plot the flow data.
pno=input('Enter first flow page number: ','s')
figure(1)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
plot(time(1:H), flo(1:H,1), 'y')
      ylabel('Flow Deviation  [g/s]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
            axis([0 (H+1)*T -60 60])
            set(gca, 'FontName', 'Palatino')
            fig=input('Enter seperate FC flow figure number: ','s')
            h=text(4500, -85.8, ['Figure ', fig, '.  ', ...
                  ' FC Inventory Flow from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      h=text(9270, 78.6, pno);
      set(h, 'HorizontalAlignment', 'right')
      grid
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
plot(time(1:H), flo(1:H,2), 'b')
      ylabel('Flow Deviation  [g/s]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
            axis([0 (H+1)*T -60 60])
            set(gca, 'FontName', 'Palatino')
            fig=input('Enter seperate FD flow figure number: ','s')
            h=text(4500, -85.8, ['Figure ', fig, '.  ', ...
                  ' FD Inventory Flow from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      grid
print -dps2
```

```
pause
pno=input('Enter second flow page number: ','s')
figure(2)
set(gcf,'DefaultTextFontName','Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
      plot(time(1:H), flo(1:H,3), 'r')
      ylabel('Flow Deviation  [g/s]')
      hy=get(gca,'ylabel');
      set(hy,'FontName','Palatino')
      xlabel('Time  [sec]')
      hx=get(gca,'xlabel');
      set(hx,'FontName','Palatino')
            axis([0 (H+1)*T -60 60])
            set(gca,'FontName','Palatino')
            fig=input('Enter seperate flow figure number: ','s')
            h=text(4500, -85.8, ['Figure ', fig, '. ', ...
                  ' Sum of F2 Flows from Data Set ', ...
                  num2str(data)]);
            set(h,'HorizontalAlignment','center')
      h=text(9270, 78.6, pno);
      set(h,'HorizontalAlignment','right')
      grid
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
      plot(time(1:H), flo(1:H,4), 'm')
      ylabel('Flow Deviation  [g/s]')
      hy=get(gca,'ylabel');
      set(hy,'FontName','Palatino')
      xlabel('Time  [sec]')
      hx=get(gca,'xlabel');
      set(hx,'FontName','Palatino')
            axis([0 (H+1)*T -60 60])
            set(gca,'FontName','Palatino')
            fig=input('Enter seperate flow figure number: ','s')
            h=text(4500, -85.8, ['Figure ', fig, '. ', ...
      ' Net Flow Change into Discharge Header from Data Set ', ...
                  num2str(data)]);
            set(h,'HorizontalAlignment','center')
      grid
print -dps2
pause
%
% Calculate step size directly from data. Average values up to step
% for initial flow, and skip five sample times for step to settle
```

```
% prior to evaluating final flow as an average of 25 values.
      simstep=mean(D(ko+5:ko+30,28))-mean(D(1:ko,28))
% Set up the input. Flow step size taken directly from the data.
      u=zeros(L,1);
      u(ko+1:L,1)=ones(L-ko,1)*simstep;      % build the step
                                             % starting at point ko+1
%
% Create theta from previously determined coefficients residing
in matrix C.
ysimP2=zeros(L,1);
%
for i=1:9
      P2a1=C(37,i);
      P2b1=C(38,i);
      A2=[1 P2a1];
      B2=[0 P2b1];
      thP2=poly2th(A2,B2);
% Detrend pressure data to the average of values prior to step.
      yP1o=mean(D(1:ko,i));                  % average up to step for P1
      yP1(:,i)=D(:,i)-yP1o;                  % subtract initial value
                                             % from all output
      yP2o=mean(D(1:ko,i+9));                % average up to step for P2
      yP2(:,i)=D(:,i+9)-yP2o;                % subtract initial value
                                             % from all output
% Simulate the P2 response to a step.
      ysimP2(:,i)=idsim(u,thP2);             % simulate P2 using input
      seP2=(ysimP2(1:L-kx,i)-yP2(1:L-kx,i)).^2;      % sum of
                                   % squared errors between model
      SSE(i,2)=sum(seP2);          % and unfiltered output, P2
                                   % note that SSE(i,1) is empty
% Plot P1 and P2 data, and P2 simulation at individual locations
% for the flow step.
% Plot P1 data just for reference.
pno=input( ...
      ['Enter ', HOUSE(i,1), HOUSE(i,2), ' page number: '],'s')
figure(3)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
      plot(time(1:L-kx,1), yP1(1:L-kx,i), 'g:')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
```

```
        set(hx, 'FontName', 'Palatino')
              axis([0 (H+1)*T -0.1501 0.1501])
              set(gca, 'FontName', 'Palatino')
              text(ko*5+1000, -0.125, [num2str(simstep), ...
                    ' g/s Step'], 'VerticalAlignment', 'top', ...
                    'HorizontalAlignment','left')
              text(ko*5+1000, -0.125, ['Data Set ', num2str(data)], ...
                    'VerticalAlignment','bottom', ...
                    'HorizontalAlignment','left')
              fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                    ' P1 figure number: '],'s')
              h=text(4500, -0.215, ['Figure ', fig, '. ', ...
                     HOUSE(i,1), HOUSE(i,2), ...
                    ' P1 Data for Step in FC']);
              set(h, 'HorizontalAlignment', 'center')
        h=text(9270, 0.197, pno);
        set(h, 'HorizontalAlignment', 'right')
        grid
% Plot P2 data and result at individual locations for the flow step.
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
        plot(time(1:H,1), ysimP2(1:H,i), 'y-', ...
              time(1:L-kx,1), D(1:L-kx,9+i)-yP2o, 'g:')
        ylabel('Pressure Deviation  [psi]')
        hy=get(gca, 'ylabel');
        set(hy, 'FontName', 'Palatino')
        xlabel('Time  [sec]')
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
        if simstep < 0
              axis([0 (H+1)*T -4 28])              % for FC decreases
              set(gca, 'FontName', 'Palatino')
              set(gca,'YTick',[-4 0 4 8 12 16 20 24 28])
              set(gca,'YTickLabels',[-4 0 4 8 12 16 20 24 28])
              text(ko*5+500, 2.0, [num2str(simstep),' g/s Step'], ...
                    'VerticalAlignment','top', ...
                    'HorizontalAlignment','left')
              text(ko*5+500, 2.0, ['Data Set ', num2str(data)], ...
                    'VerticalAlignment','bottom', ...
                    'HorizontalAlignment','left')
              fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                    ' P2 figure number: '],'s')
              h=text(4500, -11.0, ['Figure ', fig, '. ', ...
                    HOUSE(i,1), HOUSE(i,2), ...
                    ' P2 Data and Simulation for Step in FC']);
              set(h, 'HorizontalAlignment', 'center')
```

```
    else
        axis([0 (H+1)*T -28 4])                % for FC increases
        set(gca,'FontName','Palatino')
        set(gca,'YTick',[-28 -24 -20 -16 -12 -8 -4 0 4])
        set(gca,'YTickLabels',[-28 -24 -20 -16 -12 -8 -4 0 4])
        text(ko*5+1000, -2.0, [num2str(simstep),' g/s Step'], ...
            'VerticalAlignment','top', ...
            'HorizontalAlignment','left')
        text(ko*5+1000, -2.0, ['Data Set ', num2str(data)], ...
            'VerticalAlignment','bottom', ...
            'HorizontalAlignment','left')
        fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
            ' P2 figure number: '],'s')
        h=text(4500, -35.0, ['Figure ', fig, '. ', ...
            HOUSE(i,1), HOUSE(i,2), ...
            ' P2 Data and Simulation for Step in FC']);
        set(h,'HorizontalAlignment','center')
    end
    grid
pause                                   % required to see result
print -dps2
end
%
% Plot all the P2 simulations for the step input calculated from
% data. Sort house names on basis of last value in plot.
[s2,d2]=sort(ysimP2(ko+H-1,1:9));
pno=input('Enter all P2 models page number: ','s')
figure(4)
set(gcf,'DefaultTextFontName','Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
%subplot(2,1,2)                         % alternative positioning
%set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
plot(time(1:H,1),ysimP2(ko:ko+H-1,1),'y-', ...
    time(1:H,1),ysimP2(ko:ko+H-1,2),'m-', ...
    time(1:H,1),ysimP2(ko:ko+H-1,3),'c-', ...
    time(1:H,1),ysimP2(ko:ko+H-1,4),'r-', ...
    time(1:H,1),ysimP2(ko:ko+H-1,5),'g-', ...
    time(1:H,1),ysimP2(ko:ko+H-1,6),'b-', ...
    time(1:H,1),ysimP2(ko:ko+H-1,7),'w-', ...
    time(1:H,1),ysimP2(ko:ko+H-1,8),'r-', ...
    time(1:H,1),ysimP2(ko:ko+H-1,9),'g-')
    ylabel('Pressure Deviation  [psi]')
    hy=get(gca,'ylabel');
    set(hy,'FontName','Palatino')
```

```
xlabel('Time  [sec]')
hx=get(gca, 'xlabel');
set(hx, 'FontName', 'Palatino')
if simstep < 0
      axis([0 (H+1)*T -4 28])                % for FC decreases
      set(gca, 'FontName', 'Palatino')
      set(gca,'YTick',[-4 0 4 8 12 16 20 24 28])
      set(gca,'YTickLabels',[-4 0 4 8 12 16 20 24 28])
      md=(max(ysimP2(:,d2(9))))/2;
      for j=1:1:9
            text(3200+(j*550), md , ...
                  [HOUSE(d2(j),1), HOUSE(d2(j),2)], ...
                  'VerticalAlignment','top', ...
                  'HorizontalAlignment','right')
            hold on
            plot([3200+(j*550) 3700+(j*550)], ...
                  [md ysimP2(ko+(3700+j*550)/T, d2(j))])
            text(200, -2, ...
                  ['Flow Step = ', num2str(simstep),' g/s'], ...
                  'VerticalAlignment','middle', ...
                  'HorizontalAlignment','left')
      end
      fig=input('Enter P2 model output figure number: ','s')
      h=text(4500, -11.0, ['Figure ', fig, ...
            '.  P2 Model for Step in FC from Data Set ', ...
            num2str(data)]);
      set(h, 'HorizontalAlignment', 'center')
h=text(9270, 33.0, pno);
set(h, 'HorizontalAlignment', 'right')
else
      axis([0 (H+1)*T -28 4])                % for FC increases
      set(gca, 'FontName', 'Palatino')
      set(gca,'YTick',[-28 -24 -20 -16 -12 -8 -4 0 4])
      set(gca,'YTickLabels',[-28 -24 -20 -16 -12 -8 -4 0 4])
      md=(min(ysimP2(:,d2(9))))/2;
      for j=1:1:9
            text(3200+(j*550), md, ...
                  [HOUSE(d2(10-j),1), HOUSE(d2(10-j),2)], ...
                  'VerticalAlignment','bottom', ....
                  'HorizontalAlignment','right')
            hold on
            plot([3200+(j*550) 3700+(j*550)], ...
                  [md ysimP2(ko+(3700+j*550)/T, d2(10-j))])
            text(200, 2, ...
                  ['Flow Step = ', num2str(simstep),' g/s'], ...
                  'VerticalAlignment','middle', ...
```

```
                    'HorizontalAlignment','left')
            end
            fig=input('Enter P2 model output figure number: ','s')
            h=text(4500, -35.0, ['Figure ', fig, ...
                  '.  P2 Model for Step in FC from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      h=text(9270, 9.0, pno);
      set(h, 'HorizontalAlignment', 'right')
      end
      hold off
      grid
print -dps2
```

A.5 Matlab Program Micro1mod.m

```
% A program to model P1 responding to compressor flows.
% For P1, noe1 of both [[2][2][1]] and [[2 2][2 2][1 1]] are used to see
% differences in 1 input and 2 input results. Initialization using
% oe [[2] [2] [1]] is used. Use the perturbed house flow as one input
% and the sum of all other flows as a second input. This gives all
% flows, except the perturbed one, equal weighting. Modeling and
% simulation are done with data manipulated to give zero value
% at first time stamp. To achive this, data up to first time stamp
% is taken as the average of values up to that point, and this
% average is then subtracted from all data. The last values of the
% data are averaged and used to extend the data tail.
%
data=5;                     % data set number
hs=2;                       % loader house no (A0=1,...F0=8,SY=9)
LSi=100;                    % initial LS loader position in percent
LSf=92.30;                  % final LS loader position in percent
Patm=14.537;                % atmospheric pressure in psia
P1o=1.467;                  % initial P1 pressure at stepped loader
ko=240;                     % point prior to step input
ki1=800;                    % initial sample to build P1 extention
kf1=960;                    % final sample to build P1 extension
L=2400;                     % total length of created data record
kx=L-size(D,1);             % length of an extension of data
H=1800;                     % number of points in final plot
kn=size(D,1);               % length of data original data record
T=5;                        % sample time in seconds
%
IND=zeros(15,1);            % storage of input parameters
TH1=zeros(9,54);
time=0:5:(L-1)*5;           % in seconds
time=time';
HOUSE=['A0';'B0';'BA';'C0';'D0';'E0';'EA';'F0';'SY'];
INPUT2=['1 PRIMARY';'SECONDARY'];
%
% Manipulate the input, for use with P1, to smooth initial
% conditions and add tail.
      zoe1=zeros(L,4);                          % initialze to zero
      zoe1(1:L-kx,2:3)=[D(:,20) (D(:,19)+D(:,21)+D(:,22)+D(:,23) ...
            +D(:,24)+D(:,25)+D(:,26)+D(:,27)-D(:,28)-D(:,29))];
            % raw data
      zooe1(1,2:3)=mean(zoe1(1:ko,2:3));        % average up to step
      zoe1(1:ko,2)=ones(ko,1)*zooe1(1,2);       % set all values up to
                                                % the step to equal xo
      zoe1(1:ko,3)=ones(ko,1)*zooe1(1,3);
```

```
        zoe1(L-kx+1:L,2)=ones(kx,1)*mean(zoe1(ki1:kf1,2));
                                                % build an extention
        zoe1(L-kx+1:L,3)=ones(kx,1)*mean(zoe1(ki1:kf1,3));
        zoe1(:,2)=zoe1(:,2)-zooe1(1,2);     % subtract initial
                                            % input value from all
        zoe1(:,3)=zoe1(:,3)-zooe1(1,3);
        zoe1(:,4)=zoe1(:,2)+zoe1(:,3);      % add the two inputs for
                                            % plotting
%
% Plot the flow input data.
pno=input('Enter flow page number: ','s')
figure(1)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
plot(time(1:H), zoe1(1:H,2), 'y', time(1:H), zoe1(1:H,3), 'm')
        ylabel('Flow Deviation  [g/s]')
        hy=get(gca, 'ylabel');
        set(hy, 'FontName', 'Palatino')
        xlabel('Time  [sec]')
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
                axis([0 (H+1)*T -20 20])
                set(gca, 'FontName', 'Palatino')
                fig=input('Enter separate flow figure number: ','s')
                h=text(4500, -28.6, ['Figure ', fig, ...
        '. Primary Flow and Secondary Flow from Data Set ', ...
                        num2str(data)]);
                set(h, 'HorizontalAlignment', 'center')
                h=text(6000, zoe1(6000/5,2)-2, ['Primary Flow']);
                set(h, 'HorizontalAlignment', 'left')
                h=text(6000, zoe1(6000/5,3)-2, ['Secondary Flow']);
                set(h, 'HorizontalAlignment', 'left')
        grid
        h=text(9270, 26.2, pno);
        set(h, 'HorizontalAlignment', 'right')
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
plot(time(1:H), zoe1(1:H,4), 'c')
        ylabel('Flow Deviation  [g/s]')
        hy=get(gca, 'ylabel');
        set(hy, 'FontName', 'Palatino')
        xlabel('Time  [sec]')
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
```

```
            axis([0 (H+1)*T -20 20])
            set(gca, 'FontName', 'Palatino')
            fig=input('Enter combined flow figure number: ','s')
            h=text(4500, -28.6, ['Figure ', fig, ...
'. Sum of Primary Flow and Secondary Flow from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
            h=text(6000, zoe1(6000/5,4)-2, ['Sum of Flows']);
            set(h, 'HorizontalAlignment', 'left')
      grid
print -dps2
pause
%
% Flow step size calculation for loader changes.
Pc=15.9;                  % base pressure for loader/flow calibration
C(1:6)=[0.0015928689 -0.40467791 34.59480361 -958.9842469
5.371698113 -482.3767296];
% for initial loader position
      if LSi < 98
            F2i=C(1)*LSi^3+C(2)*LSi^2+C(3)*LSi+C(4);
            F2i=(F2i*(P1o+Patm)/Pc);
      else
            F2i=C(5)*LSi+C(6);
            F2i=(F2i*(P1o+Patm)/Pc);
      end
% for final loader position
      if LSf < 98
            F2f=C(1)*LSf^3+C(2)*LSf^2+C(3)*LSf+C(4);
            F2f=(F2f*(P1o+Patm)/Pc);
      else
            F2f=C(5)*LSf+C(6);
            F2f=(F2f*(P1o+Patm)/Pc);
      end
step=F2f-F2i                  % initial flow step size
%
% Calculate initialization parameter values using a single input
% oe model.
%
i=1;                          % index for the P1 output location
                              % used for initiallization
      noe11=[[2] [2] [1]];    % one second order output, one input
                              % with one delay
% Manipulate the outputs similarly to the input.
      zoe1(1:L-kx,1)=D(:,i);                  % raw P1 data
      zooe1(1,1)=mean(zoe1(1:ko,1));          % average up to step
      zoe1(1:ko,1)=ones(ko,1)*zooe1(1,1);     % set all values up to
```

```
                                        % step to equal yo
    zoe1(L-kx+1:L,1)=ones(kx,1)*mean(zoe1(ki1:kf1,1));
                                        % build the extension
    zoe1(:,1)=zoe1(:,1)-zooe1(1,1);     % subtract initial
                                        % output value from all
% Compute the combined single input model for parameter
% initialization purposes.
    thoe1ci=oe([zoe1(:,1) zoe1(:,4)], noe11);% c=combined,
                                        % i=initial
    thoe1t=oe([zoe1(:,1) zoe1(:,4)], noe11); % t=temporary
% Use combined inputs parameter results as initialization for
% separate inputs model.
    noe12=[[2 2] [2 2] [1 1]];          % one second order output,
                                        % two inputs w/ one delay
% Calculate a two input model for purpose of getting two input
% theta template.
    thoe1si=oe([zoe1(:,1) zoe1(:,2) zoe1(:,3)], noe12);
                                        % s=single, i=initial
% Replace sometimes unstable parameters with the single input
% initialization values.
    thoe1si(3,1)=thoe1ci(3,1);          % b1p
    thoe1si(3,2)=thoe1ci(3,2);          % b2p
    thoe1si(3,3)=thoe1ci(3,1);          % b1o
    thoe1si(3,4)=thoe1ci(3,2);          % b2o
    thoe1si(3,5)=thoe1ci(3,3);          % f1p
    thoe1si(3,6)=thoe1ci(3,4);          % f2p
    thoe1si(3,7)=thoe1ci(3,3);          % f1o
    thoe1si(3,8)=thoe1ci(3,4);          % f2o
% Calculate a better initial two input model, using the above
% initialization.
    thoe1si=oe([zoe1(:,1) zoe1(:,2) zoe1(:,3)], thoe1si); % use later
%
% Calculate parameter values for all houses using one and two
% input initializations.
%
    noe11=[[2] [2] [1]];            % one second order output,
                                    % one input with one delay
    yoe111=zeros(L,9);              % matrix holds individual input
                                    % 1 simulation, single input
    yoe211=zeros(L,9);              % matrix holds individual input
                                    % 2 simulation, single input
    yoeS11=zeros(L,9);              % matrix holds sum of inputs 1
                                    % and 2 simulation, single
    noe12=[[2 2] [2 2] [1 1]];      % one second order output, one
                                    % input with one delay
    yoe112=zeros(L,9);              % matrix holds individual input
```

```
                                    %1 simulation, two inputs
    yoe212=zeros(L,9);              % matrix holds individual input
                                    % 2 simulation, two inputs
    yoeS12=zeros(L,9);              % matrix holds sum of inputs 1
                                    % and 2 simulation, single
%
for i=1:9                           % index for P1 output location
                                    % used for initiallization
% Manipulate the outputs similarly to the input.
    zoe1(1:L-kx,1)=D(:,i);                          % raw P1 data
    zooe1(1,1)=mean(zoe1(1:ko,1));          % average up to step
    zoe1(1:ko,1)=ones(ko,1)*zooe1(1,1);     % set all values up to
                                            % step to equal yo
    zoe1(L-kx+1:L,1)=ones(kx,1)*mean(zoe1(ki1:kf1,1));
                                            % build the extension
    zoe1(:,1)=zoe1(:,1)-zooe1(1,1);     % subtract initial
                                        % output value from all
%
% Calculate the combined flow input model using initialization
% from above, and simulate.
    thoe1c=oe([zoe1(:,1) zoe1(:,4)], thoe1ci);
    yoe1(:,i)=idsim([zoe1(:,4)],thoe1c);        % combined input
                                        % model output result
    seoe11=(yoe1(1:L-kx,i)-zoe1(1:L-kx,1)).^2;      % sum of
                                        % squared errors between
    sseoe11=sum(seoe11);                % model and data
% Look at the zero-pole structure.
[zepo11, K11]=th2zp(thoe1c);            % zero, poles, and gain
% Save 19 items that characterize the results of the combined flow
% model in TH1.
% First save the parameter values and corresponding gain, zero,
% and poles.
TH1(i,1)=thoe1c(3,1);                   % b1
TH1(i,2)=thoe1c(3,2);                   % b2
TH1(i,3)=thoe1c(3,3);                   % f1
TH1(i,4)=thoe1c(3,4);                   % f2
TH1(i,5)=K11(2,1);    % K (static gain, NOT transfer function gain)
TH1(i,6)=zepo11(2,1);                   % z
TH1(i,7)=zepo11(2,3);                   % pa
TH1(i,8)=zepo11(3,3);                   % pb
% Second, save the standard deviation of above values.
TH1(i,9)=thoe1c(4,1);                   % b1
TH1(i,10)=thoe1c(5,2);                  % b2
TH1(i,11)=thoe1c(6,3);                  % f1
TH1(i,12)=thoe1c(7,4);                  % f2
TH1(i,13)=K11(3,1);                     % K
```

```
TH1(i,14)=zepo11(2,2);                    % z
TH1(i,15)=zepo11(2,4);                    % pa
TH1(i,16)=zepo11(3,4);                    % pb
% Third, save miscellaneous values.
TH1(i,17)=thoe1c(1,1);              % variance of the noise
TH1(i,18)=thoe1c(2,1);              % Akaike's final prediction error
TH1(i,19)=sseoe11;     % sum of the squared errors, unfiltered
%
% Calculate the portions of the P1 output made up by the
% individual inputs using the parameters from combined input
% oe model.
%     Simulate model output for single primary flow input.
      zoe11=zeros(L,2);
      zoe11(:,1:2)=[zoe1(:,1) zoe1(:,2)];
      yoe111(:,i)=idsim(zoe11(:,2), thoe1c);   % portion of output
                                          % due to primary flow
%     Simulate model output for single secondary flow input.
      zoe21=zeros(L,2);
      zoe21(:,1:2)=[zoe1(:,1) zoe1(:,3)];
      yoe211(:,i)=idsim(zoe21(:,2), thoe1c);   % portion of output
                                          % due to secondary flow
%     Sum all the components to see it represents whole response.
      yoeS11(:,i)=yoe111(:,i)+yoe211(:,i);       % sum of two
                                          % separate output parts
%
% Calculate a separate flow input model, using the above
% initialization parameters.
      thoe1s=oe([zoe1(:,1) zoe1(:,2) zoe1(:,3)], thoe1si);
      yoe1(:,i)=idsim([zoe1(:,2) zoe1(:,3)],thoe1s);  % two input
                                          % model output result
      seoe12=(yoe1(1:L-kx,i)-zoe1(1:L-kx,1)).^2;     % sum of
                                          % squared errors between
      sseoe12=sum(seoe12);                % model and data
% Look at the zero-pole structure.
[zepo12, K12]=th2zp(thoe1s);              % zero, poles, and gain
% Save 35 items that characterize the results of the separate flow
% model in TH1.
% Save the parameter values and corresponding gain, zero, and
poles for primary input.
TH1(i,20)=thoe1s(3,1);                    % b11
TH1(i,21)=thoe1s(3,2);                    % b21
TH1(i,22)=thoe1s(3,5);                    % f11
TH1(i,23)=thoe1s(3,6);                    % f21
TH1(i,24)=K12(2,1);  % K1 (static gain, NOT transfer function gain)
TH1(i,25)=zepo12(2,1);                    % z1
TH1(i,26)=zepo12(2,3);                    % pa1
```

```
TH1(i,27)=zepo12(3,3);                    % pb1
% Second, save the standard deviation of above values.
TH1(i,28)=thoe1s(4,1);                    % b11
TH1(i,29)=thoe1s(5,2);                    % b21
TH1(i,30)=thoe1s(8,5);                    % f11
TH1(i,31)=thoe1s(9,6);                    % f21
TH1(i,32)=K12(3,1);                       % K1
TH1(i,33)=zepo12(2,2);                    % z1
TH1(i,34)=zepo12(2,4);                    % pa1
TH1(i,35)=zepo12(3,4);                    % pb1
% Save the parameter values and corresponding gain, zero, and
% poles for secondary input.
TH1(i,36)=thoe1s(3,3);                    % b12
TH1(i,37)=thoe1s(3,4);                    % b22
TH1(i,38)=thoe1s(3,7);                    % f12
TH1(i,39)=thoe1s(3,8);                    % f22
TH1(i,40)=K12(2,2);  % K2 (static gain, NOT transfer function gain)
TH1(i,41)=zepo12(2,5);                    % z2
TH1(i,42)=zepo12(2,7);                    % pa2
TH1(i,43)=zepo12(3,7);                    % pb2
% Second, save the standard deviation of above values.
TH1(i,44)=thoe1s(6,3);                    % b12
TH1(i,45)=thoe1s(7,4);                    % b22
TH1(i,46)=thoe1s(10,7);                   % f12
TH1(i,47)=thoe1s(11,8);                   % f22
TH1(i,48)=K12(3,2);                       % K2
TH1(i,49)=zepo12(2,6);                    % z2
TH1(i,50)=zepo12(2,8);                    % pa2
TH1(i,51)=zepo12(3,8);                    % pb2
% Third, save miscellaneous values.
TH1(i,52)=thoe1c(1,1);                    % variance of the noise
TH1(i,53)=thoe1c(2,1);              % Akaike's final prediction error
TH1(i,54)=sseoe12;     % sum of the squared errors, unfiltered
%
% Calculate and plot the portions of the P1 output made up by
% individual inputs using the oe parameters from the separate
% input model.
%     Simulate model output for single primary flow input.
      thoe1t(3,1)=thoe1s(3,1);            % b1p
      thoe1t(3,2)=thoe1s(3,2);            % b2p
      thoe1t(3,3)=thoe1s(3,5);            % f1p
      thoe1t(3,4)=thoe1s(3,6);            % f2p
      zoe11=zeros(L,2);
      zoe11(:,1:2)=[zoe1(:,1) zoe1(:,2)];
      yoe112(:,i)=idsim(zoe11(:,2), thoe1t);   % portion of output
                                          % due to primary flow
```

```
%     Simulate model output for single secondary flow input.
      thoe1t(3,1)=thoe1s(3,3);              % b1o
      thoe1t(3,2)=thoe1s(3,4);              % b2o
      thoe1t(3,3)=thoe1s(3,7);              % f1o
      thoe1t(3,4)=thoe1s(3,8);              % f2o
      zoe21=zeros(L,2);
      zoe21(:,1:2)=[zoe1(:,1) zoe1(:,3)];
      yoe212(:,i)=idsim(zoe1(:,3), thoe1t);     % portion of output
                                            % due to secondary flow
%     Sum all the components.
      yoeS12(:,i)=yoe112(:,i)+yoe212(:,i);      % sum of two
                                            % separate output parts
%
% Plot both combined input and two input model outputs, along
% with data.
pno=input(['Enter  ', HOUSE(i,1), HOUSE(i,2), '  both model page
number: '],'s')
figure(2)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
% Plot for the combined flow input model
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
      plot(time(1:H,1), yoeS11(1:H,i), 'y-', ...
            time(1:ko,1), D(1:ko,i)-zooe1(1,1), 'g:', ...
            time(ko+1:L-kx,1), zoe1(ko+1:L-kx,1), 'g:')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
      if step > 0
            axis([0 (H+1)*T -0.3001 0.0501])    % for F2 increases
            set(gca, 'FontName', 'Palatino')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                  ' P1 combined figure number: '],'s')
            h=text(4500, -0.376, ['Figure ', fig, '.  ', ...
                  HOUSE(i,1), HOUSE(i,2), ...
      ' P1 Data and P1 Combined Flow Model from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      h=text(9270, 0.105, pno);
      set(h, 'HorizontalAlignment', 'right')
      else
            axis([0 (H+1)*T -0.0501 0.3001])    % for F2 decreases
```

```
            set(gca, 'FontName', 'Palatino')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                  ' P1 combined figure number: '],'s')
            h=text(4500, -0.126, ['Figure ', fig, '. ', ...
                  HOUSE(i,1), HOUSE(i,2), ...
      ' P1 Data and P1 Combined Flow Model from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      h=text(9270, 0.355, pno);
      set(h, 'HorizontalAlignment', 'right')
      end
      grid
% Plot for the separate flow input model
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
      plot(time(1:H,1), yoeS12(1:H,i), 'm-', ...
            time(1:ko,1), D(1:ko,i)-zooe1(1,1), 'g:', ...
            time(ko+1:L-kx,1), zoe1(ko+1:L-kx,1), 'g:')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
      if step > 0
            axis([0 (H+1)*T -0.3001 0.0501])    % for F2 increases
            set(gca, 'FontName', 'Palatino')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ' P1
separate figure number: '],'s')
            h=text(4500, -0.376, ['Figure ', fig, '. ', ...
                  HOUSE(i,1), HOUSE(i,2), ...
      ' P1 Data and P1 Separate Flow Model from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      h=text(9270, 0.105, pno);
      set(h, 'HorizontalAlignment', 'right')
      else
            axis([0 (H+1)*T -0.0501 0.3001])    % for F2 decreases
            set(gca, 'FontName', 'Palatino')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                  ' P1 separate figure number: '],'s')
            h=text(4500, -0.126, ['Figure ', fig, '. ', ...
                  HOUSE(i,1), HOUSE(i,2), ...
      ' P1 Data and P1 Separate Flow Model from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
```

```
        end
        grid
pause
print -dps2
%
% Plot both combined input and two input model output
% components and sum
pno=input(['Enter  ', HOUSE(i,1), HOUSE(i,2), ' components page
number: '],'s')
figure(3)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
% Plot for the combined flow input model
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
        plot(time(1:H,1), yoe111(ko:ko+H-1,i), 'y-', ...
                time(1:H,1), yoe211(ko:ko+H-1,i), 'm', ...
                time(1:H,1), yoeS11(ko:ko+H-1,i), 'r')
        ylabel('Pressure Deviation  [psi]')
        hy=get(gca, 'ylabel');
        set(hy, 'FontName', 'Palatino')
        xlabel('Time  [sec]')
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
                axis([0 (H+1)*T -1.5 1.5])      % for both F2 flow
                                                % increases and decreases
                set(gca, 'FontName', 'Palatino')
                fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                        ' P1 figure number: '],'s')
                h=text(4500, -2.13, ['Figure ', fig, '.  ', ...
                        HOUSE(i,1), HOUSE(i,2), ...
        ' P1 Combined Flow Model Components from Data Set ', ...
                        num2str(data)]);
                set(h, 'HorizontalAlignment', 'center')
                h=text(4500, yoe111(5500/5,i)-0.1, ...
                        ['Primary Component']);
                set(h, 'HorizontalAlignment', 'left')
                h=text(4500, yoeS11(5500/5,i)-0.1, ...
                        ['Sum of Components']);
                set(h, 'HorizontalAlignment', 'left')
                h=text(4500, yoe211(5500/5,i)-0.1, ...
                        ['Secondary Component']);
                set(h, 'HorizontalAlignment', 'left')
        h=text(9270, 1.94, pno);
        set(h, 'HorizontalAlignment', 'right')
        grid
```

```
% Plot for the separate flow input model
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
      plot(time(1:H,1), yoe112(ko:ko+H-1,i), 'y-', ...
            time(1:H,1), yoe212(ko:ko+H-1,i), 'm', ...
            time(1:H,1), yoeS12(ko:ko+H-1,i), 'r')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
            axis([0 (H+1)*T -1.5 1.5])      % for both F2 flow
                                            % increases and decreases
            set(gca, 'FontName', 'Palatino')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                  ' P1 figure number: '],'s')
            h=text(4500, -2.13, ['Figure ', fig, '.  ', ...
                  HOUSE(i,1), HOUSE(i,2), ...
      ' P1 Separate Flow Model Components from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
            h=text(4500, yoe112(4500/5,i)-0.1, ...
                  ['Primary Component']);
            set(h, 'HorizontalAlignment', 'left')
            h=text(4500, yoeS12(4500/5,i)-0.1, ...
                  ['Sum of Components']);
            set(h, 'HorizontalAlignment', 'left')
            h=text(4500, yoe212(4500/5,i)-0.1, ...
                  ['Secondary Component']);
            set(h, 'HorizontalAlignment', 'left')
      grid
pause
print -dps
end
%
% Plot all the P1 simulations for the combined flow model type.
% First, plot the primary flow components for the combined flow
% model.
% Sort house names on basis of last value in plot.
[s1,d1]=sort(yoe111(L-kx-ko,1:9));          % d1 has sorted house
                        % index based on last data; 1=smallest
pno=input('Enter all models page number: ','s')
figure(4)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
```

```
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
plot(time(1:H,1),yoe111(ko:ko+H-1,1),'y-', ...
      time(1:H,1),yoe111(ko:ko+H-1,2),'m-', ...
      time(1:H,1),yoe111(ko:ko+H-1,3),'c-', ...
      time(1:H,1),yoe111(ko:ko+H-1,4),'r-', ...
      time(1:H,1),yoe111(ko:ko+H-1,5),'g-', ...
      time(1:H,1),yoe111(ko:ko+H-1,6),'b-', ...
      time(1:H,1),yoe111(ko:ko+H-1,7),'w-', ...
      time(1:H,1),yoe111(ko:ko+H-1,8),'r-', ...
      time(1:H,1),yoe111(ko:ko+H-1,9),'g-')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
      if step > 0
            axis([0 (H+1)*T -1.5001 0.501])     % for F2 increases
            set(gca, 'FontName', 'Palatino')
            md=(min(yoe111(:,d1(9))))/2;
            for j=1:1:9
                  text(3200+(j*550), md, ...
                        [HOUSE(d1(j),1), HOUSE(d1(j),2)], ...
                        'VerticalAlignment','bottom', ...
                        'HorizontalAlignment','right')
                  hold on
                  plot([3200+(j*550) 3700+(j*550)], ...
                        [md yoe111(ko+(3700+j*550)/T, d1(j))])
            end
            fig=input( ...
'Enter combined model P1 primary component figure number:', ...
                  's')
            h=text(4500, -1.92, ['Figure ', fig, ...
'. P1 Primary Components for Combined Model from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      h=text(9270, 0.82, pno);
      set(h, 'HorizontalAlignment', 'right')
      else
            axis([0 (H+1)*T -0.5001 1.5001])    % for F2 decreases
            set(gca, 'FontName', 'Palatino')
            md=(max(yoe111(:,d1(1))))/2;
            for j=1:1:9
                  text(3200+(j*550), md, ...
                        [HOUSE(d1(10-j),1), HOUSE(d1(10-j),2)], ...
```

```
                    'VerticalAlignment','top', ...
                    'HorizontalAlignment','right')
                hold on
                plot([3200+(j*550) 3700+(j*550)], ...
                    [md yoe111(ko+(3700+j*550)/T, d1(10-j))])
            end
            fig=input( ...
'Enter combined model P1 primary component figure number:', ...
                's')
            h=text(4500, -0.92, ['Figure ', fig, ...
'. P1 Primary Components for Combined Model from Data Set ', ...
                num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
        h=text(9270, 1.82, pno);
        set(h, 'HorizontalAlignment', 'right')
        end
        hold off
        grid
% Second, plot the secondary flow components for the combined
% flow model.
% Sort house names on basis of last value in plot.
[s2,d2]=sort(yoe211(L-kx-ko,1:9)); % d2 has sorted house index
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
plot(time(1:H,1),yoe211(ko:ko+H-1,1),'y-', ...
        time(1:H,1),yoe211(ko:ko+H-1,2),'m-', ...
        time(1:H,1),yoe211(ko:ko+H-1,3),'c-', ...
        time(1:H,1),yoe211(ko:ko+H-1,4),'r-', ...
        time(1:H,1),yoe211(ko:ko+H-1,5),'g-', ...
        time(1:H,1),yoe211(ko:ko+H-1,6),'b-', ...
        time(1:H,1),yoe211(ko:ko+H-1,7),'w-', ...
        time(1:H,1),yoe211(ko:ko+H-1,8),'r-', ...
        time(1:H,1),yoe211(ko:ko+H-1,9),'g-')
        ylabel('Pressure Deviation  [psi]')
        hy=get(gca, 'ylabel');
        set(hy, 'FontName', 'Palatino')
        xlabel('Time  [sec]')
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
        if step > 0
            axis([0 (H+1)*T -.5001 1.501])      % for F2 increases
            set(gca, 'FontName', 'Palatino')
            md=(max(yoe211(:,d1(9))))/2;
            for j=1:1:9
                text(3200+(j*550), md, ...
                    [HOUSE(d1(j),1), HOUSE(d1(j),2)], ...
```

```
                    'VerticalAlignment','top', ...
                    'HorizontalAlignment','right')
                hold on
                plot([3200+(j*550) 3700+(j*550)], ...
                    [md yoe211(ko+(3700+j*550)/T, d1(j))])
            end
            fig=input( ...
'Enter combined model P1 secondary component fig number:', ...
                's')
            h=text(4500, -0.92, ['Figure ', fig, ...
'. P1 Secondary Components for Combined Model from Data S', ...
                num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
        else
            axis([0 (H+1)*T -1.5001 0.5001])    % for F2 decreases
            set(gca, 'FontName', 'Palatino')
            md=(min(yoe211(:,d1(1))))/2;
            for j=1:1:9
                text(3200+(j*550), md, ...
                    [HOUSE(d1(10-j),1), HOUSE(d1(10-j),2)], ...
                    'VerticalAlignment','bottom', ...
                    'HorizontalAlignment','right')
                hold on
                plot([3200+(j*550) 3700+(j*550)], ...
                    [md yoe211(ko+(3700+j*550)/T, d1(10-j))])
            end
            fig=input('Enter combined model P1 secondary
component figure number: ','s')
            h=text(4500, -1.92, ['Figure ', fig, ...
'. P1 Secondary Components for Combined Model from Data S', ...
                num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
        end
        hold off
        grid
pause
print
%
% Plot all the P1 simulations for the separate flow model type.
% First, plot the primary flow components for the separate flow
% model.
% Sort house names on basis of last value in plot.
[s1,d1]=sort(yoe112(L-kx-ko,1:9)); % d1 has sorted house index
                                   %based on last data; 1=smallest
pno=input('Enter all models page number: ','s')
figure(5)
```

```
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
plot(time(1:H,1),yoe112(ko:ko+H-1,1),'y-', ...
      time(1:H,1),yoe112(ko:ko+H-1,2),'m-', ...
      time(1:H,1),yoe112(ko:ko+H-1,3),'c-', ...
      time(1:H,1),yoe112(ko:ko+H-1,4),'r-', ...
      time(1:H,1),yoe112(ko:ko+H-1,5),'g-', ...
      time(1:H,1),yoe112(ko:ko+H-1,6),'b-', ...
      time(1:H,1),yoe112(ko:ko+H-1,7),'w-', ...
      time(1:H,1),yoe112(ko:ko+H-1,8),'r-', ...
      time(1:H,1),yoe112(ko:ko+H-1,9),'g-')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
      if step > 0
            axis([0 (H+1)*T -1.5001 0.501])     % for F2 increases
            set(gca, 'FontName', 'Palatino')
            md=(min(yoe112(:,d1(9))))/2+0.1;
            for j=1:1:9
                  text(3200+(j*550), md, ...
                        [HOUSE(d1(j),1), HOUSE(d1(j),2)], ...
                        'VerticalAlignment','bottom', ...
                        'HorizontalAlignment','right')
                  hold on
                  plot([3200+(j*550) 3700+(j*550)], ...
                        [md yoe112(ko+(3700+j*550)/T, d1(j))])
            end
            fig=input( ...
'Enter separate model P1 primary component figure number: ','s')
            h=text(4500, -1.92, ['Figure ', fig, ...
'. P1 Primary Components for Separate Model from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      h=text(9270, 0.82, pno);
      set(h, 'HorizontalAlignment', 'right')
      else
            axis([0 (H+1)*T -0.5001 1.5001])    % for F2 decreases
            set(gca, 'FontName', 'Palatino')
            md=(max(yoe112(:,d1(1))))/2-0.1;
            for j=1:1:9
                  text(3200+(j*550), md, ...
```

```
                    [HOUSE(d1(10-j),1), HOUSE(d1(10-j),2)], ...
                    'VerticalAlignment','top', ...
                    'HorizontalAlignment','right')
                hold on
                plot([3200+(j*550) 3700+(j*550)], ...
                    [md yoe112(ko+(3700+j*550)/T, d1(10-j))])
            end
            fig=input( ...
'Enter separate model P1 primary component figure number: ','s')
            h=text(4500, -0.92, ['Figure ', fig, ...
'. P1 Primary Components for Separate Model from Data Set ', ...
                num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
        h=text(9270, 1.82, pno);
        set(h, 'HorizontalAlignment', 'right')
        end
        hold off
        grid
% Second, plot the secondary flow components for the separate
% flow model.
% Sort house names on basis of last value in plot.
[s2,d2]=sort(yoe212(L-kx-ko,1:9)); % d2 has sorted house index
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
plot(time(1:H,1),yoe212(ko:ko+H-1,1),'y-', ...
        time(1:H,1),yoe212(ko:ko+H-1,2),'m-', ...
        time(1:H,1),yoe212(ko:ko+H-1,3),'c-', ...
        time(1:H,1),yoe212(ko:ko+H-1,4),'r-', ...
        time(1:H,1),yoe212(ko:ko+H-1,5),'g-', ...
        time(1:H,1),yoe212(ko:ko+H-1,6),'b-', ...
        time(1:H,1),yoe212(ko:ko+H-1,7),'w-', ...
        time(1:H,1),yoe212(ko:ko+H-1,8),'r-', ...
        time(1:H,1),yoe212(ko:ko+H-1,9),'g-')
        ylabel('Pressure Deviation  [psi]')
        hy=get(gca, 'ylabel');
        set(hy, 'FontName', 'Palatino')
        xlabel('Time  [sec]')
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
        if step > 0
            axis([0 (H+1)*T -.5001 1.501])      % for F2 increases
            set(gca, 'FontName', 'Palatino')
            md=(max(yoe212(:,d1(9))))/2-0.1;
            for j=1:1:9
                text(3200+(j*550), md, ...
                    [HOUSE(d1(j),1), HOUSE(d1(j),2)], ...
```

```
                    'VerticalAlignment','top', ...
                    'HorizontalAlignment','right')
                hold on
                plot([3200+(j*550) 3700+(j*550)], ...
                    [md yoe212(ko+(3700+j*550)/T, d1(j))])
            end
            fig=input( ...
'Enter separate model P1 secondary component fig number: ', ...
                's')
            h=text(4500, -0.92, ['Figure ', fig, ...
                '. P1 Secondary Components for Separate Model
from Data Set ', ...
                num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
        else
            axis([0 (H+1)*T -1.5001 0.5001])    % for F2 decreases
            set(gca, 'FontName', 'Palatino')
            md=(min(yoe212(:,d1(1))))/2+0.1;
            for j=1:1:9
                text(3200+(j*550), md, ...
                    [HOUSE(d1(10-j),1), HOUSE(d1(10-j),2)], ...
                    'VerticalAlignment','bottom', ...
                    'HorizontalAlignment','right')
                hold on
                plot([3200+(j*550) 3700+(j*550)], ...
                    [md yoe212(ko+(3700+j*550)/T, d1(10-j))])
            end
            fig=input( ...
'Enter separate model P1 secondary component fig number: ', ...
                's')
            h=text(4500, -1.92, ['Figure ', fig, ...
'. P1 Secondary Components for Separate Model from Data S ', ...
                num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
        end
        hold off
        grid
pause
print -dps2
%
% Save information used to set up calculations for individual
% data set.
IND(1)=data;                % data set number
IND(2)=hs;                  % loader house no (A0=1,...F0=8,SY=9)
IND(3)=LSi;                 % initial LS loader position in percent
IND(4)=LSf;                 % final LS loader position in percent
```

```
IND(5)=Patm;          % atmospheric pressure in psia
IND(6)=P1o;           % initial P1 pressure at stepped loader
IND(7)=ko;            % point prior to step input
IND(8)=ki1;           % initial sample to build P1 extention
IND(9)=kf1;           % final sample to build P1 extension
IND(10)=L;            % total length of data record created
IND(11)=kx;           % length of extension of data points
IND(12)=H;            % number of points in final plot
IND(13)=kn;           % length of original data record
IND(14)=T;            % sample time in seconds
IND(15)=step;         % flow step size
```

A.6 Matlab Program Micro2mod.m

```
% A program to model P2 responding to compressor flows. For P2,
% noe of both [[3] [3] [1]] and [[3 3][3 3][1 1]] are used to see
% differences between 1 input and 2 input results. Initialization
% using arx [[3] [3] [1]] is used on sum of primary and secondary
% flows to come up with initial values for parameters. Use the
% perturbed house flow as one input and the sum of all other
% flows as a second input. This gives all flows, except the
% perturbed one, equal weighting. Modeling and simulation are
% done with data manipulated to give zero value at first time
% stamp. To achive this, data up to first time stamp is taken as
% the average of values up to that point, and this average is then
% subtracted from all data. The last values of the data are
% averaged and used to extend the data tail.
%
data=3;                     % data set number
hs=2;                       % loader house no (A0=1,...F0=8,SY=9)
LSi=92.437;                 % initial LS loader position in percent
LSf=100;                    % final LS loader position in percent
Patm=14.557;                % atmospheric pressure in psia
P1o=1.700;                  % initial P1 pressure at stepped loader
ko=60;                      % point prior to step input
ki2=680;                    % initial sample to build P2 extention
kf2=720;                    % final sample to build P2 extension
L=2400;                     % total length of created data record
kx=L-size(D,1);             % length of an extension of data
H=1800;                     % number of points in final plot
kn=size(D,1);               % length of data original data record
T=5;                        % sample time in seconds
%
IND=zeros(15,1);            % storage of input parameters
TH2=zeros(9,54);
time=0:5:(L-1)*5;           % in seconds
time=time';
HOUSE=['A0';'B0';'BA';'C0';'D0';'E0';'EA';'F0';'SY'];
INPUT2=['1 PRIMARY';'SECONDARY'];
%
% Manipulate the input, for use with P2, to smooth initial
conditions and add tail.
      zoe2=zeros(L,4);        % initialze to zero
      zoe2(1:L-kx,2:3)=[D(:,20) (D(:,19)+D(:,21)+D(:,22)+D(:,23) ...
            +D(:,24)+D(:,25)+D(:,26)+D(:,27)-D(:,28)-D(:,29))];
                              % raw data
      zooe2(1,2:3)=mean(zoe2(1:ko,2:3));      % average up to step
      zoe2(1:ko,2)=ones(ko,1)*zooe2(1,2);     % set all values up to
```

```
                                        % the step to xo value
    zoe2(1:ko,3)=ones(ko,1)*zooe2(1,3);
    zoe2(L-kx+1:L,2)=ones(kx,1)*mean(zoe2(ki2:kf2,2));
                                        % build an extention
    zoe2(L-kx+1:L,3)=ones(kx,1)*mean(zoe2(ki2:kf2,3));
    zoe2(:,2)=zoe2(:,2)-zooe2(1,2);     % subtract initial
                                    % input value from all
    zoe2(:,3)=zoe2(:,3)-zooe2(1,3);
    zoe2(:,4)=zoe2(:,2)+zoe2(:,3);      % add the two inputs
                                        % for plotting
%
% Plot the flow input data.
pno=input('Enter flow page number: ','s')
figure(1)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
plot(time(1:H), zoe2(1:H,2), 'y', time(1:H), zoe2(1:H,3), 'm')
    ylabel('Flow Deviation  [g/s]')
    hy=get(gca, 'ylabel');
    set(hy, 'FontName', 'Palatino')
    xlabel('Time  [sec]')
    hx=get(gca, 'xlabel');
    set(hx, 'FontName', 'Palatino')
        axis([0 (H+1)*T -20 20])
        set(gca, 'FontName', 'Palatino')
        fig=input('Enter separate flow figure number: ','s')
        h=text(4500, -28.6, ['Figure ', fig, ....
    '. Primary Flow and Secondary Flow from Data Set ', ...
            num2str(data)]);
        set(h, 'HorizontalAlignment', 'center')
        h=text(6000, zoe2(6000/5,2)-2, ['Primary Flow']);
        set(h, 'HorizontalAlignment', 'left')
        h=text(6000, zoe2(6000/5,3)-2, ['Secondary Flow']);
        set(h, 'HorizontalAlignment', 'left')
    grid
    h=text(9270, 26.2, pno);
    set(h, 'HorizontalAlignment', 'right')
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
plot(time(1:H), zoe2(1:H,4), 'c')
    ylabel('Flow Deviation  [g/s]')
    hy=get(gca, 'ylabel');
    set(hy, 'FontName', 'Palatino')
    xlabel('Time  [sec]')
```

```
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
                axis([0 (H+1)*T -20 20])
                set(gca, 'FontName', 'Palatino')
                fig=input('Enter combined flow figure number: ','s')
                h=text(4500, -28.6, ['Figure ', fig, ...
'. Sum of Primary Flow and Secondary Flow from Data Set ', ...
                        num2str(data)]);
                set(h, 'HorizontalAlignment', 'center')
                h=text(6000, zoe2(6000/5,4)-2, ['Sum of Flows']);
                set(h, 'HorizontalAlignment', 'left')
        grid
print -dps2
pause
%
% Flow step size calculation for loader changes.
Pc=15.9;                % base pressure for loader/flow calibration
C(1:6)=[0.0015928689 -0.40467791 34.59480361 -958.9842469
5.371698113 -482.3767296];
% for initial loader position
        if LSi < 98
                F2i=C(1)*LSi^3+C(2)*LSi^2+C(3)*LSi+C(4);
                F2i=(F2i*(P1o+Patm)/Pc);
        else
                F2i=C(5)*LSi+C(6);
                F2i=(F2i*(P1o+Patm)/Pc);
        end
% for final loader position
        if LSf < 98
                F2f=C(1)*LSf^3+C(2)*LSf^2+C(3)*LSf+C(4);
                F2f=(F2f*(P1o+Patm)/Pc);
        else
                F2f=C(5)*LSf+C(6);
                F2f=(F2f*(P1o+Patm)/Pc);
        end
step=F2f-F2i                    % initial flow step size
%
% Calculate parameter values for all houses using one input
initialization.
        yoe121=zeros(L,9);              % matrix to hold individual
                                %input 1 simulation, single input
        yoe221=zeros(L,9);              % matrix to hold individual
                                % input 2 simulation, single input
        yoeS21=zeros(L,9);              % matrix to hold sum of inputs 1
                                % and 2 simulation, single
        yoe122=zeros(L,9);              % matrix to hold individual
```

```
                                % input 1 simulation, two inputs
      yoe222=zeros(L,9);              % matrix to hold individual
                                % input 2 simulation, two inputs
      yoeS22=zeros(L,9);              % matrix to hold sum of inputs 1
                                      % and 2 simulation, single
      narx21=[[3] [3] [1]];           % one third order output, one
                                      % input with one delay
      noe21=[[3] [3] [1]];            % one third order output, one
                                      % input with one delay
      noe22=[[3 3] [3 3] [1 1]];      % one third order output, two
                                      % input with one delay
%
i=1;  %index for the P1 output location used for initiallization
% Manipulate the outputs similarly to the input.
      zoe2(1:L-kx,1)=D(:,9+i);              % raw P2 data
      zooe2(1,1)=mean(zoe2(1:ko,1));        % average up to step
      zoe2(1:ko,1)=ones(ko,1)*zooe2(1,1);       % set all values up to
                                                % step to equal yo
      zoe2(L-kx+1:L,1)=ones(kx,1)*mean(zoe2(ki2:kf2,1));
                                            % build the extension
      zoe2(:,1)=zoe2(:,1)-zooe2(1,1);       % subtract initial output
                                            % value from all
%
% Compute the combined single input arx model for parameter
% initialization purposes.
      tharx2ci=arx([zoe2(:,1) zoe2(:,4)], narx21);     % c=combined,
                                                       % i=initial
% Use the arx result in an initial theta built for an oe model with
% one input
      thoe2ci=oe([zoe2(:,1) zoe2(:,4)], noe21);        % get single
                                                   % input theta form
      thoe2ci(3,1)=tharx2ci(3,4);           % b1
      thoe2ci(3,2)=tharx2ci(3,5);           % b2
      thoe2ci(3,3)=tharx2ci(3,6);           % b3
      thoe2ci(3,4)=tharx2ci(3,1);           % f1
      thoe2ci(3,5)=tharx2ci(3,2);           % f2
      thoe2ci(3,6)=tharx2ci(3,3);           % f3
% Use the arx result in an initial theta built for oe model with
% two inputs
      thoe2si=oe([zoe2(:,1) zoe2(:,2) zoe2(:,3)], noe22);   % get oe
                                                            % theta form
      thoe2si(3,1)=tharx2ci(3,4);           % b11
      thoe2si(3,4)=tharx2ci(3,4);           % b12
      thoe2si(3,2)=tharx2ci(3,5);           % b21
      thoe2si(3,5)=tharx2ci(3,5);           % b22
      thoe2si(3,3)=tharx2ci(3,6);           % b31
```

```
      thoe2si(3,6)=tharx2ci(3,6);          % b32
      thoe2si(3,7)=tharx2ci(3,1);          % f11
      thoe2si(3,10)=tharx2ci(3,1);         % f12
      thoe2si(3,8)=tharx2ci(3,2);          % f21
      thoe2si(3,11)=tharx2ci(3,2);         % f22
      thoe2si(3,9)=tharx2ci(3,3);          % f31
      thoe2si(3,12)=tharx2ci(3,3);         % f32
%
for i=1:9                        %index for the P2 output location
% Manipulate the outputs similarly to the input.
      zoe2(1:L-kx,1)=zeros(L-kx,1);
      zoe2(1:L-kx,1)=D(:,9+i);             % raw P2 data
      zooe2(1,1)=mean(zoe2(1:ko,1));       % average up to step
      zoe2(1:ko,1)=ones(ko,1)*zooe2(1,1);      % set all values up to
                                               % step to equal yo
      zoe2(L-kx+1:L,1)=ones(kx,1)*mean(zoe2(ki2:kf2,1));
                                           % build the extension
      zoe2(:,1)=zoe2(:,1)-zooe2(1,1);      % subtract initial output
                                           %value from all
%
% Compute a combined single input arx model for parameter
% initialization purposes. Use the arx result in an initial theta
% built for an oe model with one input
      thoe2ci=oe([zoe2(:,1) zoe2(:,4)], noe21);       % get single
                                               % input theta form
      thoe2ci(3,1)=tharx2ci(3,4);          % b1
      thoe2ci(3,2)=tharx2ci(3,5);          % b2
      thoe2ci(3,3)=tharx2ci(3,6);          % b3
      thoe2ci(3,4)=tharx2ci(3,1);          % f1
      thoe2ci(3,5)=tharx2ci(3,2);          % f2
      thoe2ci(3,6)=tharx2ci(3,3);          % f3
% Calculate the combined flow input model using initialization
from above, and simulate.
      thoe2c=oe([zoe2(:,1) zoe2(:,4)], thoe2ci);  % obtain the model
      yoe2(:,i)=idsim([zoe2(:,4)],thoe2c);       % combined input
                                           % model output result
      seoe21=(yoe2(1:L-kx,i)-zoe2(1:L-kx,1)).^2;     % sum of
                                           % squared errors between
      sseoe21=sum(seoe21);                 % model and data
% Look at the zero-pole structure.
[zepo21, K21]=th2zp(thoe2c);               % zero, poles, and gain
% Save 27 items that characterize the results of the combined flow
% model in TH2.
% First save the parameter values and corresponding gain, zero,
% and poles.
TH2(i,1)=thoe2c(3,1);                      % b1
```

```
TH2(i,2)=thoe2c(3,2);                    % b2
TH2(i,3)=thoe2c(3,3);                    % b3
TH2(i,4)=thoe2c(3,4);                    % f1
TH2(i,5)=thoe2c(3,5);                    % f2
TH2(i,6)=thoe2c(3,6);                    % f3
TH2(i,7)=K21(2,1);    % K (static gain, NOT transfer function gain)
TH2(i,8)=zepo21(2,1);                    % za
TH2(i,9)=zepo21(3,1);                    % zb
TH2(i,10)=zepo21(2,3);                   % pa
TH2(i,11)=zepo21(3,3);                   % pb
TH2(i,12)=zepo21(4,3);                   % pc
% Second, save the standard deviation of above values.
TH2(i,13)=thoe2c(4,1);                   % b1
TH2(i,14)=thoe2c(5,2);                   % b2
TH2(i,15)=thoe2c(6,3);                   % b3
TH2(i,16)=thoe2c(7,4);                   % f1
TH2(i,17)=thoe2c(8,5);                   % f2
TH2(i,18)=thoe2c(9,6);                   % f3
TH2(i,19)=K21(3,1);                      % K
TH2(i,20)=zepo21(2,2);                   % za
TH2(i,21)=zepo21(3,2);                   % zb
TH2(i,22)=zepo21(2,4);                   % pa
TH2(i,23)=zepo21(3,4);                   % pb
TH2(i,24)=zepo21(4,4);                   % pc
% Third, save miscellaneous values.
TH2(i,25)=thoe2c(1,1);                   % variance of the noise
TH2(i,26)=thoe2c(2,1);              % Akaike's final prediction error
TH2(i,27)=sseoe21;    % sum of the squared errors, unfiltered
%
% Calculate the portions of the P2 output made up by the
% individual inputs using the parameters from combined input
oe model.
%     Simulate model output for single primary flow input.
      zoe12=zeros(L,2);
      zoe12(:,1:2)=[zoe2(:,1) zoe2(:,2)];
      yoe121(:,i)=idsim(zoe12(:,2), thoe2c);   % portion of output
                                               % due to primary flow
%     Simulate model output for single secondary flow input.
      zoe22=zeros(L,2);
      zoe22(:,1:2)=[zoe2(:,1) zoe2(:,3)];
      yoe221(:,i)=idsim(zoe22(:,2), thoe2c);   % portion of output
                                               % due to secondary flow
%     Sum all the components to see that it represents whole
%     response.
      yoeS21(:,i)=yoe121(:,i)+yoe221(:,i);     % sum of two
                                               % separate output parts
```

```
% Plot these results later, after the two input model is made.
%
% Calculate a separate flow input model, using the above
% initialization parameters. Use the arx result in an initial theta
% built for oe model with two inputs. Note this is still the
% tharx2ci computed prior to entering this loop.
     thoe2si=oe([zoe2(:,1) zoe2(:,2) zoe2(:,3)], noe22);   % get oe
                                                           % theta form
     thoe2si(3,1)=tharx2ci(3,4);          % b11
     thoe2si(3,4)=tharx2ci(3,4);          % b12
     thoe2si(3,2)=tharx2ci(3,5);          % b21
     thoe2si(3,5)=tharx2ci(3,5);          % b22
     thoe2si(3,3)=tharx2ci(3,6);          % b31
     thoe2si(3,6)=tharx2ci(3,6);          % b32
     thoe2si(3,7)=tharx2ci(3,1);          % f11
     thoe2si(3,10)=tharx2ci(3,1);         % f12
     thoe2si(3,8)=tharx2ci(3,2);          % f21
     thoe2si(3,11)=tharx2ci(3,2);         % f22
     thoe2si(3,9)=tharx2ci(3,3);          % f31
     thoe2si(3,12)=tharx2ci(3,3);         % f32
%
     thoe2s=oe([zoe2(:,1) zoe2(:,2) zoe2(:,3)], thoe2si);
     yoe2(:,i)=idsim([zoe2(:,2) zoe2(:,3)],thoe2s);  % two input
                                          % model output result
     seoe22=(yoe2(1:L-kx,i)-zoe2(1:L-kx,1)).^2;     % sum of
                                          % squared errors between
     sseoe22=sum(seoe22);                 % model and data
% Look at the zero-pole structure.
[zepo22, K22]=th2zp(thoe2s);              % zero, poles, and gain
% Save 51 items that characterize the results of the separate flow
% model in TH2.
% Save the parameter values and corresponding gain, zero, and
poles for primary input.
TH2(i,28)=thoe2s(3,1);                    % b11
TH2(i,29)=thoe2s(3,2);                    % b21
TH2(i,30)=thoe2s(3,3);                    % b31
TH2(i,31)=thoe2s(3,7);                    % f11
TH2(i,32)=thoe2s(3,8);                    % f21
TH2(i,33)=thoe2s(3,9);                    % f31
TH2(i,34)=K22(2,1);  % K1 (static gain, NOT transfer function gain)
TH2(i,35)=zepo22(2,1);                    % za1
TH2(i,36)=zepo22(3,1);                    % zb1
TH2(i,37)=zepo22(2,3);                    % pa1
TH2(i,38)=zepo22(3,3);                    % pb1
TH2(i,39)=zepo22(4,3);                    % pc1
% Second, save the standard deviation of above values.
```

```
TH2(i,40)=thoe2s(4,1);                    % b11
TH2(i,41)=thoe2s(5,2);                    % b21
TH2(i,42)=thoe2s(6,3);                    % b31
TH2(i,43)=thoe2s(10,7);                   % f11
TH2(i,44)=thoe2s(11,8);                   % f21
TH2(i,45)=thoe2s(12,9);                   % f31
TH2(i,46)=K22(3,1);                       % K1
TH2(i,47)=zepo22(2,2);                    % za1
TH2(i,48)=zepo22(3,2);                    % zb1
TH2(i,49)=zepo22(2,4);                    % pa1
TH2(i,50)=zepo22(3,4);                    % pb1
TH2(i,51)=zepo22(4,4);                    % pc1
% Save the parameter values and corresponding gain, zero, and
% poles for secondary input.
TH2(i,52)=thoe2s(3,4);                    % b12
TH2(i,53)=thoe2s(3,5);                    % b22
TH2(i,54)=thoe2s(3,6);                    % b32
TH2(i,55)=thoe2s(3,10);                   % f12
TH2(i,56)=thoe2s(3,11);                   % f22
TH2(i,57)=thoe2s(3,12);                   % f32
TH2(i,58)=K22(2,2);  % K2 (static gain, NOT transfer function gain)
TH2(i,59)=zepo22(2,5);                    % za2
TH2(i,60)=zepo22(3,5);                    % zb2
TH2(i,61)=zepo22(2,7);                    % pa2
TH2(i,62)=zepo22(3,7);                    % pb2
TH2(i,63)=zepo22(4,7);                    % pc2
% Second, save the standard deviation of above values.
TH2(i,64)=thoe2s(7,4);                    % b12
TH2(i,65)=thoe2s(8,5);                    % b22
TH2(i,66)=thoe2s(9,6);                    % b32
TH2(i,67)=thoe2s(8,5);                    % f12
TH2(i,68)=thoe2s(9,6);                    % f22
TH2(i,69)=thoe2s(9,6);                    % f32
TH2(i,70)=K22(3,1);                       % K2
TH2(i,71)=zepo22(2,6);                    % za2
TH2(i,72)=zepo22(3,6);                    % zb2
TH2(i,73)=zepo22(2,8);                    % pa2
TH2(i,74)=zepo22(3,8);                    % pb2
TH2(i,75)=zepo22(4,8);                    % pc2
% Third, save miscellaneous values.
TH1(i,76)=thoe2s(1,1);                    % variance of the noise
TH1(i,77)=thoe2s(2,1);              % Akaike's final prediction error
TH1(i,78)=sseoe22;     % sum of the squared errors, unfiltered
%
% Calculate the portions of the P2 output made up by
% individual inputs using the oe parameters from the separate
```

```
% input model.
%     Simulate model output for single primary flow input.
      thoe2t=oe([zoe2(:,1), zoe2(:,2)], noe21); % a temporary one
                  %input theta form filled with parameter values
      thoe2t(3,1)=thoe2s(3,1);              % b11
      thoe2t(3,2)=thoe2s(3,2);              % b21
      thoe2t(3,3)=thoe2s(3,3);              % b31
      thoe2t(3,4)=thoe2s(3,7);              % f11
      thoe2t(3,5)=thoe2s(3,8);              % f21
      thoe2t(3,6)=thoe2s(3,9);              % f31
      zoe21=zeros(L,2);
      zoe21(:,1:2)=[zoe2(:,1) zoe2(:,2)];
      yoe122(:,i)=idsim(zoe21(:,2), thoe2t);    % portion of output
                                            % due to primary flow
%     Simulate model output for single secondary flow input.
      thoe2t(3,1)=thoe2s(3,4);              % b12
      thoe2t(3,2)=thoe2s(3,5);              % b22
      thoe2t(3,3)=thoe2s(3,6);              % b32
      thoe2t(3,4)=thoe2s(3,10);             % f12
      thoe2t(3,5)=thoe2s(3,11);             % f22
      thoe2t(3,6)=thoe2s(3,12);             % f32
      zoe22=zeros(L,2);
      zoe22(:,1:2)=[zoe2(:,1) zoe2(:,3)];
      yoe222(:,i)=idsim(zoe22(:,2), thoe2t);    % portion of output
                                            % due to secondary flow
%     Sum all the components.
      yoeS22(:,i)=yoe122(:,i)+yoe222(:,i);        % sum of two
                                            % separate output parts
%
% Plot both combined input and two input model outputs, along
with data.
pno=input(['Enter  ', HOUSE(i,1), HOUSE(i,2), ...
      '  both model page number: '],'s')
figure(2)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
% Plot for the combined flow input-model
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
      plot(time(1:H,1), yoeS21(1:H,i), 'y-', ...
            time(1:ko,1), D(1:ko,i+9)-zooe2(1,1), 'g:', ...
            time(ko+1:L-kx,1), zoe2(ko+1:L-kx,1), 'g:')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
```

```
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
            axis([0 (H+1)*T -4 4])
            set(gca, 'FontName', 'Palatino')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                  ' P2 figure number: '],'s')
            h=text(4500, -5.7, ['Figure ', fig, '. ', ...
                  HOUSE(i,1), HOUSE(i,2), ...
      ' P2 Data and P2 Combined Flow Model from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      h=text(9270, 5.17, pno);
      set(h, 'HorizontalAlignment', 'right')
      grid
% Plot for the separate flow input model
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
      plot(time(1:H,1), yoeS22(1:H,i), 'm-', ...
            time(1:ko,1), D(1:ko,i+9)-zooe2(1,1), 'g:', ...
             time(ko+1:L-kx,1), zoe2(ko+1:L-kx,1), 'g:')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
            axis([0 (H+1)*T -4 4])
            set(gca, 'FontName', 'Palatino')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                  ' P2 figure number: '],'s')
            h=text(4500, -5.7, ['Figure ', fig, '. ', ...
                  HOUSE(i,1), HOUSE(i,2), ...
      ' P2 Data and P2 Separate Flow Model from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      grid
pause
print -dps2
%
% Plot both combined input and two input model output
components and sum
pno=input(['Enter  ', HOUSE(i,1), HOUSE(i,2), ...
      ' components page number: '],'s')
figure(3)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
```

```
% Plot for the combined flow input model
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
      plot(time(1:H,1), yoe121(ko:ko+H-1,i), 'y-', ...
            time(1:H,1), yoe221(ko:ko+H-1,i), 'm', ...
            time(1:H,1), yoeS21(ko:ko+H-1,i), 'r')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
            axis([0 (H+1)*T -4 4])  % for F2 increases and decreases
            set(gca, 'FontName', 'Palatino')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2),  ...
                  ' P2 figure number: '],'s')
            h=text(4500, -5.7, ['Figure ', fig, '.  ', ...
                  HOUSE(i,1), HOUSE(i,2), ...
      ' P2 Combined Flow Model Components from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
            h=text(6000, yoe121(6000/5,i)-0.3, ...
                  ['Primary Component']);
            set(h, 'HorizontalAlignment', 'center')
            h=text(6000, yoeS21(6000/5,i)-0.3, ...
                  ['Sum of Components']);
            set(h, 'HorizontalAlignment', 'center')
            h=text(6000, yoe221(6000/5,i)-0.3, ...
                  ['Secondary Component']);
            set(h, 'HorizontalAlignment', 'center')
      h=text(9270, 5.17, pno);
      set(h, 'HorizontalAlignment', 'right')
      grid
% Plot for the separate flow input model
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
      plot(time(1:H,1), yoe122(ko:ko+H-1,i), 'y-', ...
            time(1:H,1), yoe222(ko:ko+H-1,i), 'm', ...
            time(1:H,1), yoeS22(ko:ko+H-1,i), 'r')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
      xlabel('Time  [sec]')
      hx=get(gca, 'xlabel');
      set(hx, 'FontName', 'Palatino')
            axis([0 (H+1)*T -4 4])  % for F2 increases and decreases
```

```
            set(gca, 'FontName', 'Palatino')
            fig=input(['Enter ', HOUSE(i,1), HOUSE(i,2), ...
                  ' P2 figure number: '],'s')
            h=text(4500, -5.7, ['Figure ', fig, '. ', ...
                  HOUSE(i,1), HOUSE(i,2), ...
      ' P2 Separate Flow Model Components from Data Set ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
            h=text(6000, yoe122(6000/5,i)-0.3, ...
                  ['Primary Component']);
            set(h, 'HorizontalAlignment', 'center')
            h=text(6000, yoeS22(6000/5,i)-0.3, ...
                  ['Sum of Components']);
            set(h, 'HorizontalAlignment', 'center')
            h=text(6000, yoe222(6000/5,i)-0.3, ...
                  ['Secondary Component']);
            set(h, 'HorizontalAlignment', 'center')
      grid
pause
print -dps
end
%
% Plot all the P2 simulations for the combined flow model type.
%
% First, plot the primary flow components for the combined flow
model.
% Sort house names on basis of last value in plot.
[s1,d1]=sort(yoe121(L-kx-ko,1:9)); % d1 has sorted house index
                                   % based on last data; 1=smallest
pno=input('Enter all models page number: ','s')
figure(4)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
plot(time(1:H,1),yoe121(ko:ko+H-1,1),'y-', ...
      time(1:H,1),yoe121(ko:ko+H-1,2),'m-', ...
      time(1:H,1),yoe121(ko:ko+H-1,3),'c-', ...
      time(1:H,1),yoe121(ko:ko+H-1,4),'r-', ...
      time(1:H,1),yoe121(ko:ko+H-1,5),'g-', ...
      time(1:H,1),yoe121(ko:ko+H-1,6),'b-', ...
      time(1:H,1),yoe121(ko:ko+H-1,7),'w-', ...
      time(1:H,1),yoe121(ko:ko+H-1,8),'r-', ...
      time(1:H,1),yoe121(ko:ko+H-1,9),'g-')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
```

```
        set(hy, 'FontName', 'Palatino')
        xlabel('Time [sec]')
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
        if step > 0
                axis([0 (H+1)*T -1 4])  % for F2 increases
                set(gca, 'FontName', 'Palatino')
                md=(max(yoe121(:,d1(9))))/2;
                for j=1:1:9
                        text(3200+(j*550), md, ...
                                [HOUSE(d1(j),1), HOUSE(d1(j),2)], ...
                                'VerticalAlignment','top', ...
                                'HorizontalAlignment','right')
                        hold on
                        plot([3200+(j*550) 3700+(j*550)], ...
                                [md yoe121(ko+(3700+j*550)/T, d1(j))])
                end
                fig=input( ...
'Enter combined model P2 primary component fig number: ', 's')
                h=text(4500, -2.073, ['Figure ', fig, ...
'. P2 Primary Component for Combined Model from Data Set ', ...
                        num2str(data)]);
                set(h, 'HorizontalAlignment', 'center')
        h=text(9270, 4.732, pno);
        set(h, 'HorizontalAlignment', 'right')
        else
                axis([0 (H+1)*T -4 1])  % for F2 decreases
                set(gca, 'FontName', 'Palatino')
                md=(min(yoe121(:,d1(1))))/2;
                for j=1:1:9
                        text(3200+(j*550), md, ...
                                [HOUSE(d1(10-j),1), HOUSE(d1(10-j),2)], ...
                                'VerticalAlignment','bottom', ...
                                'HorizontalAlignment','right')
                        hold on
                        plot([3200+(j*550) 3700+(j*550)], ...
                                [md yoe121(ko+(3700+j*550)/T, d1(10-j))])
                end
                fig=input( ...
'Enter combined model P2 primary component fig number: ', 's')
                h=text(4500, -5.073, ['Figure ', fig, ...
'. P2 Primary Component for Combined Model from Data Set ', ...
                        num2str(data)]);
                set(h, 'HorizontalAlignment', 'center')
        h=text(9270, 1.732, pno);
        set(h, 'HorizontalAlignment', 'right')
```

```
        end
        hold off
        grid
% Second, plot the secondary flow components for the combined
% flow model.
% Sort house names on basis of last value in plot.
[s2,d2]=sort(yoe221(L-kx-ko,1:9));      % d2 has sorted house index
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
plot(time(1:H,1),yoe221(ko:ko+H-1,1),'y-', ...
        time(1:H,1),yoe221(ko:ko+H-1,2),'m-', ...
        time(1:H,1),yoe221(ko:ko+H-1,3),'c-', ...
        time(1:H,1),yoe221(ko:ko+H-1,4),'r-', ...
        time(1:H,1),yoe221(ko:ko+H-1,5),'g-', ...
        time(1:H,1),yoe221(ko:ko+H-1,6),'b-', ...
        time(1:H,1),yoe221(ko:ko+H-1,7),'w-', ...
        time(1:H,1),yoe221(ko:ko+H-1,8),'r-', ...
        time(1:H,1),yoe221(ko:ko+H-1,9),'g-')
        ylabel('Pressure Deviation  [psi]')
        hy=get(gca, 'ylabel');
        set(hy, 'FontName', 'Palatino')
        xlabel('Time  [sec]')
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
        if step > 0
                axis([0 (H+1)*T -4 1])          % for F2 increases
                set(gca, 'FontName', 'Palatino')
                md=(min(yoe221(:,d2(9))))/2;
                for j=1:1:9
                        text(3200+(j*550), md, ...
                                [HOUSE(d2(10-j),1), HOUSE(d2(10-j),2)], ...
                                'VerticalAlignment','bottom', ...
                                'HorizontalAlignment','right')
                        hold on
                        plot([3200+(j*550) 3700+(j*550)], ...
                                [md yoe221(ko+(3700+j*550)/T, d2(10-j))])
                end
                fig=input( ...
'Enter combined model P2 secondary component fig number: ','s')
                h=text(4500, -5.073, ['Figure ', fig, ...
'.  P2 Secondary Component for Combined Model from Data S ', ...
                        num2str(data)]);
                set(h, 'HorizontalAlignment', 'center')
        else
                axis([0 (H+1)*T -1 4])  % for F2 decreases
                set(gca, 'FontName', 'Palatino')
```

```
            md=(max(yoe221(:,d2(1))))/2;
            for j=1:1:9
                  text(3200+(j*550), md, ...
                        [HOUSE(d2(j),1), HOUSE(d2(j),2)], ...
                        'VerticalAlignment','top', ...
                        'HorizontalAlignment','right')
                  hold on
                  plot([3200+(j*550) 3700+(j*550)], ...
                        [md yoe221(ko+(3700+j*550)/T, d2(j))])
            end
            fig=input( ...
'Enter combined model P2 secondary component fig number: ','s')
            h=text(4500, -2.073, ['Figure ', fig, ...
'. P2 Secondary Component for Combined Model from Data S ', ...
                  num2str(data)]);
            set(h, 'HorizontalAlignment', 'center')
      end
      hold off
      grid
pause
print
%
% Plot all the P2 simulations for the separate flow model type.
% First, plot the primary flow components for the separate flow
model.
% Sort house names on basis of last value in plot.
[s1,d1]=sort(yoe122(L-kx-ko,1:9));        % d1 has sorted house
                                    % index based on last data; 1=smallest
pno=input('Enter all models page number: ','s')
figure(5)
set(gcf, 'DefaultTextFontName', 'Palatino')
orient tall
subplot(2,1,1)
set(gca,'Position',[0.258, 0.630, 0.625, 0.302])
plot(time(1:H,1),yoe122(ko:ko+H-1,1),'y-', ...
      time(1:H,1),yoe122(ko:ko+H-1,2),'m-', ...
      time(1:H,1),yoe122(ko:ko+H-1,3),'c-', ...
      time(1:H,1),yoe122(ko:ko+H-1,4),'r-', ...
      time(1:H,1),yoe122(ko:ko+H-1,5),'g-', ...
      time(1:H,1),yoe122(ko:ko+H-1,6),'b-', ...
      time(1:H,1),yoe122(ko:ko+H-1,7),'w-', ...
      time(1:H,1),yoe122(ko:ko+H-1,8),'r-', ...
      time(1:H,1),yoe122(ko:ko+H-1,9),'g-')
      ylabel('Pressure Deviation  [psi]')
      hy=get(gca, 'ylabel');
      set(hy, 'FontName', 'Palatino')
```

```
xlabel('Time  [sec]')
hx=get(gca,'xlabel');
set(hx,'FontName','Palatino')
if step > 0
      axis([0 (H+1)*T -1 4])  % for F2 increases
      set(gca,'FontName','Palatino')
      md=(max(yoe122(:,d1(9))))/2;
      for j=1:1:9
            text(3200+(j*550), md, ...
                  [HOUSE(d1(10-j),1), HOUSE(d1(10-j),2)], ...
                  'VerticalAlignment','top', ...
                  'HorizontalAlignment','right')
            hold on
            plot([3200+(j*550) 3700+(j*550)], ...
                  [md yoe122(ko+(3700+j*550)/T, d1(10-j))])
      end
      fig=input( ...
'Enter separate model P2 primary component figure number: ','s')
      h=text(4500, -2.073, ['Figure ', fig, ...
'. P2 Primary Components for Separate Model from Data Set ', ...
            num2str(data)]);
      set(h,'HorizontalAlignment','center')
h=text(9270, 4.732, pno);
set(h,'HorizontalAlignment','right')
else
      axis([0 (H+1)*T -4 1])  % for F2 decreases
      set(gca,'FontName','Palatino')
      md=(min(yoe122(:,d1(1))))/2;
      for j=1:1:9
            text(3200+(j*550), md, ...
                  [HOUSE(d1(j),1), HOUSE(d1(j),2)], ...
                  'VerticalAlignment','bottom', ...
                  'HorizontalAlignment','right')
            hold on
            plot([3200+(j*550) 3700+(j*550)], ...
                  [md yoe122(ko+(3700+j*550)/T, d1(j))])
      end
      fig=input( ...
'Enter separate model P2 primary component figure number: ','s')
      h=text(4500, -5.073, ['Figure ', fig, ...
'. P2 Primary Components for Separate Model from Data Set ', ...
            num2str(data)]);
      set(h,'HorizontalAlignment','center')
h=text(9270, 1.732, pno);
set(h,'HorizontalAlignment','right')
end
```

```
        hold off
        grid
% Second, plot the secondary flow components for the separate
% flow model.
% Sort house names on basis of last value in plot.
[s2,d2]=sort(yoe222(L-kx-ko,1:9));        % d2 has sorted house index
subplot(2,1,2)
set(gca,'Position',[0.258, 0.160, 0.625, 0.302])
plot(time(1:H,1),yoe222(ko:ko+H-1,1),'y-', ...
        time(1:H,1),yoe222(ko:ko+H-1,2),'m-', ...
        time(1:H,1),yoe222(ko:ko+H-1,3),'c-', ...
        time(1:H,1),yoe222(ko:ko+H-1,4),'r-', ...
        time(1:H,1),yoe222(ko:ko+H-1,5),'g-', ...
        time(1:H,1),yoe222(ko:ko+H-1,6),'b-', ...
        time(1:H,1),yoe222(ko:ko+H-1,7),'w-', ...
        time(1:H,1),yoe222(ko:ko+H-1,8),'r-', ...
        time(1:H,1),yoe222(ko:ko+H-1,9),'g-')
        ylabel('Pressure Deviation  [psi]')
        hy=get(gca, 'ylabel');
        set(hy, 'FontName', 'Palatino')
        xlabel('Time  [sec]')
        hx=get(gca, 'xlabel');
        set(hx, 'FontName', 'Palatino')
        if step > 0
                axis([0 (H+1)*T -4 1])  % for F2 increases
                set(gca, 'FontName', 'Palatino')
                md=(min(yoe222(:,d1(9))))/2;
                for j=1:1:9
                        text(3200+(j*550), md, ...
                                [HOUSE(d2(j),1), HOUSE(d2(j),2)], ...
                                'VerticalAlignment','bottom', ...
                                'HorizontalAlignment','right')
                        hold on
                        plot([3200+(j*550) 3700+(j*550)], ...
                                [md yoe222(ko+(3700+j*550)/T, d2(j))])
                end
                fig=input( ...
'Enter separate model P2 secondary component fig number: ','s')
                h=text(4500, -5.073, ['Figure ', fig, ...
'. P2 Secondary Components for Separate Model from Data S ', ...
                        num2str(data)]);
                set(h, 'HorizontalAlignment', 'center')
        else
                axis([0 (H+1)*T -1 4])  % for F2 decreases
                set(gca, 'FontName', 'Palatino')
                md=(max(yoe222(:,d2(1))))/2;
```

```
            for j=1:1:9
                text(3200+(j*550), md, ...
                    [HOUSE(d2(10-j),1), HOUSE(d2(10-j),2)], ...
                    'VerticalAlignment','top', ...
                    'HorizontalAlignment','right')
                hold on
                plot([3200+(j*550) 3700+(j*550)], ...
                    [md yoe222(ko+(3700+j*550)/T, d2(10-j))])
            end
            fig=input( ...
'Enter separate model P2 secondary component fig number: ','s')
            h=text(4500, -2.073, ['Figure ', fig, ...
'. P2 Secondary Components for Separate Model from Data S ', ...
                num2str(data)]);
            set(h,'HorizontalAlignment','center')
        end
        hold off
        grid
print -dps2
%
% Save information used to set up calculations for individual
data set.
IND(1)=data;                % data set number
IND(2)=hs;                  % loader house no (A0=1,...F0=8,SY=9)
IND(3)=LSi;                 % initial LS loader position in percent
IND(4)=LSf;                 % final LS loader position in percent
IND(5)=Patm;                % atmospheric pressure in psia
IND(6)=P1o;                 % initial P1 pressure at stepped loader
IND(7)=ko;                  % point prior to step input
IND(8)=ki2;                 % initial sample to build P2 extention
IND(9)=kf2;                 % final sample to build P2 extension
IND(10)=L;                  % total length of data record created
IND(11)=kx;                 % length of extension of data points
IND(12)=H;                  % number of points in final plot
IND(13)=kn;                 % length of original data record
IND(14)=T;                  % sample time in seconds
IND(15)=step;               % flow step size
```

BIBLIOGRAPHY

[Bar85] Barron, R., Cryogenic Systems, Oxford University Press, New York, NY, 1985.

[Col79] Cole, F., Donaldson, M., Edwards, D., Edwards, H., Koehler, P., A Report on the Design of the Fermi National Accelerator Laboratory Superconducting Accelerator, Fermilab, Batavia, IL, May, 1979.

[Lju87] Ljung, L., System Identification: Theory for the User, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1987.

[Lju95] Ljung, L., System Identification Toolbox, The MathWorks, Inc., Natick, MA, 1995.

[Mar96] Martinez, A., Personal Correspondence, Fermilab, Batavia, IL, October, 1996.

[Mar97] Martinez, A., Personal Communication, Fermilab, Batavia, IL, March, 1997.

[Nor94] Norris, B., Brubaker, J., Franck, A., Lublinsky, B., Lucas, P., Theilacker, J., Seino, K., Smolucha, J., "New Cryogenic Controls for the Tevatron Low Temperature Upgrade," Advances in Cryogenic Engineering, Vol. 39, Part B, pp. 1185-1192, Plenum Press, New York, NY, 1994.

[Osi87] Osiadacz, A., Simulation and Analysis of Gas Networks, Gulf Publishing Company, Houston, TX, 1987.

[Rod83] Rode, C., "Tevatron Cryogenic System," Proceedings of the 12th International Conference on High Energy Accelerators, pp. 529-535, Batavia, IL, August, 1983.

[Zie93] Ziemer, R., Tranter, W., Fannin, D., Signals and Systems: Continuous and Discrete, Macmillan Publishing Company, New York, NY, 1993.