

**DYNAMIC MODELING FOR SIMULATION AND CONTROL OF A  
CIRCULATING FLUIDIZED BED COMBUSTOR**

by

**JAMES RIDLEY MUIR**

B.Sc., The University of Calgary, 1986

M.Sc., The University of Calgary, 1989

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR  
THE DEGREE OF DOCTOR OF PHILOSOPHY**

in

**THE FACULTY OF GRADUATE STUDIES**

**(Department of Chemical Engineering)**

We accept this thesis as conforming  
to the required standard

**THE UNIVERSITY OF BRITISH COLUMBIA**

February 1995

© James Ridley Muir, 1995

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature)

Department of CHEMICAL ENGINEERING

The University of British Columbia  
Vancouver, Canada

Date Feb 12/95

## **ABSTRACT**

Circulating fluidized bed (CFB) technology has been recognized by industry as a viable gas/solid contacting process with applications including combustion for power generation and waste incineration, pyrolysis, calcining, and catalysis. One important and growing use of circulating fluidized beds is as utility boilers. In this application, the combustor spends most of its life under transient conditions, following the demand of the steam-side. This, in addition to stricter emission requirements, has led to the increasing importance of understanding the transient nature of the circulating fluidized bed combustion (CFBC) process, specifically for control system design and development. In response to this need, the work in this thesis was carried out to investigate the dynamic behaviour of a circulating fluidized bed combustor.

A dynamic model has been developed which predicts the transient behaviour of the combustion temperature, rate of heat removal by the in-bed heat exchanger, and the flue gas oxygen concentration for a circulating fluidized bed combustor. These three factors fully define the combustor at any time. The model was incorporated into a simulator to provide an environment for the reproduction of the combustion process on computer. The simulator predicts the behaviour of the combustor under either manual or automatic control. In order to accommodate the chemical kinetic behaviour of various fuels, separate tests were carried out in a bench scale fluid bed combustor to provide order-of-magnitude estimates of the kinetic parameters to approximate the reactivity of these fuels in the combustor. To obtain an estimate of the solids circulation rate, experiments were performed to measure this rate as a function of gas velocity, P/S air ratio, and average riser solids loading. These tests were carried out using a novel time-of-descent method referred to as "line-and-sinker" in which a cage immersed in the moving packed bed descends attached to a chain which passes through a sealed port to the outside of the upper region of the standpipe.

Validation of the model is provided through comparison with step response tests carried out on the UBC pilot CFB combustor. Discrepancies are attributed to unmodelled

disturbances rather than from basic model concepts. Further validation, necessary to ensure the applicability of the simulator to control development, is provided through a comparison of control models identified experimentally on the pilot CFB to those obtained by simulation. Favourable comparison leads to the conclusion that the dynamic model is suitable for use in control simulation.

In parallel with the modeling work, advanced control methods were applied to the pilot CFB combustor. This included process identification through open-loop PRBS testing and time series analysis, application of single loop Generalized Predictive Control (GPC) control of combustor temperature, and multivariable control of the combustor to meet heat removal, flue gas oxygen concentration, and combustion temperature specifications. The identification methods tended to provide higher order control models than probably necessary. This was due to operational limitations of the pilot plant leading to the inability to collect sufficient data, and is not expected to be a problem on an industrial scale. Although model order is not minimal, since the controller may well operate under conditions of changing fuel type, where knowledge of the model for the new fuel is lacking, it would be advantageous to employ an over-parameterized model in order to be able to account for changing dead-time and possible inverse responses. Finally, the applicability of GPC within a CFB combustor framework is demonstrated. The complete control structure takes advantage of the differing time constants of the various parameters within the system. The water outlet temperature control setpoint can be realized in the order of a few seconds with solids circulation rate; flue gas oxygen (notwithstanding optimization), can be maintained within  $\pm 1\%$  with a detuned PID controller adjusting the total air flow based on this measurement; and a GPC controller adjusts the rotary valve control signal in order to maintain combustion temperature requirements with fuel feedrate.

## **TABLE OF CONTENTS**

<b>Abstract</b>		<b>ii</b>
<b>Table of Contents</b>		<b>iv</b>
<b>List of Tables</b>		<b>ix</b>
<b>List of Figures</b>		<b>x</b>
<b>Acknowledgements</b>		<b>xviii</b>
<b>Chapter 1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	Background	1
1.2	Fluidization	2
1.3	The Circulating Fluidized Bed	3
1.4	The CFB Boiler	4
1.5	Dynamic Modeling and Control	7
1.6	Outline of Thesis	9
<b>Chapter 2</b>	<b>FAST FLUIDIZATION AND THE CIRCULATING FLUIDIZED BED COMBUSTOR</b>	<b>12</b>
2.1	Introduction	12
2.2	Hydrodynamics of Fast Fluidization	12
2.2.1	Regimes of Fluidization	13
2.2.2	Axial Structure of Fast Fluidization	19
2.2.3	Radial Solids Distribution	23
2.2.4	Gas Flow	25
2.3	Heat Transfer	25
2.4	CFB Boiler	32
2.4.1	Riser	32
2.4.2	Gas/Solids Separation	35

2.4.3	Solids Return Systems	35
2.5	Dynamic Modeling	40
Chapter 3	EXPERIMENTAL FACILITIES	44
3.1	Reactor Shaft	44
3.2	Fuel Feed System	47
3.3	Heat Transfer Surfaces	49
3.4	Solids Recycle Systems	52
3.5	Flue Gas Cooling and Cleaning	54
3.6	Gas Sampling System	54
3.7	Instrumentation, Data Acquisition, and Control	55
3.8	Modifications for Current Project	59
3.9	Fuels and Inerts Characteristics	60
Chapter 4	MODEL DESCRIPTION	63
4.1	Introduction	63
4.2	Mass Balance	64
4.2.1	Solids Distribution	66
4.2.2	Solids Balance Zones	71
4.2.3	Gas Balance Zones	75
4.2.4	Combustion	76
4.3	Heat Balance	80
4.3.1	Heat From Reaction	80
4.3.2	Heat Removed by Heat Exchanger	80
4.3.3	Heat Loss to Refractory	84
4.3.4	Heat Added to Solids Returned to Riser	86
4.4	Model Summary	88
4.5	Simulation	90

Chapter 5	PROCESS IDENTIFICATION AND CONTROL	91
5.1	Introduction	91
5.2	Overview of Control	94
5.3	Generalized Predictive Control	98
5.4	Adaptive Control	105
5.5	Process Identification	108
5.5.1	Plant Testing	110
5.5.2	Data Analysis	110
5.5.2.1	Transfer Function Structure	112
5.5.2.2	Noise Model Structure	117
5.5.2.3	Parameter Fitting	118
5.5.2.4	Diagnostic Checking	119
Chapter 6	RESULTS AND DISCUSSION - CONTROL IMPLEMENTATION	120
6.1	Introduction	120
6.2	Basic Control Loops	121
6.2.1	Air Flow Control	121
6.2.2	Solids Circulation Control	124
6.3	Temperature Control	127
6.3.1	Single Loop Temperature Control	127
6.3.1.1	Run #12	129
6.3.1.2	Run #13	140
6.3.1.3	Run #14	151
6.4	The Multi-loop Control Structure	162
6.4.1	Tuning the PID Loops	171
6.4.2	Model Identification for GPC Control	171
6.4.3	Controller Testing	184
Chapter 7	RESULTS AND DISCUSSION - SIMULATOR	197
7.1	Introduction	197

7.2	Model Validation	198
7.2.1	Transfer Function Comparison	198
7.2.2	Step Testing Comparison	204
7.3	Model Characterization	215
7.3.1	Effect of Char Size	215
7.3.2	Effect of Heat Transfer Area	220
7.3.3	Controls Test Simulation	224
7.4	Summary	233
Chapter 8	SUMMARY AND CONCLUSIONS	234
8.1	Summary	234
8.2	Conclusions	238
8.3	Recommendations for Further Work	239
Nomenclature		241
References		249
Appendix A	KINETIC RATE DETERMINATION	255
A.1	Introduction	255
A.2	Experimental Apparatus	256
A.3	Theory and Analysis	261
A.4	Results	270
Appendix B	SOLIDS CIRCULATION RATE MEASUREMENT	277
B.1	Theory	277
B.2	Experimental Apparatus	278



<b>B.3</b>	<b>Experimental Results</b>	<b>281</b>
<b>Appendix C</b>	<b>PROGRAM LISTING - DATA LOGGING AND CONTROL FOR THE UBC PILOT CFB COMBUSTOR</b>	<b>293</b>
<b>Appendix D</b>	<b>PROGRAM LISTING - DYNAMIC SIMULATOR OF A CIRCULATING FLUIDIZED BED COMBUSTOR</b>	<b>338</b>
<b>Appendix E</b>	<b>PROGRAM LISTING - TIME SERIES ANALYSIS AND TRANSFER FUNCTION FITTING</b>	<b>375</b>

## **LIST OF TABLES**

1.1	Comparison of FBC Boiler Systems (adapted from Thomas et al., 1986)	6
3.1	Key Features of Gas Analysers	57
3.2	Proximate and Ultimate Analyses of Fuels	61
3.3	Screen Size Distribution of Ottawa 50 Silica Sand	62
4.1	Thermal Properties of Materials of Construction	85
A.1	Comparison of Orcutt Bubbling Bed Model with Well-Mixed Assumption	265
A.2	Results of Fuel Reactivity Experiments	276
B.1	Results of Circulation Rate Experiments	283

## **LIST OF FIGURES**

1.1	Modeling for Process Control	8
2.1	Regimes of Fluidization (from Yerushalmi et al., 1978)	14
2.2	Axial Bed Voidage Characterization - profiles of cross-sectional overage bed voidage determined by Li et al. (1981) for a 8 m tall column of diameter 90 mm	16
2.3	Regime of Fast Fluidization according to Karri and Knowlton (1991)	18
2.4	Typical Bed Voidage Profiles in Fast Fluidization (from Li et al., 1981)	20
2.5	Typical Radial Variance in Solids Flux - profiles from a column of 152 mm diameter with alumina of mean particle size of 64 $\mu\text{m}$ by Rhodes et al. (1988)	24
2.6	Illustration of Solids Distribution in the Riser of a CFB (from Senior and Brereton, 1992)	26
2.7	Typical Membrane Wall Geometry from Wu (1990)	27
2.8	Variation of Local Heat Transfer Coefficient with Height (from Mahalingam and Kolar, 1993)	29
2.9	Effect of Abrupt Exit on Vertical Profile of Axial Solids Loading (from Senior and Brereton, 1992)	34
2.10	U-Beam Separator (from Basu and Fraser, 1991)	36
2.11	CFB System Pressure Balance (from Rhodes and Geldart, 1987)	38
3.1	Simplified Schematic of CFBC Facility at UBC	45
3.2	View of Principal Refractory Lined Reactor Column	46
3.3	Primary Air Distributor	48
3.4	Membrane Wall Heat Transfer Surface	50
3.5	Hairpin Heat Transfer Surface Configuration	51

3.6	Eductor Configuration for Secondary Cyclone Return	53
3.7	Gas Sampling System	56
3.8	Thermocouple and Pressure Tap Locations	58
4.1	Modeling Zones for Material Balance in Primary Loop of the CFB	65
4.2	Sketch Showing Cyclone Dimensional Parameters (from Leith and Licht, 1972)	74
5.1	Simplified Block Diagram Showing Internal Model Control (IMC) Structure	96
5.2	Impulse Response Functions of Various Transfer Function Models	116
6.1	Effect of Compressor Fluctuations on Flue Gas Oxygen Concentration with Manual Setting of Air Flowrate (Highvale coal, combustor temperature = 835 °C, gas velocity = 7 m/s, P/S air ratio = 3)	123
6.2	Operation of Solids Loading Controller Showing Its Success in Following Step Changes in Setpoint of Solids Loading in Riser (Highvale coal, combustor temperature = 855 °C, gas velocity = 7.3 m/s, P/S air ratio = 2)	125
6.3	Temperature Response - Run #12 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	132
6.4	Rotary Valve Speed - Run #12 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	135
6.5	CO Response during Run #12 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	137
6.6	Adaptation of B Polynomial - Run #12 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	139
6.7	Temperature Response - Run #13, Test #1 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	142

6.8	Adapted Model Coefficients - Run #13, Test #1 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	143
6.9	Rotary Valve Control Signal - Run #13, Test #1 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	144
6.10	CO Response - Run #13, Test #1 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	146
6.11	Rotary Valve Speed - Run #13, Test#2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	147
6.12	Rotary Valve Speed - Run #13, Test #3 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	148
6.13	CO Response - Run #13, Test #2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	149
6.14	CO Response - Run #13, Test#3 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	150
6.15	Temperature Response - Run #13, Test#2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	152
6.16	Temperature Response - Run # 13, Test #3 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	153
6.17	Temperature Response - Run #14, Test #1 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	154
6.18	Rotary Valve Signal - Run #14, Test #1 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	155

6.19	CO Concentration - Run #14, Test #1 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	156
6.20	Temperature Response - Run #14, Test#2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	158
6.21	Solids Loading - Run #14, Test #2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 100 kg/m <sup>3</sup> )	159
6.22	Rotary Valve Speed - Run #14, Test #2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	160
6.23	Time Variation of Model Coefficients - Run #14, Test #2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	161
6.24	Rotary Valve Signal - Run #14, Test #2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	163
6.25	Temperature Response - Run #14, Test #2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	164
6.26	CO Concentration - Run #14, Test #2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m <sup>3</sup> )	165
6.27	Investigated Control Structure	167
6.28	Schematic of Heat Exchanger Outlet Water Temperature Controller	169
6.29	Schematic of Flue Gas Oxygen Concentration Controller	170
6.30	Schematic of Combustor Temperature Control Loop	172
6.31	Rotary Valve Signal during PRBS Test Run #48 (Syncrude coke)	176
6.32	Temperature during PRBS Test Run #48 (Syncrude coke)	177

6.33	Fuel Feedrate during PRBS Test of Run #48 - Averaged over 16 Minute Intervals (Syncrude coke)	178
6.34	Comparison of Step Response of Models Obtained from Time Series Analysis and the GPC Adaptor - Run #48 (Syncrude coke)	180
6.35	Comparison of Transfer Function Step Responses from PRBS Test (Run #22) and Adapted Variables (Run #40)	181
6.36	Comparison of Step Response of PRBS Test (Run #44) and Adapted Model (Run #45)	183
6.37	Temperature during Run #49 (Syncrude coke)	185
6.38	Rotary Valve Signal during Run #49 (Syncrude coke)	187
6.39	Oxygen during Run #49 (Syncrude coke)	188
6.40	Air Flow during Run #49 (Syncrude coke)	189
6.41	Water Outlet Temperature during Run #49 (Syncrude coke)	190
6.42	Solids Loading during Run #49 (Syncrude coke)	191
6.43	A Polynomial Coefficients during Run #49 (Syncrude coke)	195
6.44	B polynomial Coefficients during Run #49 (Syncrude coke)	196
7.1	Possible Models for Process Sampled Discretely	199
7.2	Fuel Feedrate for Simulated PRBS Test with Highvale Coal	200
7.3	Temperature Response of PRBS Test Simulation with Highvale Coal	201
7.4	Comparison of Step Responses of Discrete Time Models for Highvale Coal	202
7.5	Comparison of Step Responses of Discrete Time Models for Conoco Coke	203
7.6	Comparison of Rotary Valve Signal with 10-Minute-Averaged Fuel Feedrate (Highvale coal, Run #52)	206
7.7	Transient Comparison of Temperature for Highvale Coal (Run #52)	207

7.8	Transient Comparison of Flue Gas Oxygen Concentration for Highvale Coal (Run #52)	208
7.9	Transient Comparison of Heat Exchanger Water Outlet Temperature for Highvale Coal (Run #52)	210
7.10	Step Test Comparison of Temperature for Conoco Coke (Run #51)	212
7.11	Step Test Comparison of Flue Gas Oxygen Concentration for Conoco Coke (Run #51)	213
7.12	Step Test Comparison of Heat Exchanger Outlet Water Temperature for Conoco Coke (Run #51)	214
7.13	Predicted Effect of Particle Size on Temperature Response (Conoco coke, solids loading = 120 kg/m <sup>3</sup> , total air feedrate = 205 kg/h, P/S ratio = 2)	217
7.14	Effect of Fuel Particle Size on Bed Carbon Inventory (Conoco coke, solids loading = 120 kg/m <sup>3</sup> , total air feedrate = 205 kg/h, P/S ratio = 2)	219
7.15	Effect of Particle Size on Oxygen Concentration - Conoco Coke (Conoco coke, solids loading = 120 kg/m <sup>3</sup> , total air feedrate = 205 kg/h, P/S ratio = 2)	221
7.16	Effect of Bed Inventory on Temperature in Large Scale Unit (Conoco coke, solids loading = 120 kg/m <sup>3</sup> , total air feedrate = 205 kg/h, P/S ratio = 2, char diameter = 900 µm)	223
7.17	Temperature during Simulated Controls Test (Conoco coke, P/S ratio = 2, char diameter = 900 µm)	225
7.18	Fuel Feedrate during Simulated Control Test (Conoco coke, P/S ratio = 2, char diameter = 900 µm)	226
7.19	Oxygen during Simulated Control Test (Conoco coke, P/S ratio = 2, char diameter = 900 µm)	227
7.20	Total Air Flow during Simulated Control Test (Conoco coke, P/S ratio = 2, char diameter = 900 µm)	229
7.21	Water Temperature during Control Test Simulation (Conoco coke, P/S ratio = 2, char diameter = 900 µm)	230



7.22	Average Solids Loading during Simulated Controls Test (Conoco coke, P/S ratio = 2, char diameter = 900 $\mu\text{m}$ )	231
7.23	Coefficients during Simulated Controls Test (Conoco coke, P/S ratio = 2, char diameter = 900 $\mu\text{m}$ )	232
A.1	Schematic Diagram of the 50 mm Diameter Fluidized Bed Reactor	257
A.2	External View of the Bench Scale Fluidized Bed Reactor	258
A.3	Details of the Bench Scale Reactor Setup	259
A.4	Typical Carbon Dioxide Response Curve during Char Burnout for Highvale Coal Char Particles of Initial Diameter of 1.6 mm at a Reactor Temperature of 810 $^{\circ}\text{C}$	271
A.5	Effect of Reaction Order, $n$ , on Calculated Kinetic Rate Constant, Highvale Coal Char with Initial Particle Diameter of 1.6 mm, Reactor Temperature = 810 $^{\circ}\text{C}$ , $A = 6 \text{ (kmol/m}^2\cdot\text{s}\cdot\text{(kmol/m}^3)^n\text{)}$ , $E_a = 1 \times 10^7 \text{ (kJ/kmol)}$	272
A.6	Effect of Temperature on Kinetic Rate Constant, Highvale Coal Char with Initial Particle Diameter of 1.6 mm, $n = 0.9$ , $A = 6 \text{ (kmol/m}^2\cdot\text{s}\cdot\text{(kmol/m}^3)^n\text{)}$ , $E_a = 1 \times 10^7 \text{ (kJ/kmol)}$	273
A.7	Confirmation of Determined Kinetic Parameters, Highvale Coal Char with Initial Particle Diameter of 1.6 mm, Reaction Temperature = 850 $^{\circ}\text{C}$ , $n = 0.9$ , $A = 1092 \text{ (kmol/m}^2\cdot\text{s}\cdot\text{(kmol/m}^3)^n\text{)}$ , $E_a = 8.0733 \times 10^7 \text{ (kJ/kmol)}$	275
B.1	Schematic of Line-and-Sinker System	280
B.2	Effect of Riser Solids Loading on Net Solids Flux for a Gas Velocity of 8 m/s (Ottawa 50 sand, total inventory = 100 kg, combustor temperature = 850 $^{\circ}\text{C}$ )	284
B.3	Effect of Riser Solids Loading on Net Solids Flux for a Gas Velocity of 7.3 m/s (Ottawa 50 sand, total inventory = 100 kg, combustor temperature = 850 $^{\circ}\text{C}$ )	285
B.4	Effect of Riser Solids Loading on Net Solids Flux for a Gas Velocity of 6.7 m/s (Ottawa 50 sand, total inventory = 100 kg, combustor temperature = 850 $^{\circ}\text{C}$ )	286

B.5	Effect of P/S Air Ratio on Net Solids Flux for a Riser Solids Loading of 90 kg/m <sup>3</sup> (Ottawa 50 sand, total inventory = 100 kg, combustor temperature = 850 °C)	287
B.6	Effect of P/S Air Ratio on Net Solids Flux for a Riser Solids Loading of 120 kg/m <sup>3</sup> (Ottawa 50 sand, total inventory = 100 kg, combustor temperature = 850 °C)	288
B.7	Effect of P/S Air Ratio on Net Solids Flux for a Riser Solids Loading of 150 kg/m <sup>3</sup> (Ottawa 50 sand, total inventory = 100 kg, combustor temperature = 850 °C)	289
B.8	Effect of Gas Velocity on Net Solids Flux for a Riser Solids Loading of 90 kg/m <sup>3</sup> (Ottawa 50 sand, total inventory = 100 kg, combustor temperature = 850 °C)	290
B.9	Effect of Gas Velocity on Net Solids Flux for a Riser Solids Loading of 120 kg/m <sup>3</sup> (Ottawa 50 sand, total inventory = 100 kg, combustor temperature = 850 °C)	291
B.10	Effect of Gas Velocity on Net Solids Flux for a Riser Solids Loading of 150 kg/m <sup>3</sup> (Ottawa 50 sand, total inventory = 100 kg, combustor temperature = 850 °C)	292

## **ACKNOWLEDGEMENTS**

I would like to express my sincere appreciation to a number of people whose efforts contributed greatly to the completion of this thesis.

To my supervisors, Dr. Clive Brereton, Dr. John R. Grace, and Dr. C. Jim Lim, for their ideas, discussion, and support provided throughout this project in addition to their guidance and encouragement both towards this work as well as in personal areas beyond the scope of this project.

The operation and maintenance of the UBC pilot CFB combustor involved the collective effort of a large number of people over the years. In particular the contributions of Hugh Ackroyd, Dr. Robert Legros, Dr. Richard Senior, Dr. Jiansheng Zhao, Dr. Adam Luckos, Dr. Serge Julien, Irene Hwang, Polly Wong, Wenqi Luan, Fenglei Liu, Dr. Seng Lim, and Lawrence Sung are greatly appreciated.

Financial support from the Natural Sciences and Engineering Research Council, the U.B.C. Graduate Fellowship Fund, and Energy, Mines and Resources Canada is gratefully acknowledged.

Most of all, I would like to express my gratitude and thanks to my parents for their continuous support and encouragement.

## **CHAPTER 1 - INTRODUCTION**

### **1.1 Background**

The purpose of the work reported in this thesis was to investigate the dynamics of the Circulating Fluidized Bed (CFB) combustion process. This objective has been achieved by consideration of the problem both in terms of process control and modeling. Advanced control methods have been applied to the University of British Columbia (UBC) pilot scale CFB combustor. These have included various dynamic identification methods as well as implementation of the Generalized Predictive Control (GPC) algorithm for prediction and control. A dynamic simulator is described which was developed specifically for application to control structure design. Most crucial to this simulator is a dynamic model providing responses of key variables for circulating fluidized bed combustion.

Although the concept of fast fluidization was first identified over 50 years ago, it is only recently that industry has recognized the viability and the substantial advantages of this fluidization regime. This commercial interest has sparked a flood of researchers into the area of fast fluidization and CFB technology. An inadequate level of understanding exists because

of the complexity and relative newness of the area. The deluge of research activity has led to abundant controversy.

The field can be characterized as having a fast-growing and continually expanding knowledge base. Research includes investigation of micro/macro scale phenomena, as well as gross operational characterization. Although incomplete, the current level of understanding is adequate for extensions to the time domain. These extensions may be found wanting due to ignorance of some basic phenomena, but are timely as a beginning to the understanding of the dynamics of industrial CFB process equipment. As more of the basics of CFBs are resolved, the time dependencies of these systems will become clearer.

## **1.2 Fluidization**

Fluidized beds are used in both physical and chemical processes involving solid particles. Gas/solid reactions are of prime importance in applications such as those in the metallurgical and chemical industries, the processing of stones and ceramics, and in fuel and energy engineering (Weiss et al., 1988). Such reactions occur at the surface or within the porous structure of individual particles and involve a complicated interplay of heat transfer, mass transfer, and reaction kinetics.

Both heat and mass transfer involve a number of steps. For endothermic reactions, heat is supplied to the gas, transferred by convection to the surface of the solid, and then transferred to the interior by conduction. Gaseous reactants are contacted with the particle by a similar mechanism of diffusion to the particle surface and pore diffusion into the particle. In cases where the kinetics of the gas/solid reaction can be considered fast (e.g.,  $t_{1/2} < 100$  ms), heat and mass transfer often become the rate controlling steps (Behie, 1987). For such reactions, conversion is a function of the rate at which the gaseous components can be supplied to and removed from the particle and the uniformity of temperature. A higher particle specific surface results in reduced internal heat and mass transfer resistances. Externally, high relative velocities between gas and solids reduce the boundary layer

surrounding the particle, giving lower external heat and mass transfer resistances. For the above reasons, over the past few decades there has been a trend toward using finer-grained solids and higher gas velocities in industrial practice. This trend culminated in the development of the Circulating Fluidized Bed.

### **1.3 The Circulating Fluidized Bed**

High gas velocities and small particles lead to conditions of fast fluidization (Chapter 2). Characterized by a high carryover of solids, discussion of this hydrodynamic regime cannot be limited to the reaction zone. Key to operation under the regime of fast fluidization is the separation of gas and solids and the continuous return of solids to the reaction zone. The Circulating Fluidized Bed (CFB) is a technology developed to operate under these conditions. It consists of a gas/solids contacting zone (riser), some form of gas/solids separation device (typically one or more cyclones), and a device for returning the solids to the base of the riser (commonly a non-mechanical valve). These components are addressed in detail in Chapter 2.

Over the past 15 years, CFB technology has received rapidly growing attention. Industry quickly recognized the broad potential of high velocity fluidization. Among its unique advantages are greater gas/solids contacting efficiency, short solids residence time (of the order of seconds), and independent control of gas and solids retention times. Applications of fast fluidization technology for chemical reactions include combustion for power generation and waste incineration, pyrolysis, calcining, and catalytic processes. Currently, one of the more significant applications of this technology has been in the form of CFB boilers. These boilers provide many advantages over conventional fluidized bed steam and electrical power generation processes.

#### **1.4 The CFB Boiler**

The use of fluidized bed combustion (FBC) for industrial steam and hot water boilers began in the 1960's and has grown rapidly since. Advantages such as the ability to cope with varying fuel quality, and control of sulphur and nitrogen oxides, have encouraged the development of this technology (Thomas et al., 1986). Atmospheric pressure FBC boiler systems may be divided into three general categories: shallow bubbling bed boilers, deep bubbling bed boilers, and circulating fluidized bed boilers.

Shallow bubbling beds employ bed depths of 15 to 30 cm and fluidization velocities of 2.0 to 3.0 m/s. These beds are very sensitive to oversize particles which tend to cause de-fluidization. Oversized particles may arise either through introduction with the solids feed stream or by formation within the bed due to agglomeration.

Deep bubbling beds are characterized by typical depths of 70 to 110 cm and fluidizing velocities of 1.5 to 2.5 m/s. These beds are more suited for higher ash coals (> 10% ash) as bed material removal systems can be employed.

Circulating fluidized bed boilers operate in the regime of fast fluidization (Chapter 2). Typical gas velocities range from 4.5 to 9 m/s leading to a continuous gradation in suspension density, with the depth of a lower dense zone typically from 0.4 to 1.5 m. The CFB combustor particulate material generally consists of sand, ash from the fuel, combusting fuel or char particles, and limestone for SO<sub>2</sub> capture. These solids circulate through the system in a well mixed state, providing high combustion efficiency, good SO<sub>2</sub> removal, and favorable rates of heat transfer. Large, unentrainable stones may be added to the bed to enhance the dense zone in the combustor (Stockdale and Stonebridge, 1985).

Ash agglomeration is virtually non-existent in CFB combustors. The high degree of mixing and recycle prevent hot spots from occurring in the reactor which could cause agglomeration. In addition, the vigorous motion gives little opportunity for bonds to form between particles. The ability to handle agglomerating fuels is commonly enhanced by using a diverging conical lower section.

Table 1.1, reproduced from Thomas et al. (1986), compares FBC boiler systems. Shallow bubbling beds are often preferred because of the simplicity of their design and their small size. The low height requirement allows for their retro-fitting of existing non-FBC facilities as FBC boilers. Deep bubbling beds allow both oil and gas firing and some tolerance in solids fuel variations. Strict emission control is possible with FBC systems, with limestone used to capture SO<sub>2</sub> and low temperature/staged combustion giving considerable reductions in NO<sub>x</sub> emissions relative to conventional combustion/boiler systems.

As can be seen from Table 1.1, CFB combustion (CFBC) systems provide versatile and favourable combustion characteristics. Their only disadvantage is the requirement of a tall combustion chamber with large combustion zone volumes, and large cyclone solids collection systems. These size requirements make the CFB combustor capital-intensive, but superior capabilities often make them most cost-effective, especially for large-scale utility boilers.

FBC systems burning coal typically operate at combustion temperatures below 900 °C. At these temperatures very little thermal NO<sub>x</sub> is formed. (NO<sub>x</sub> emissions become a problem in poorly designed fluidized beds or with those beds operated at other than design conditions where hot-spots can develop, increasing temperatures to levels leading to elevated NO<sub>x</sub> formation). CFBC systems commonly introduce secondary air at several points along the riser to reduce NO<sub>x</sub> even further. This method of combustion promotes reduction of NO<sub>x</sub> by carbon and carbon monoxide, e.g. as shown by Thomas et al. (1986):



Also, and more importantly, staging prevents formation by reducing the partial pressure of oxygen in the primary zone.



**Table 1.1: Comparison of FBC Boiler Systems**

(adapted from Thomas et al., 1986)

Category	Shallow Bubbling Beds	Deep Bubbling Beds	Circulating Fluidized Beds
boiler design or size	simple design	overall boiler size is reduced compared to conventional firing due to enhanced heat transfer	overall boiler size is often relatively large and can be a disadvantage
	compact combustor, particularly compatible with shell boilers		
fuels and combustion	gas-firing capabilities, oil to a lesser extent	oil and gas firing capabilities	oil and gas firing capabilities
	coal grade tolerance superior to stoker boilers	proper designs can utilize solid fuels which are incompatible with conventional firing	proper designs can burn almost any combustible solid or liquid, including wastes
	some waste-burning abilities	fuel flexibility during operation	broad fuel flexibility during operation
	adequate combustion efficiency	broad waste combustion possibilities	superior combustion efficiency
		good combustion efficiency	
emissions	modest SO <sub>2</sub> removal is possible	SO <sub>2</sub> removal is easily achieved using limestone for bed material	superior SO <sub>2</sub> removal is obtained with low sorbent use
	NO <sub>x</sub> may be limited by temperature control and other techniques	NO <sub>x</sub> can be controlled very well with staged combustion	superior NO <sub>x</sub> control with staged combustion

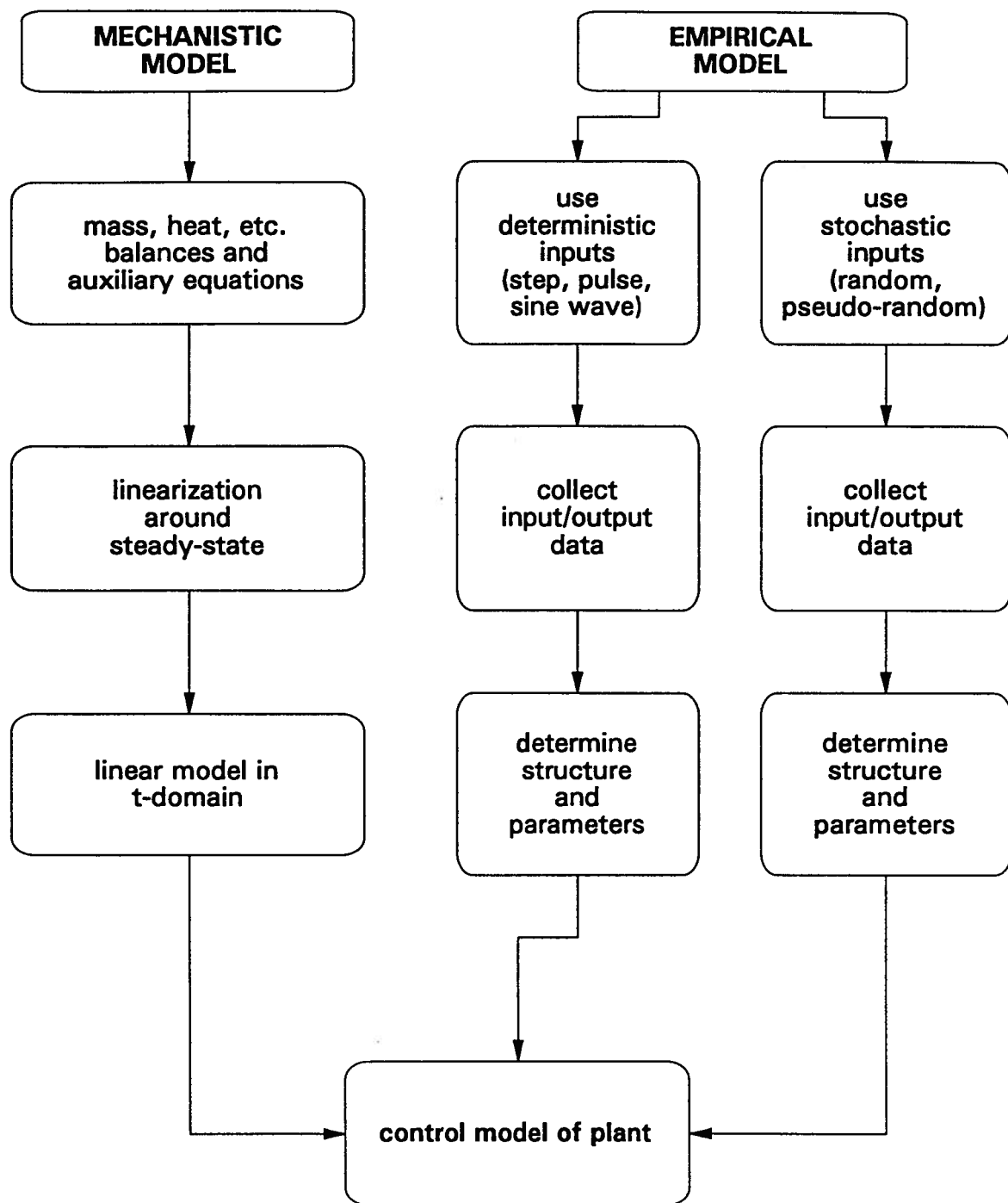
Ca/S molar ratios used in conventional CFB combustors range from 1.5 to 3 resulting in 90 to 95% sulphur capture. For the same ratios, sulphur capture tends to be somewhat lower (as low as 60%) in conventional fluidized beds where larger limestone particles are used.

Most CFB boilers are capable of turndown to 33 to 25% of full load, with load following in the range of 4 to 10%/min. (Sahagian, 1985 and Kullendorff et al., 1986). More complex CFBC systems with incorporation of external heat exchangers can allow turndown to as low as 13%, with load following as high as 20%/min. (Beisswenger et al., 1985; Stockdale and Stonebridge, 1985).

### **1.5 Dynamic Modeling and Control**

To date there is only limited understanding of the dynamic behaviour of CFBs. This has led to extreme difficulties in the development of reliable design methods and control models. What is needed is a model which can describe both the internal riser and external loop of the CFB, predicting both steady-state behaviour and dynamic responses. Steady-state analysis is required for the design of a CFB around a given operating condition. Dynamic response is necessary for both physical design and, more importantly, for design and implementation of a reliable control structure. Such a model could be used not only for simulation of a control structure, but also in controllers as a predictor.

For control purposes, a model can be developed based on either a mechanistic or an empirical approach. The general development paths for each are presented in Figure 1.1. In either case, the ultimate form of the control model is a z-domain or discrete t-domain representation of the plant. This is a common form used for process control as, at this time, supervisory and other new generation controllers are predominantly digital computer based. The plant is monitored and calculations are carried out on discrete time intervals rather than continuously, as with their simpler analog counterparts.



**Figure 1.1: Modeling for Process Control**

A mechanistic model is required for macroscopic understanding of the combustor's operation. Ideally such a model is developed directly through mass and heat balances and other theoretical equations. In many instances, empirical correlations need to be incorporated to complete the model. Although often incorporating a degree of empiricism, this type of model is much more valuable than a truly empirical model because it enables design and simulation, as well as control, whereas the latter is only useful for control purposes around a limited range of tested operating conditions. Empirical models, on the other hand, may be found to be more reliable in situations where the process is complicated or poorly understood. In this case the model is developed directly from plant data driven by deterministic (step, pulse, sine function, etc.) or stochastic (random or pseudo-random) disturbances.

The most important benefit of simulation in control development is the opportunity it affords to investigate not only control parameters, but also a large number of control schemes. Simulators offer a particular advantage in upgrading current controls, and in developing and optimizing new units, in that they enable this to be done off-line. Once on-line, although limited dynamic testing is possible, complete testing of a particular control system is difficult due to demands on the performance of the unit. The ability to perform these tests off-line avoids disturbances to the plant and enhances the opportunity for controls upgrading and design. In addition, simulators are invaluable for training those inexperienced with the operation of a particular unit.

## **1.6 Outline of Thesis**

Although the CFB has been in industrial use for a number of years, it is not a technology that is well understood. In particular, to date there has only been limited investigation of the transient behaviour of these units. Industrial acceptance of CFBC technology is justification enough to merit research in this area. Though already economically viable for industrial application, understanding of the dynamic behaviour is desirable for more efficient combustion, resulting in lower fuel and operating costs and conservation of energy

resources. Knowledge of the dynamic behaviour should permit more advanced control structures to be developed. Ultimately, this would lead to improved operation around optimum conditions, resulting in lower fuel/air consumption and even stricter control of emissions. "To develop advanced control systems of CFB boilers, dynamic mathematic models describing the transient behaviour of the CFBC are required" (Mori et al., 1991).

This thesis presents the results of research in two related areas - dynamic modeling and control. A dynamic model is developed which focusses on the prediction of the key variables for control. The model is then incorporated into a simulator for controls investigation. Developmental methods of advanced control and a demonstration of application to CFB combustion are included.

Chapter 2 provides a brief overview of fast fluidization with respect to hydrodynamics and heat transfer. The application of CFBs to boiler technology is described, focusing on various configurations currently used industrially. A survey of the current state of dynamic modeling is also presented.

The main experimental set-up used in this work, the UBC pilot scale CFB combustor is described in Chapter 3. The equipment description is relatively brief as detailed reports are readily available elsewhere. This chapter focuses on the modifications required for the present work.

Chapter 4 describes the dynamic simulator developed for the study of control of CFB combustion. Assumptions necessary for modeling, and those required to simplify the solution for this application, are specified and justified.

Chapter 5 is a discussion of relevant aspects of process control. The theory behind Generalized Predictive Control (GPC) and methods of process identification are presented. GPC has been described as a general control algorithm providing stability and robustness. The reasons for its choice as the control method in this work are discussed in this chapter. Identification methods are presented, and the theory underlying the methods employed in this research is introduced.

Chapter 6 includes the experimental implementation of minor control loops as well as identification for, and implementation of, the GPC algorithm for combustor temperature control of the UBC pilot scale CFB combustor. The temperature controller is demonstrated both in a single loop application as well as within the framework of a control structure for complete combustion control.

Chapter 7 provides a comparison/verification of the simulator with experimental data. An analysis of the proposed model is included. This was an investigation of the effect of char size, total bed inventory, and heat loss on the dynamic response of the model. In addition, the combustion control structure implemented in Chapter 6 is demonstrated on the simulator.

Chapter 8 concludes the thesis with a summary of findings. Recommendations are provided for further investigations into the dynamics of CFB combustion both from a general modeling point of view and for process control.

## **CHAPTER 2 - FAST FLUIDIZATION AND THE CIRCULATING FLUIDIZED BED COMBUSTOR**

### **2.1 Introduction**

This chapter provides a brief overview of key aspects of fluidization, focusing on the regime of fast fluidization utilized in Circulating Fluidized Beds. Fast bed hydrodynamics as well as heat transfer mechanisms are introduced. References are provided to a number of recent reviews for more in-depth study. The CFB is discussed with regard to boiler applications, particularly concentrating on the physical aspects and variants of units employed in industry. Finally, the current state of transient modeling of the CFB combustor is surveyed.

### **2.2 Hydrodynamics of Fast Fluidization**

The hydrodynamic behavior of the fast fluidized bed has been the focus of much research in recent years. Since the recognition of the significant advantages of this regime (circa. 1970), researchers have been endeavoring to keep up with the extensive applications in industry through fundamental understanding and modeling for scale-up, design, and process optimization.

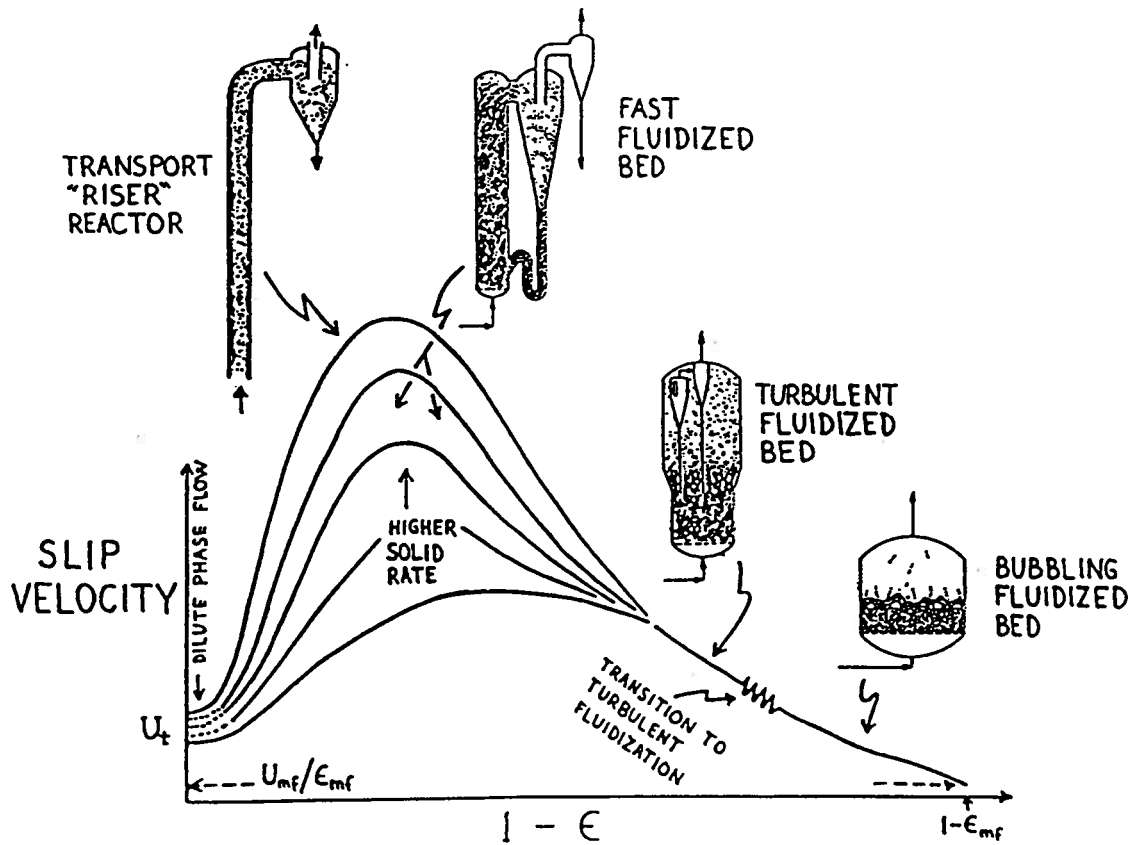
Recent reviews (Grace, 1990; Horio, 1991; Basu and Fraser, 1991) describe research on the hydrodynamics and highlight major controversies. In view of this, only a brief definition and overview of the behaviour of solids and gases in fast fluidized beds are presented here to provide the reader with a basic understanding of this complex regime.

### **2.2.1 Regimes of Fluidization**

As the velocity of a gas passing through a bed of solids is increased, the bed traverses a number of characteristic operating modes or regimes. Figure 2.1 (Yerushalmi et al., 1978) shows the commonly accepted regimes of gas/solid contacting. The transition between each successive regime is directly related to the properties of the solids and gas. At the point of minimum fluidization, a packed bed expands slightly so that particles move about freely with frequent collisions. An increase in gas velocity results in bubbling fluidization, characterized by the coexistence of two phases - a bubble phase (essentially no solids) and the surrounding dense (or emulsion) phase. It is commonly assumed that the emulsion phase is in a state of minimum fluidization and that any excess gas supplied to the bed above that required to achieve minimum fluidization passes through the bed as bubbles. If the column diameter is relatively small, the bubbles may grow to be similar in diameter, resulting in slug flow behaviour.

A further increase in gas velocity results in a more pronounced two-phase character which, with increasing velocity, gradually diminishes to a condition of increasing uniformity in which any voids present are transient and of small scale. The bed is then referred to as being in a turbulent state. A definite, though more diffuse, upper bed surface exists, together with a more dilute dispersion or suspension in the freeboard. The turbulent regime extends up to velocities approaching the transport velocity. As the transport velocity is approached, there is a sharp increase in particle carryover. Beyond the transport velocity, a continuous feed of particles to the bottom of the column is required to maintain a relatively dense suspension. This is the transport regime.



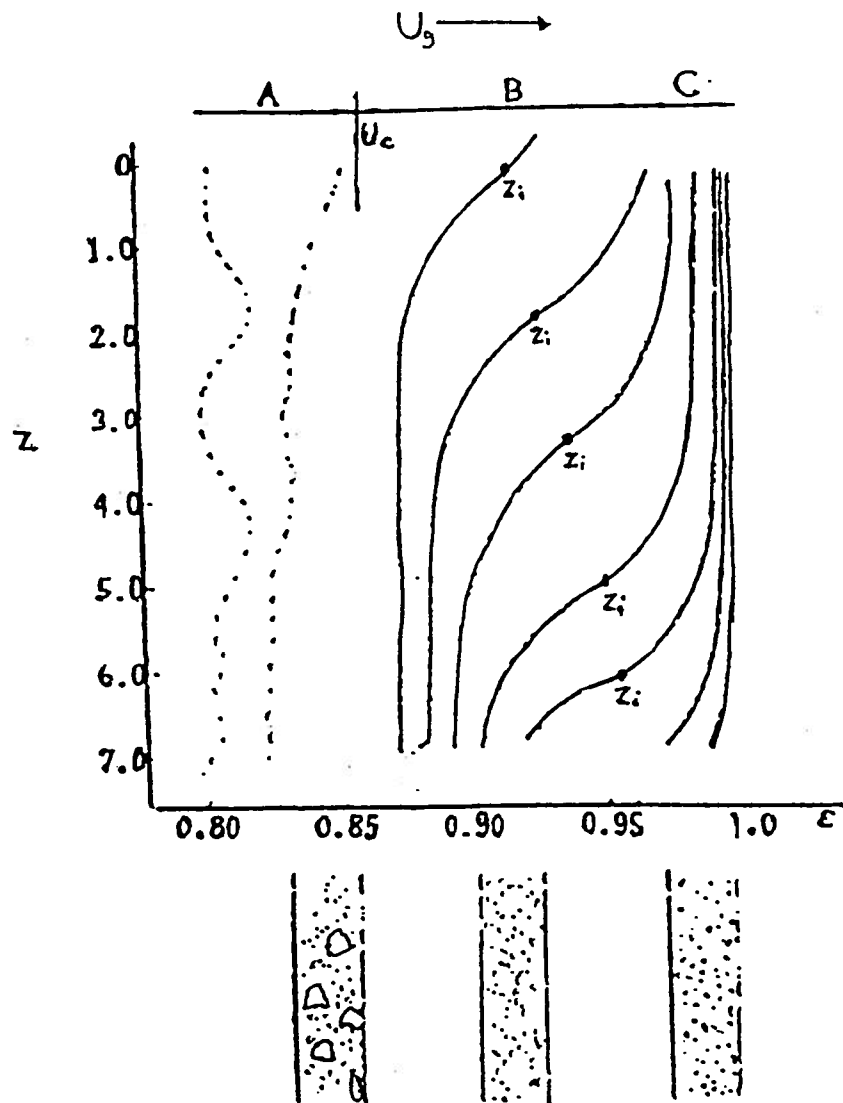


**Figure 2.1: Regimes of Fluidization**  
(from Yerushalmi et al., 1978)

Operation in the transport regime offers a striking advantage over that of the others described above. Above the transport velocity, solids are fully entrained through the column and the density of the suspension depends not only on the gas velocity but also on the rate at which solids are fed to the base of the column. It is this added ability to control the solids concentration that makes this regime unique. At low solids rates, the particles are carried upwards in straight paths with slip velocities close to the terminal velocity of individual particles. This has been termed dilute phase flow. With increasing solids feed rates, the flow through the column becomes increasingly dense until the condition of fast fluidization is reached, characterized by high solids concentration, flow of solids in clusters and strands (resulting in slip velocities an order of magnitude greater than the terminal velocity of individual particles), and extensive backmixing of solids (Yerushalmi, 1986).

"High velocity fluidization" usually refers to both the turbulent and fast fluidization regimes. Li et al. (1981) carried out a study of high velocity fluidization in a 90 mm ID by 8 m tall column. Their investigation incorporated a number of different solids all of Geldart type A classification. They identified three separate regimes of fluidization based on the shape of the bed porosity profile. Their results are presented in Figure 2.2. Region A is indicative of operations in which the gas velocity is below that required to achieve fast fluidization and is most likely representative of a turbulent fluidized bed. Region B shows two distinct phases coexisting within the riser, a dense lower region and a comparatively lean upper region. Region C was encountered at very high gas velocities and characteristically showed a constant bed voidage over the entire height of the column, except in the vicinity of the solids inlet. This is the realm of pneumatic transport (or dilute phase flow).

For many years, "turbulent fluidization" was a term used to qualitatively describe fluidization past the bubbling/slugging regime (essentially high velocity fluidization). The current conceptualization of turbulent fluidization was defined by Kehoe and Davidson (1971). Lanneau (1960) provided evidence that this regime offered opportunities surpassing that of the conventional bubbling bed. Carrying out experiments on fine powders with gas



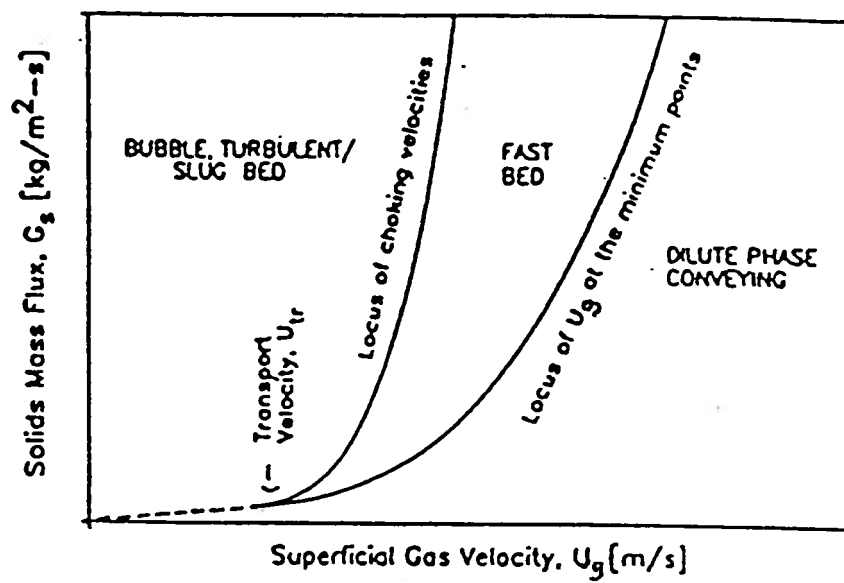
**Figure 2.2: Axial Bed Voidage Characterization**  
 profiles of cross-sectional average bed voidage determined by Li et al. (1981) for a 8 m tall  
 column of diameter 90 mm

velocities up to 1.5 m/s, he noted that with increasing gas velocity, the two-phase nature of the bubbling bed transforms to that of increasing homogeneity leading him to conclude that gas/solid contacting is better at higher velocities. In fact, Laneau's work demonstrates that operation in the bubbling bed regime is probably the worst condition in terms of the efficiency of gas/solid contacting.

Kehoe and Davidson (1971) viewed the transition to the turbulent regime and described the regime as a state of continuous coalescence. They found that this transition is quite gradual rather than sharp. On the other hand, Brereton (1987) found that the transition to the turbulent state involved slugging and dilute phase clustering occurring intermittently, implying that the turbulent regime is itself a transition from slug flow to fast fluidization (Grace and Tuot, 1979).

The voidage profile represented by region B in Figure 2.2 has been considered to correspond to fast fluidization, with the inflection point ( $z_i$ ) separating a dense phase at the bottom and a dilute phase at the top of the riser used as the indicator of the fast fluidization regime. Horio and Morishita (1988) concluded, based on their investigation of upflow systems, that any dense bed experiencing entrainment adjusts the upper boundary of the dense region to changes in solids circulation rates so that the rate of elutriation equals the solids flow rate. Horio (1991) states that "the coexistence of dense and dilute regions could be a rather trivial aspect of fast fluidization phenomena".

Karri and Knowlton (1991) define the fast fluidization regime as bounded by two transition velocities for a given mass flux. These velocities are described in Figure 2.3. For a given mass flux, the lower velocity is defined by the choking velocity (velocity at which the solids saturate the gas). The upper bound on velocity was concluded to be that where the frictional resistance to gas flow through the column begins to dominate over the pressure drop imposed by the presence of entrained solids.



**Figure 2.3: Regime of Fast Fluidization**  
according to Karri and Knowlton (1991)

The solids distribution in fast fluidization is by no means uniform. Particles move in all directions throughout the column. For the most part they can be found in suspended clusters and wall streamers.

Streamers are seen as a number of particles collected together and behaving as a group for a period of time. They appear as long strands or strings of particles at the wall of the riser, moving either upward or, more usually, downward. Streamer motion and breakup involve stochastic processes. Wall streamers intermittently pause, exposing their leading edge to the dilute suspension. This leading edge grows rapidly and protrudes into the core where particles are stripped away at a high rate (Senior and Brereton, 1992). These streamers, being adjacent to the wall, have a marked effect on the heat transfer process to surfaces as discussed below.

### **2.2.2 Axial Structure of Fast Fluidization**

Typically the axial solids distribution in fast fluidization is considered to be sigmoidal as indicated by curve 2 in Figure 2.4. As this figure illustrates, there is a dense zone at the base of the riser, a relatively dilute region at the top, and a gradual transition inbetween, passing through a point of inflection. This point of inflection is usually taken to represent the top of the dense region. The location of this point is a function of the gas velocity, solids circulation rate, and the system pressure balance (or solids inventory). Although a sigmoidal profile is not seen in all units or at all times, it is the most common. Discrepancies can often be attributed to geometrical effects, addressed below.

Li and Kwauk (1980) developed a first order differential equation to describe the vertical voidage profile in fast fluidization. A 90 mm ID by 8 m tall column was employed in which pressure profiles were measured for various gas velocities and solids circulation rates. The solids used in the experiments were all of Geldart type A classification. The results (Li et al, 1981) are generalized into three curves typical of fast fluidization, reproduced in Figure 2.4. Curve 2 shows the complete profile with the limiting cases of curve 1 whose dense bed

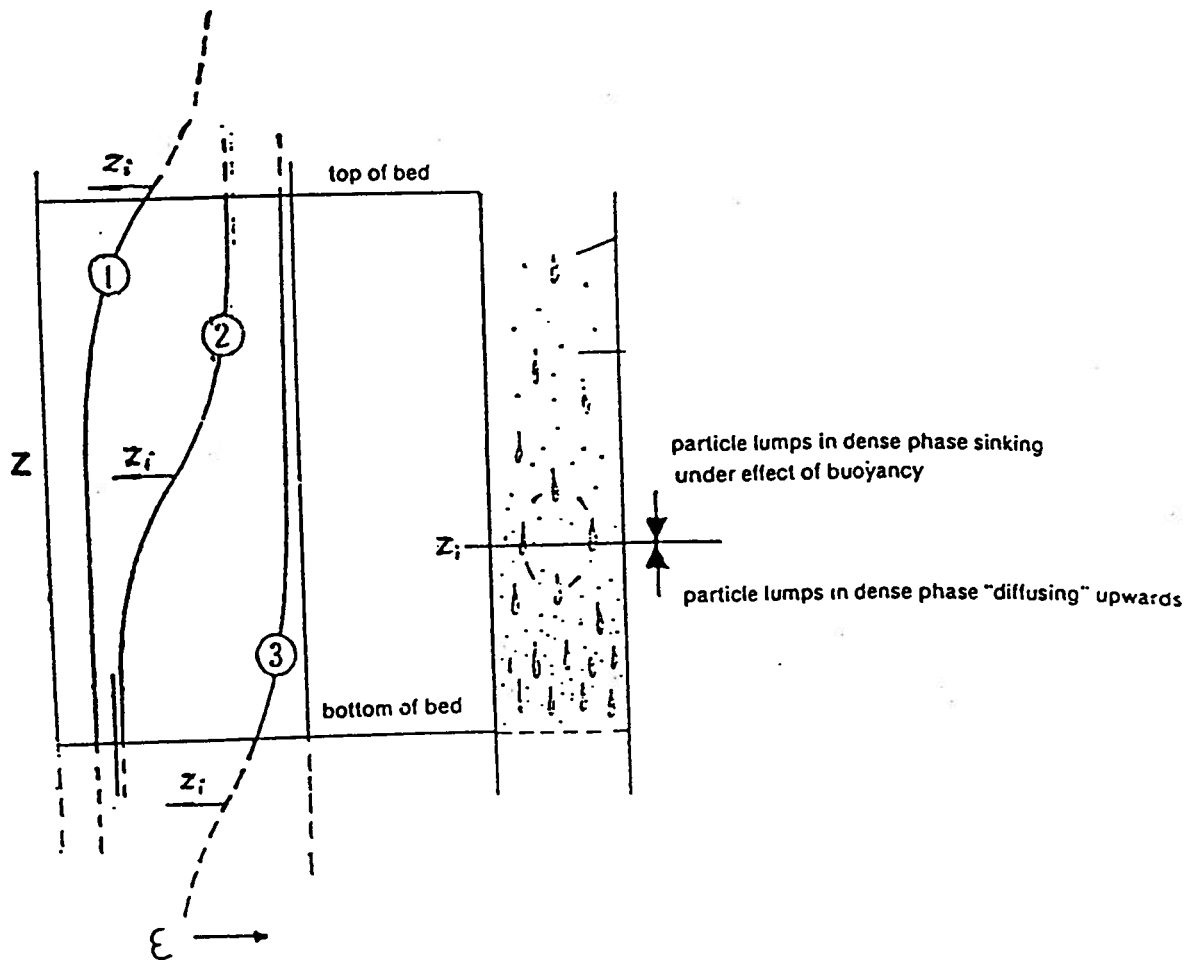


Figure 2.4: Typical Bed Voidage Profiles in Fast Fluidization (from Li et al., 1981)

upper level would require a taller riser and curve 3 which is dilute throughout the entire lengths of the riser. Observations show that clusters of particles migrate about in the bed. Clusters were assumed to rise by a diffusion-type mechanism from the relatively dense lower section and to descend by gravity from the less dense upper region. Li and Kwauk (1980) proposed that at a given height,  $z$ , the upward and downward fluxes of the clusters are equal, leading to:

$$\xi \frac{d}{dz} [\rho_s f_v (1 - \varepsilon_a)] = \omega [\Delta \rho (1 - \varepsilon_a) - \Delta \rho (1 - \varepsilon)] f_v (1 - \varepsilon_a) \quad (2.1)$$

where:  $f_v$  = volume fraction of particle clusters in the fast-fluidized bed  
 $z$  = vertical height (m)  
 $\varepsilon$  = bed voidage  
 $\varepsilon_a$  = voidage of particle clusters  
 $\rho_g$  = gas density (kg/m<sup>3</sup>)  
 $\rho_s$  = apparent density of particles (kg/m<sup>3</sup>)  
 $\Delta \rho$  = effective density ( $\rho_s - \rho_g$ ) (kg/m<sup>3</sup>)  
 $\omega$  = segregation coefficient (a proportionality constant)  
 $\xi$  = diffusive coefficient (a proportionality constant)

At  $z$ , the mean concentration is given by:

$$(1 - \varepsilon) = f_v (1 - \varepsilon_a) + (1 - f_v) (1 - \varepsilon') \quad (2.2)$$

where:  $\varepsilon'$  = voidage of dilute phase

so that:

$$f_v = \frac{\varepsilon' - \varepsilon}{\varepsilon' - \varepsilon_a} \quad (2.3)$$



After substitution, Equation (2.1) becomes:

$$-\frac{d\varepsilon}{(\varepsilon' - \varepsilon)(\varepsilon - \varepsilon_a)} = \left( \frac{\omega \Delta \rho}{\xi \rho_s} \right) dz \quad (2.4)$$

The inflection point on the  $\varepsilon$  vs.  $z$  profile (denoted by  $\varepsilon_i$ ,  $z_i$ ) is found by setting the second derivative of  $\varepsilon$  equal to zero resulting in:

$$\varepsilon_i = \frac{\varepsilon' + \varepsilon_a}{2} \quad (2.5)$$

showing that the point of inflection occurs at the arithmetic mean of the two boundary voidages.

Integration of Equation (2.4) yields:

$$\ln \left( \frac{\varepsilon - \varepsilon_a}{\varepsilon' - \varepsilon} \right) = -\frac{1}{Z_o} (z - z_i) \quad (2.6)$$

where:

$$Z_o = \left( \frac{\xi \rho_s}{\omega \Delta \rho} \right) \frac{1}{(\varepsilon' - \varepsilon_a)} \quad (2.7)$$

Equation (2.6) describes the vertical voidage distribution in a fast fluidized bed based on axial diffusion of clusters.

More recently, several groups (Rhodes and Geldart, 1987; Kunii and Levenspiel, 1991) have described density profiles in terms of entrainment from a vigorously bubbling or turbulent bed. Wen and Chen (1982) in their investigation of entrainment and elutriation, propose an equation for the entrainment rate of solids in the freeboard (similar to Large et al., 1976) represented by:

$$E = E_{\infty} + (E_o - E_{\infty}) \exp(-ah) \quad (2.8)$$

where:  $a$  = decay constant ( $m^{-1}$ )

$E$  = entrainment flux at riser exit ( $kg/m^2 \cdot s$ )

$E_o$  = entrainment flux at dense phase/dilute phase interface ( $kg/m^2 \cdot s$ )

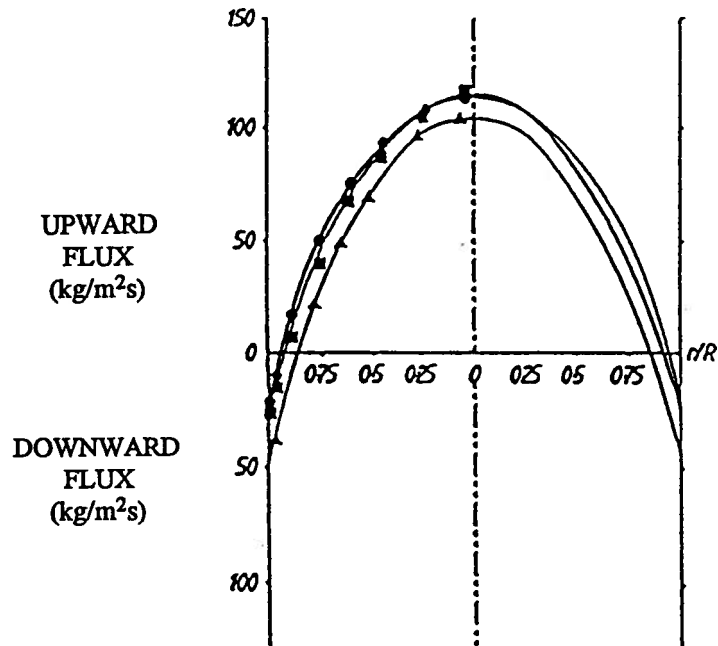
$E_{\infty}$  = entrainment flux above TDH ( $kg/m^2 \cdot s$ )

$h$  = height above bed interface (m)

Here, the decay constant,  $a$ , must be correlated to experimental data from the particular unit in question. Wen and Chen (1982) recommend  $a = 4.0 \text{ m}^{-1}$  based on experimental data collected in a bubbling fluidized bed. However, Rhodes and Geldart (1987), in applying this entrainment model to fast fluidization, found that  $a = 0.5 \text{ m}^{-1}$  provided better predictions.

### **2.2.3 Radial Solids Distribution**

Radially, there is a high degree of nonuniformity in solids density. Figure 2.5 (Rhodes et al., 1988) represents the typical radial variation in solids flux measured under fast bed conditions. Results such as those presented in Figure 2.5 have led to the identification of two radial zones in the riser of the CFB - the core of the riser, characterized by a dilute upward flowing suspension, and a relatively thin annular zone, in which solids move downwards at or near the wall of the riser. Many investigators (e.g. Weinstein et al., 1986; Monceaux et al., 1986; Brereton, 1987; Bader et al., 1988; Hartge et al., 1988) have clearly demonstrated this



Mean solids flux = 63.4 kg/m<sup>2</sup>s  
 Superficial gas velocity = 4 m/s  
 Riser diameter = 152 mm

**Figure 2.5: Typical Radial Variance in Solids Flux**  
 profiles from a column of 152 mm diameter with alumina of mean particle size of 64  $\mu\text{m}$   
 by Rhodes et al. (1988)

now commonly accepted core/annular flow structure. The annular or wall region is characterized by sheets or streamers of falling particles interspersed with periods where there is a dilute upward-moving solids suspension. Typical voidages of the streamers are in the range of 0.6 to 0.8 (Senior and Brereton, 1992).

Figure 2.6 illustrates the core/annular structure together with the direction of flow of both gas and solids. Both gas and solids flow upward in dilute suspension in the core. In the annulus, solids flow predominantly downward, while gas flows upward along the bare wall and downward within the streamers. Note that the thickness of the annular wall region diminishes with height for the case shown.

#### **2.2.4 Gas Flow**

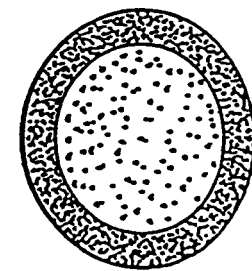
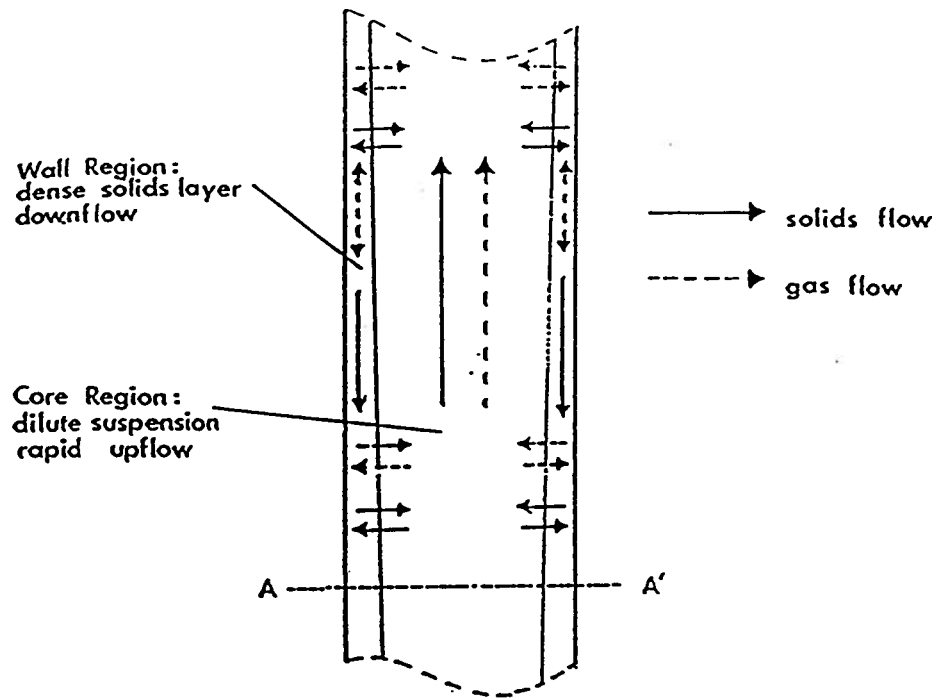
The gas flows in near plug flow throughout the core, while there is a sharp decrease in gas velocity from the inner edge of the annular region to the column wall. The reduced gas velocities in the annular region are due to both frictional effects of the wall and to the presence of the predominantly downward flow of the streamers in this region.

### **2.3 Heat Transfer**

Heat transfer in CFBs has been reviewed by, among others, Glicksman (1988), Leckner (1991), and Basu and Fraser (1991). This section briefly describes the mechanism of heat transfer in CFB boilers.

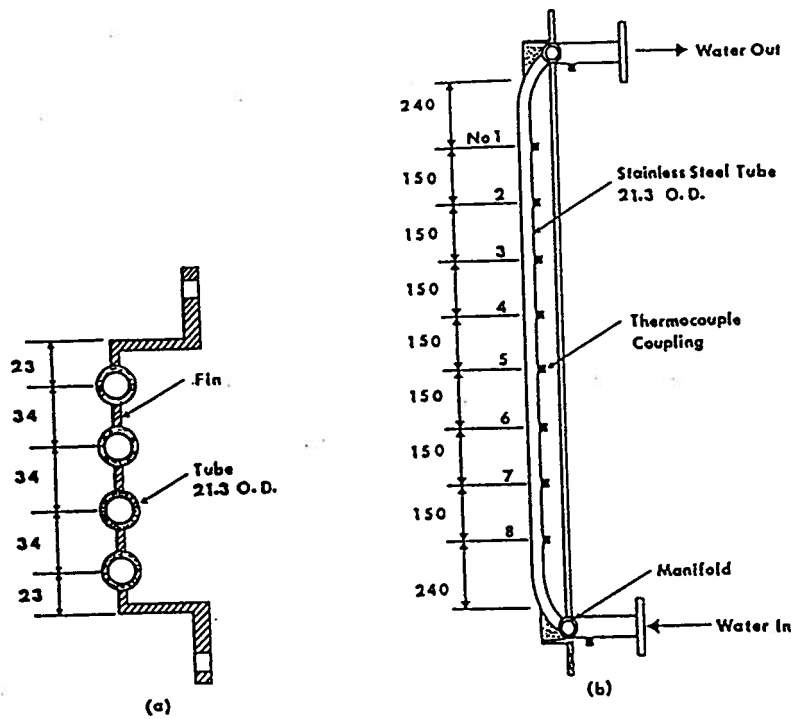
Heat transfer surfaces within CFB boilers are usually in the form of exposed membrane waterwalls. These surfaces form the containing wall of the unit. Figure 2.7 illustrates a membrane wall, which is a series of tubes connected by fins, with half of the tube surface exposed to the combustion side of the CFB reactor.

The mechanism of heat transfer to the membrane wall is of utmost importance, as it is this wall which is responsible for the removal of the majority of the heat generated within the combustor. Although some work has been reported on time-averaged heat transfer



section A-A'

**Figure 2.6: Illustration of Solids Distribution in the Riser of a CFB**  
(from Senior and Brereton, 1992)

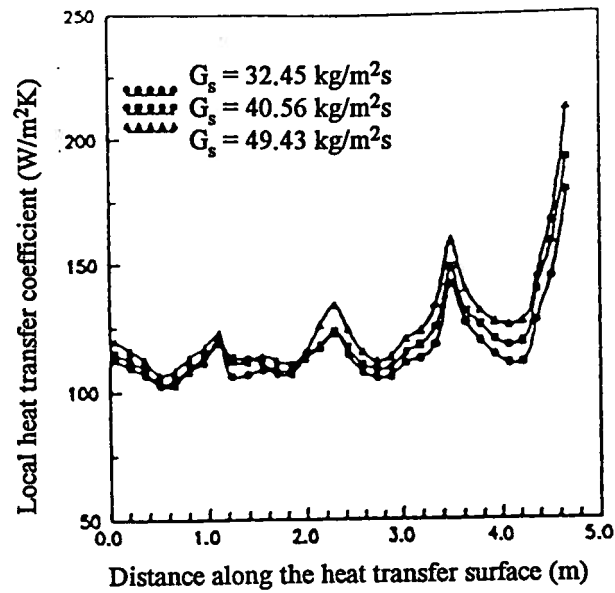


**Figure 2.7: Illustration of a Membrane Wall**  
 from Wu (1989)  
 (a) plane view, (b) side view  
 - all dimensions in mm

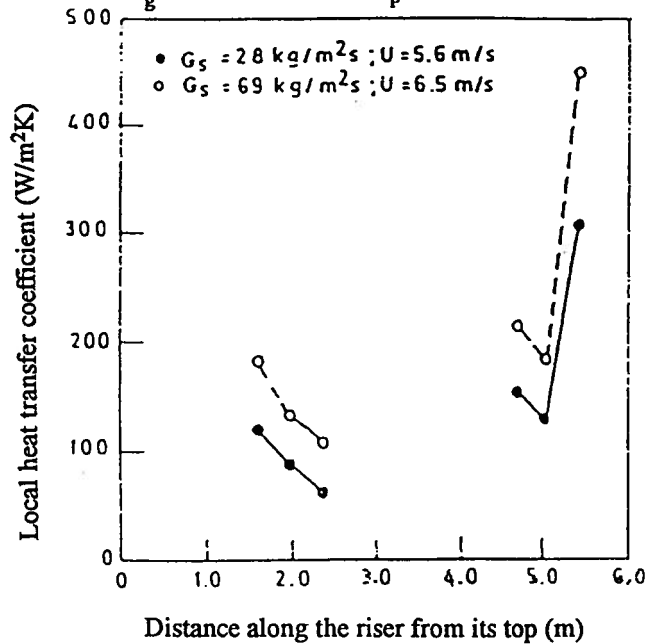
measurements (e.g. Kobro and Brereton, 1986; Subbarao and Basu, 1986), very little is known of the instantaneous local heat transfer coefficient (Wu et. al, 1989; Mahalingam and Kolar, 1993). The local heat transfer coefficient provides insight into the mechanism of transfer of heat to the walls of the CFB. The local instantaneous heat transfer coefficient is directly related to the hydrodynamics at that point. As discussed earlier, the walls of the CFB are covered by sheets, streamers, and dilute suspension, all in a state of dynamic instability. The greatest heat transfer initially occurs in areas covered by sheets and streamers, as particle convective heat transfer can more than double the heat transfer coefficient at combustion temperatures.

On a strictly qualitative basis, the heat transfer coefficient is composed of a gas convective component, a particle convective component, and a radiative component. In heat transfer from a streamer in a typical CFBC system, the radiation and particle convective components are most significant. As a streamer or packet of particles slides downwards along the heat transfer surface, it loses heat to the walls, resulting in a loss in particle temperature. Therefore, as the packet continues down the wall, there is a reduction in the local heat transfer coefficient. In studies of local heat transfer coefficients, insight has been gained into the hydrodynamic characteristics of streamers. As these packets are unstable, they fall only a certain distance before breaking up and being dispersed. As they are dispersed, new particles come into contact with the wall resulting in a rise in heat transfer coefficient.

Mahalingam and Kolar (1993) (Figure 2.8) investigated heat transfer over a 4.64 m simulated membrane wall and found for the conditions of their study (superficial velocity 4.2 to 8.2 m/s, solids fluxes 17 to 110 kg/m<sup>2</sup>·s) that there was a certain length over which the heat transfer coefficient decreases, followed by a distance over which it increases. This pattern repeated itself over the entire length of the wall. The renewal lengths were approximately 0.6 m, with areas of decreasing coefficient being termed "thermal decay length" and those of increasing coefficient as "thermal regeneration length". The authors conclude that the thermal life span of particles near the wall (thus possibly that of streamers) is 0.6 m and compare this



Variation of heat transfer coefficient along the surface length with  $G_s$  as a parameter;  $U_g = 5.7 \text{ m/s}$  and sand  $d_p = 265 \text{ microns}$ .



Variation of heat transfer coefficient along the heat transfer surface, data of Wu et al [1987]

**Figure 2.8: Variation of Local Heat Transfer Coefficient with Height**  
(figures from Mahalingam and Kolar, 1993)



with Basu (1990) who claimed a value of 0.7 m. Although generation and renewal are generally considered to involve a random process, it was found to be periodic for the conditions and apparatus investigated.

There is still a significant lack of knowledge for detailed design and experimental analysis of the heat transfer to the walls of boilers. Missing in particular are details of the local flow structures and their time and spatial variations. The effect of wall geometry and other surface defects (roughness) are also of direct importance as these determine how and where particles collect and the stability of these structures.

Of greater importance to the designer is the time-averaged, spatially-averaged heat transfer coefficient as a function of the local cross-section average suspension density. This information allows heat removal over a particular surface to be estimated. This heat transfer coefficient generally increases with suspension density due to the contribution of particle convective heat transfer. As mentioned above, as density is increased, there is an increase in the number of streamers at the wall, as well as an increase in the wall surface covered by streamers.

It is common that the heat transfer coefficient be estimated from:

$$h_{to} = h_{gc} + h_{pc} + h_{rad} \quad (2.9)$$

where:  $h_{to}$  = overall heat transfer coefficient (W/m<sup>2</sup>·K)

$h_{gc}$  = gas convective component (W/m<sup>2</sup>·K)

$h_{pc}$  = particle convective component (W/m<sup>2</sup>·K)

$h_{rad}$  = radiative component (W/m<sup>2</sup>·K)

This equation is an over-simplification, but provides a practical basis for estimating the heat transfer in CFB boilers (Grace, 1986a). The gas convective heat transfer component,  $h_{gc}$ , results from heat transfer occurring between the wall and the gas passing over the surface. For a CFB riser under conditions of relatively dilute phase flow, this component may be

estimated from correlations referring  $h_{gc}$  to an empty tube. Although the presence of particles may alter turbulence and local gas flow near the wall, this estimate is deemed adequate as, under combustion conditions, the gas convective component contributes less than 10 % to the overall heat transfer coefficient (Wu, 1989).

The radiative component,  $h_{rad}$ , can be estimated by treating the suspension as a gray body so that:

$$h_{rad} = \frac{\sigma(T_{susp}^4 - T_{surf}^4)}{\left[\left(1/e_{surf}\right) + \left(1/e_{susp}\right) - 1\right](T_{susp} - T_{surf})} \quad (2.10)$$

where:  $e_{surf}$  = emissivity of heat transfer surface

$e_{susp}$  = suspension emissivity

$T_{surf}$  = temperature of heat transfer surface (K)

$T_{susp}$  = suspension temperature (K)

$\sigma$  = Stefan-Boltzmann constant,  $5.6696 \times 10^{-8} \text{ W/m}^2 \cdot \text{K}^4$

The suspension emissivity,  $e_{susp}$ , may be approximated (Grace, 1982) as:

$$e_{susp} = 0.5(1 + e_p) \quad (2.11)$$

for risers of sufficient size and suspension densities high enough that opposite walls do not "see" each other.

Grace (1986a) utilized Martin's (1980, 1984) theory for heat transfer in fluidized beds for the prediction of the particle convective component,  $h_{pc}$ . This model is based on an analogy between particle motion and kinetic theory of gases for molecular motion. Although somewhat controversial, this model has been successful for some high velocity systems (Reh, 1985).

There is wide variation in heat transfer data from different units. The situation is somewhat analogous to hydrodynamic studies in that no single model has been able to completely explain all available information. Until the local heat transfer process is completely

understood, any attempt to determine the overall heat transfer coefficient tends to be a gross approximation requiring experimental data from the unit in question for improved calculation. This is the approach taken in this work. Wu (1989) measured heat transfer on membrane walls and in-bed tubes and provided an experimental fit for both  $e_{\text{surf}}$  and  $e_{\text{susp}}$  equal to 0.91. These values were determined in the same pilot scale equipment and under conditions similar to those employed in the work reported here. Hence they are utilized in this thesis.

With the convective component,  $h_{\text{gc}}$ , estimated as suggested earlier, the radiative component,  $h_{\text{rad}}$ , from Equation (2.10), and the overall heat transfer coefficient,  $h_{\text{to}}$ , from experimental data, the particle convective component,  $h_{\text{pc}}$ , can be determined from Equation (2.9);  $h_{\text{pc}}$  is then typically correlated to the corresponding suspension density. Glicksman (1988) and Basu (1990) found  $h_{\text{pc}}$  to be proportional to  $(1-\epsilon)^{0.5}$ , while Wu et al. (1989) imply a linear relationship. The work here confirms a linear relationship for this particular apparatus.

## **2.4 CFB Boiler**

Circulating Fluidized Bed units provide the means by which fast fluidization can be achieved. CFB equipment consists of a number of elements including a riser, some form of gas/solids separation device, a reservoir to contain bed inventory, and a solids return device. These various aspects are considered here with specific focus on boiler applications.

### **2.4.1 Riser**

Most commercial combustors are rectangular in cross-section. The cross-sectional area is designed on the basis of gas velocity and circulation rates. All units are now of membrane waterwall construction. It is common for some of the waterwalls to be covered with refractory to protect them from erosion.

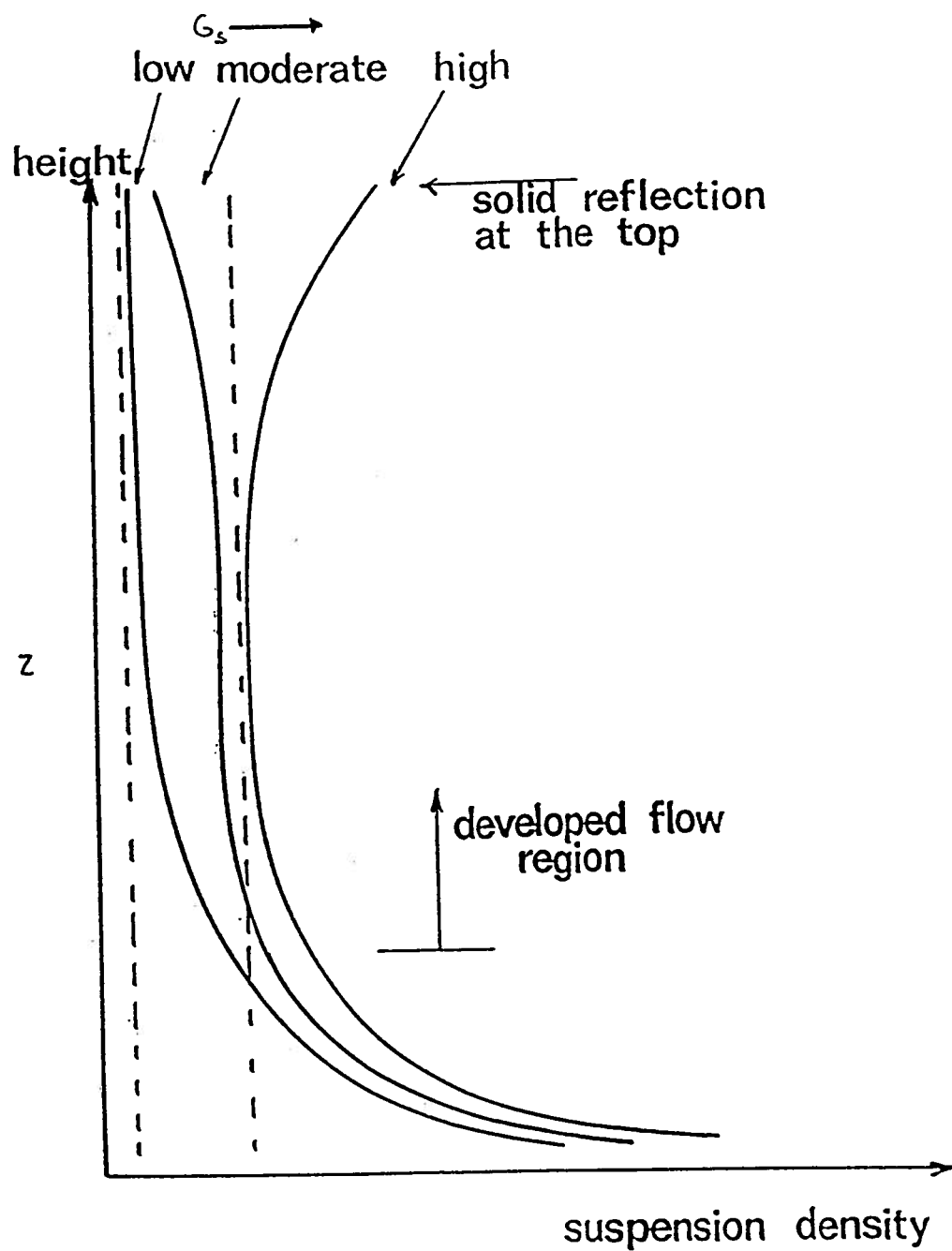
In boiler design, staged air introduction is essential. This entails injection of part of the total air at the base of the riser (primary air) and the rest at one or more levels through ports

higher up the riser (secondary air). Staged air combustion significantly reduces  $\text{NO}_x$  emissions. The section of the furnace below the secondary air ports is operated in a sub-stoichiometric condition to reduce  $\text{NO}_x$  formation. The part of the riser below the secondary air ports is usually an extensively mixed bubbling, or more typically a turbulent bed, with fast fluidization above the secondary air ports. Usually 40 to 60% of the total air is introduced as secondary air (Basu and Fraser, 1991) resulting in primary zone gas velocities of 2 to 5 m/s.

The base of the riser is sometimes tapered, with its cross-sectional area slowly expanded with increasing height over some vertical distance. The purpose of a tapered base is to increase the primary air gas velocity to aid vertical particle movement in the lower part of the riser, thereby resulting in lower solids loading and preventing agglomeration. However, Senior (1992) has provided evidence that, under some conditions, tapering can increase the suspension density in the lower zone as a result of jetting of the gas through the solids.

The riser exit geometry affects the hydrodynamic behavior in the upper portion of the combustor. If there is a gradual transition from the vertical flow direction to the horizontal passage to the solids separator (smooth exit), almost all solids reaching the top of the riser pass directly to the separator. This results in the typical axial solids loading profiles discussed earlier (e.g. Figure 2.4) where there is a continuous increase in voidage with height, right to the exit.

If, as commonly used in industry, the exit is abrupt ( $90^\circ$  bend) and especially if the exit cross-sectional area is less than that of the riser, the local bed density may increase towards the riser exit. As the solids pass through the riser, their predominant direction of flow is vertical. As they reach the exit, their vertical momentum may be great enough to carry them right to the top (or roof) of the column. Upon striking the roof, this momentum is converted to the downward direction, causing these particles to rebound back down the riser as illustrated in Figure 2.9. Whether or not a given particle negotiates the exit is a function of the momentum of the particle and the drag of the exiting gas. The fraction of solids exiting the riser is a complex function of gas velocity and solids loading. An abrupt exit tends to



**Figure 2.9: Effect of Abrupt Exit on Vertical Profile of Axial Solids Loading**  
(from Senior and Brereton, 1992)

cause an increase in solids loading towards the top, accompanied by an increase in solids residence time.

#### **2.4.2 Gas/Solids Separation**

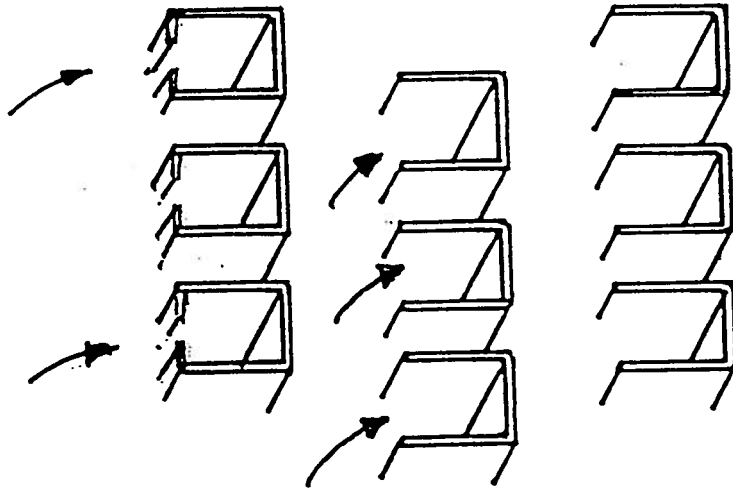
The most common primary gas/solids separation device is the cyclone. The high gas throughput and high solids concentrations in CFBC systems create operating conditions different from those in process industries (Basu and Fraser, 1991). As a result, most manufacturers use more than one cyclone in parallel or in series to maintain high separation efficiencies.

An alternative to a cyclone is the use of an inertial separator, a labyrinth of multiple rows of U-beams vertically suspended in the gas stream (Figure 2.10). The simplicity of construction (a rectangular extension of the boiler) reduces fabrication cost. As well, the nature of these separators allows them to be made as large as required, eliminating the need for multiple cyclone separators in larger units. These U-beam separators cause lower pressure drops which may lead to a significant reduction in fan requirements at the expense of somewhat lower collection efficiencies than properly designed cyclones.

#### **2.4.3 Solids Return Systems**

Solids return systems, used to return solids captured in the gas/solids separator(s) to the base of the combustor, are quite varied. These can be classified into two main categories - mechanical and non-mechanical devices. Mechanical devices employ some physical mechanism to move the solids to the riser, while non-mechanical valves function without any mechanical device.

Mechanical valves are rarely used, except in small research facilities. They can vary from butterfly valves where the size of opening determines the rate of flow of solids to screw-type feeders where the solids flowrate is set by the screw speed. Although providing extensive control over the rate of solids returned to the riser, they are complex additions at the



**Figure 2.10: U-Beam Separator**  
(from Basu and Fraser, 1991)

industrial level adding to capital cost and maintenance, and lowering reliability, especially at high temperatures.

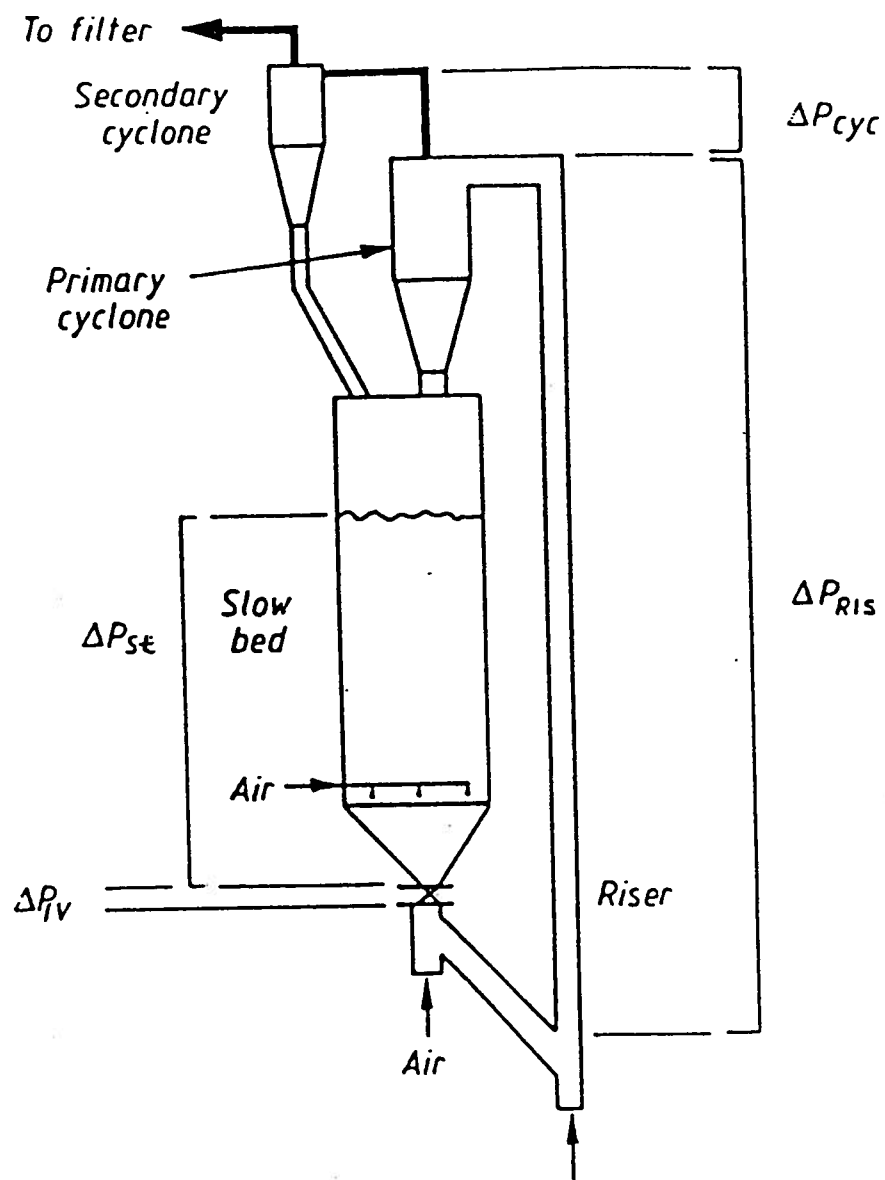
Due to their inherent simplicity, non-mechanical valves are widely used in CFB boiler technology. These valves provide for solids recycle without moving parts, the primary purpose being the transfer of the solids from the low pressure point in the loop (solids separator) back to the higher pressure riser base. Non-mechanical valves can be subclassified into two categories - those providing a controlled rate of flow of solids and those providing no control, acting only as a solids flow device.

In general, non-mechanical valves all operate by providing a high pressure point in the loop to prevent gas by-passing the riser through the return leg and to facilitate the transfer of solids from the low pressure point to the high pressure point in the system. Figure 2.11 represents a typical system pressure balance. Key to the operation of these systems is the standpipe. The solids inventory in this standpipe provides a certain head or pressure drop. The pressure drop across this moving bed dictates the drop around the rest of the loop.

Examples of controllable valves include the L-valve, J-valve, and reverse seal. Automatic valves include the seal pot, loop seal, and V-valve. These devices provide a pressure seal allowing for the passage of solids back to the riser. If the flow of solids is changed, the valve simply adjusts itself to the new flowrate. In general, these valves operate under fluidized conditions. By fluidizing the solids, the loop pressure drop is essentially fixed, thereby fixing the solids throughput.

The L-valve is the simplest and most widely used in CFB applications (see Knowlton, 1986). It can be operated both as a controlled or uncontrolled solids flow device. Solids are passed from the solids separation device to the valve through a hopper or standpipe. There is a certain inventory of solids in the system, the majority of which (typically > 70 %) is stored in this section. These solids move downward under moving bed flow, with very little axial or radial mixing of solids.





**Figure 2.11: CFB System Pressure Balance**  
(from Rhodes and Geldart, 1987))

A key to the operation of an L-valve (or any non-mechanical valve) is the system pressure balance illustrated in Figure 2.11. The low pressure point in the system is the solids separator, while the high pressure point is at the location of aeration. The whole system operates such that the pressure drop across the slow bed is the same as that through the riser loop, i.e.:

$$\Delta P_{st} = \Delta P_{lv} + \Delta P_{ris} + \Delta P_{cyc} \quad (2.12)$$

where:  $\Delta P_{cyc}$  = pressure drop across solids separator (kPa)  
 $\Delta P_{lv}$  = pressure drop around L-valve to riser base (kPa)  
 $\Delta P_{ris}$  = pressure drop across riser (kPa)  
 $\Delta P_{st}$  = pressure drop across standpipe moving bed (kPa)

As the solids in the standpipe are in moving bed flow, an increase in the aeration air causes an increase in the pressure drop across the standpipe. The system responds with an increase in solids flow rate resulting in an increase in riser solids loading and an increase in riser pressure drop, satisfying the pressure balance of Equation (2.12).

The operation of an L-valve is strongly dependent on solids inventory (standpipe bed height), with the height of this moving bed essentially dictating the range of solids return rates achievable. This is best illustrated by a conventional fixed bed. As the flowrate of gas through a bed of solids is increased, there is a corresponding increase in pressure drop. If this flow is increased enough, the bed becomes fluidized. Once in a fluidized state, any changes in gas velocity result in changes in bed height, but very little change in bed pressure drop. The pressure drop required to fluidize this bed is directly related to its static bed depth, with a deeper bed resulting in a higher pressure drop upon fluidization. The greater the bed depth, the greater the range of pressures available for solids flow control. Once this bed has been fluidized, the L-valve behaves as an uncontrolled non-mechanical valve since the system pressure drop can no longer be varied appreciably.

## **2.5 Dynamic Modeling**

Dynamic models of the CFB tend to be modular in nature and to require cell block solution schemes. This is due to the recognition that a CFB system is not a single unit but a complete assembly comprised of a riser, solids separation device, solids storage vessel, and solids return conduit and valve - all interactively affecting the overall behaviour of the system.

At present, all dynamic models are extreme simplifications, as inclusion of time into the modeling realm enhances unresolved system behaviour at the micro-scale level. No continuum approach has been published. This is not surprising since, as noted above, the current level of steady-state modeling (especially for riser hydrodynamics and heat transfer) is itself far from complete. For instance, as clustering and wall strand formation have yet to be quantitatively addressed, there is no advantage to their inclusion at the dynamic level. This is illustrated by the fact that, although known to be critical to the process of heat transfer to surfaces, heat transfer calculations are generally performed and coefficients are reported, based on average solids loading across the riser cross-section, without direct reference to wall coverage.

Given the paucity of understanding, modelers have approached the system from a macro-scale point of view, leading to cell models for the characterization of the reaction process in the riser of the CFB. Weiss and Fett (1986) developed one such model for the decomposition of sodium bicarbonate, a simple reaction which could be used to provide model validation for application to more complex reactions. The model assumes a gas/solids suspension phase of uniform radial density across the entire column cross-section. Backmixing of solids is included, although it is unclear how the rate of backmixing is to be determined. The authors also state that the void fraction "may" be a function of height inside the reactor. Whether or not a profile was used or void fraction was considered uniform over the reactor height is not clear.

Weiss et al. (1988) extended this model to coal combustion. Here they clearly state that voidage is a function of height and used the solids backmixing rate as a fitting parameter.

Mori et. al (1991) proposed a dynamic model for a 70 T/h Multi-Solid Circulating Fluidized Bed Boiler. The purpose of this model was to investigate the control system. The riser is considered in three zones - a dense zone, a reducing zone, and an oxidizing zone. Both gas and solids are considered to be well mixed in each of these zones and backflow of solids between zones is taken into account. Devolatilization and subsequent combustion of volatiles are deemed to be instantaneous, while the char combustion reactions are taken to be:



SO<sub>2</sub> generation and capture, as well as NO<sub>x</sub> formation and reduction, are not considered. The solids loading is assumed to vary in the axial direction only and is obtained from steady-state experimental pressure drop measurements. The hold-up is then assumed to vary linearly with load between initial and final conditions.

The only unknown parameter in this model is the solids downflow ratio,  $X_1$ , representing the backmixing between the three riser zones.  $X_1$  is formally defined as the ratio of downflow of solid particles in the combustor to the outside recycle flowrate. In comparing experimental temperature profiles with those calculated for various downflow ratios,  $X_1 = 15$  was selected in order to match their almost uniform measurements.

Zhang et. al (1991) developed a dynamic model for CFB combustion emphasizing the effect of particle size distribution of both bed and feed materials. The combustor is considered to consist of two zones - a dense (bubbling bed) zone and a dilute zone. The dilute zone is

sub-divided into an arbitrary number of lump-parameter blocks in series, with gas and solids assumed to be well-mixed in each block, and with no backmixing of the gas between blocks. The hydrodynamics of the riser are considered to be at steady-state since the gas and solids dynamics are considered relatively fast. The voidage profile is obtained from the correlation of Yu (1989), while the solids entrainment rate along the height of the riser is taken from Wen and Chen (1982) (Equation 2.8). This equation is solved for each discrete particle size. Included in the reaction scheme are the reduction of NO by carbon and the capture of SO<sub>2</sub> by CaO. Dynamic changes of gas concentration are directly related to the rates of reaction.

Hyppanan et. al (1993) presented a dynamic simulation of an Ahlstrom Pyroflow 125 MWe CFB utility power plant. This model incorporated not only the boiler, but also the steam process. The riser is divided into numerous compartments in series, each assumed to be well-mixed. The axial suspension density profile is determined using an empirical correlation of unknown origin. It is also stated that the "dynamics of the suspension density profiles are determined by using an experimental time constant for flow dynamics." The meaning of this statement is unclear. It probably implies a smoothing of calculated profiles for a time-dependent transition from one operating condition to another. As in the model presented in this work, the solids circulation flux has been correlated based on experimental data. Volatiles release, burning, and mixing are combined in a single expression and fitted to experimental data. Solids mixing between the cells is described using a dispersion model whose details are not discussed. The heat transfer coefficient is correlated as a function of suspension density.

Noting a paucity of experimental validation in previous work, Hyppanan et. al present experimental results from the 125 MWe Seinäjoki power plant to provide for verification. They also developed an advanced control structure with this model and applied it to the plant, presenting data showing improved load following based on this new structure. Their work shows the utility of the dynamic model for developing control structures and optimizing particular aspects of the plant through simulation. Unfortunately, most likely due to proprietary factors, no mention of the modified control strategy is made other than that it is

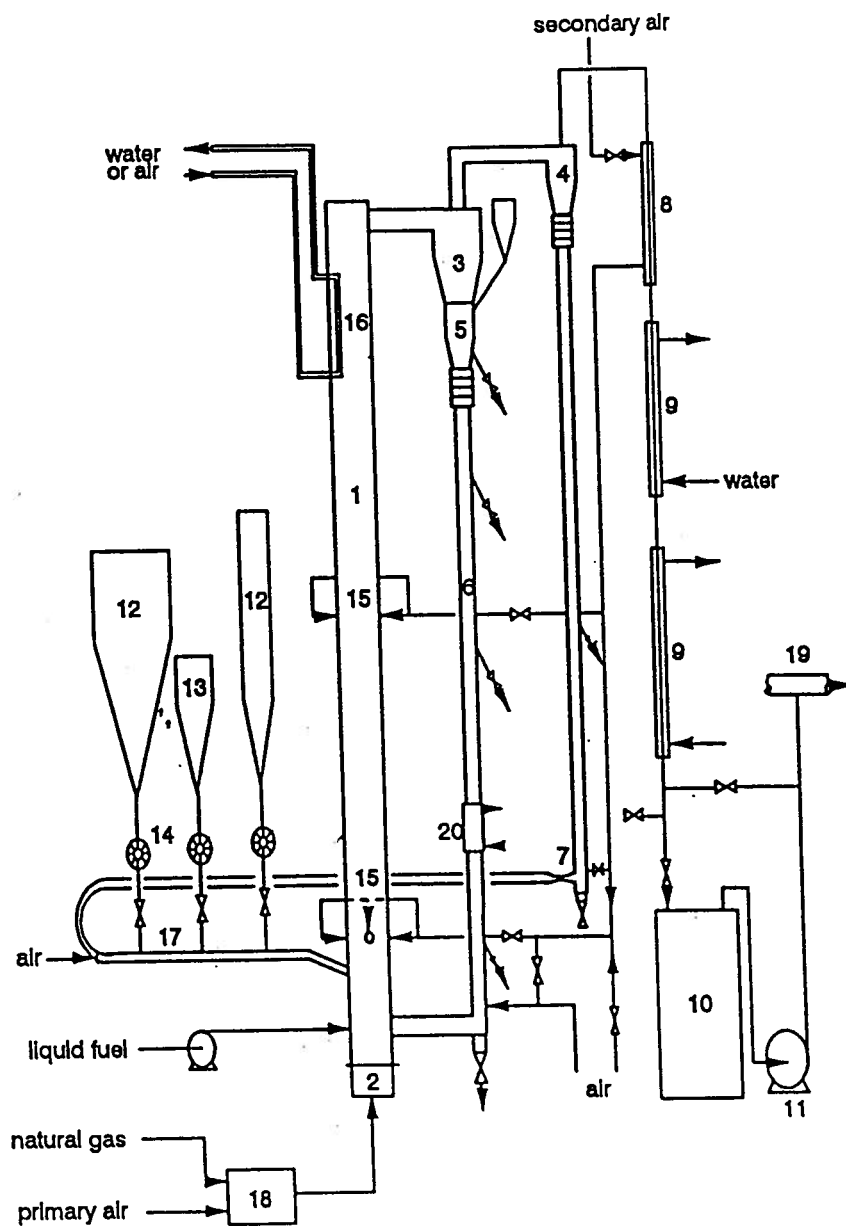
**"advanced". Whether advanced from a control theory point of view or a schematic modification leading to uniqueness is open to speculation.**

## **CHAPTER 3 - EXPERIMENTAL FACILITIES**

The UBC-CFBC pilot plant used for the tests carried out in this work is shown in Figure 3.1. Detailed descriptions of each of the key reactor and return system components follow.

### **3.1 Reactor Shaft**

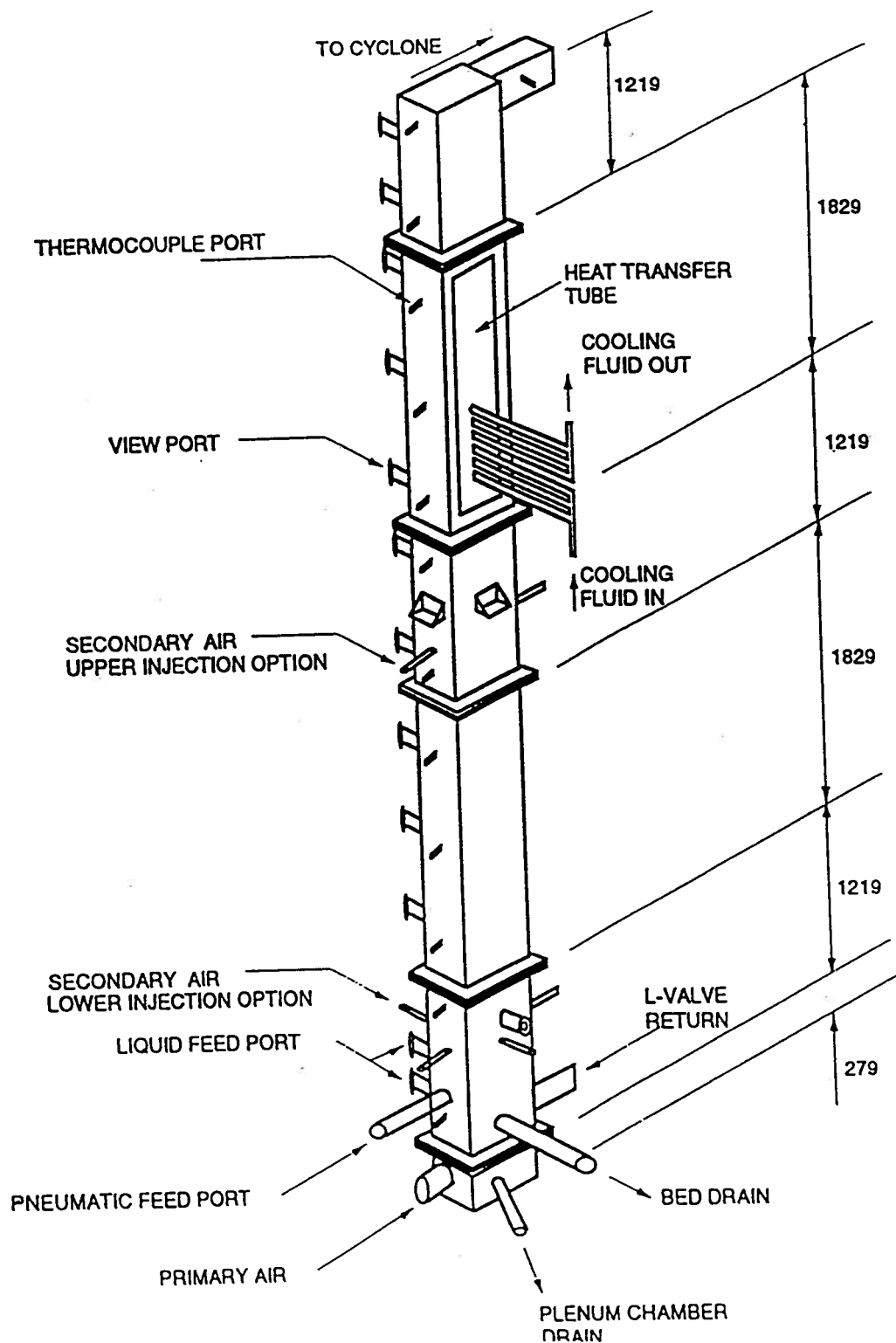
The principal reactor shaft or riser is composed of five refractory-lined flanged sections providing a chamber of 152 mm square cross-section with an overall height of 7.32 m. A view of the reactor is shown in Figure 3.2. The refractory is erosion resistant and held in place by pins welded to the outside steel walls. The bottom section, which has stainless steel walls, is tapered gradually on the inside from a 51 mm x 152 mm cross-section at the bottom to 152 mm x 152 mm over a height of 1.22 m. The purpose of this feature is to provide a high velocity region which helps prevent agglomeration and sintering. Pressure taps and thermocouples are located at 610 mm maximum intervals along opposite left and right faces of the column. In addition, there are twelve regularly spaced 41 mm diameter ports which can be used as viewing ports, for withdrawal of gas and/or solids samples, or for



1. Reactor; 2. Windbox; 3. Primary cyclone; 4. Secondary cyclone; 5. Recycle hopper; 6. Standpipe; 7. Eductor; 8. Secondary air preheater; 9. Flue gas cooler; 10. Baghouse; 11. Induced draught fan; 12. Fuel hopper; 13. Sorbent hopper; 14. Rotary valves; 15. Secondary air ports; 16. Membrane wall; 17. Pneumatic feed line; 18. External burner; 19. Ventilation; 20. Calorimetric section;

**Figure 3.1: Simplified Schematic of CFBC Facility at UBC**





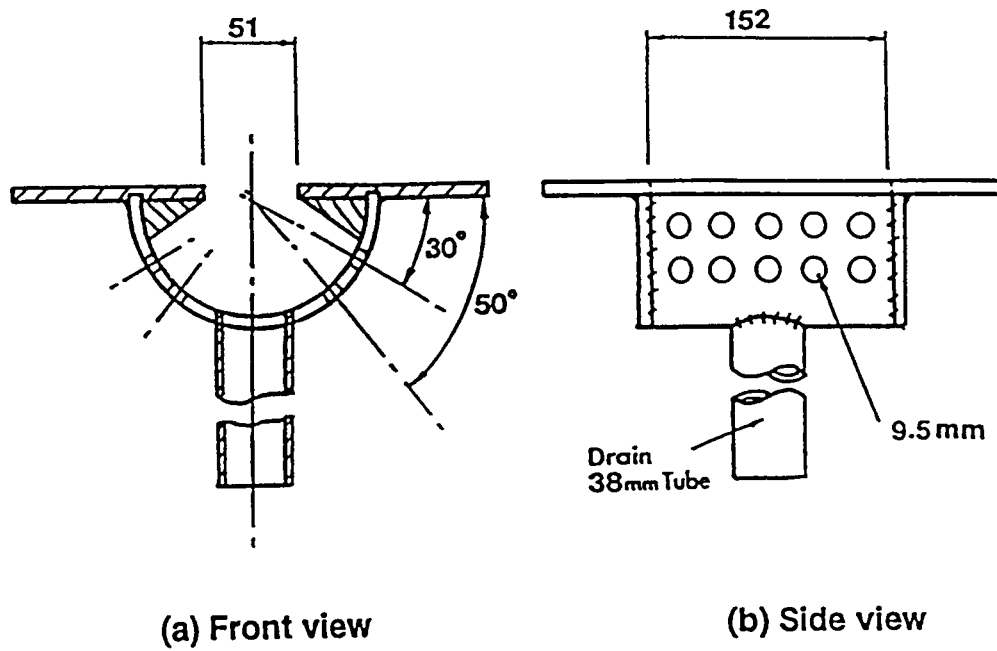
**Figure 3.2: View of Principal Refractory Lined Reactor Column**

insertion of probes and feed nozzles. The distributor and plenum chamber are suspended from the bottom of the reactor for easy removal. Primary air is introduced to the bottom of the reactor through a novel distributor shown in Figure 3.3. This distributor has twenty 9.5 mm diameter orifices drilled at 30 and 50 degrees to the horizontal axis. Located at the centre of the base is a 38 mm tube which allows easy removal of small agglomerates of sintered solids, and other oversize material. The plenum chamber is also equipped with a drain pipe for removal of solids that might flow back through the distributor during shut-down. For start-up of the unit, the primary air is preheated in an external burner by combustion of natural gas.

As shown in Figure 3.1, there are four levels of air introduction to the riser. Primary air enters principally through the distributor at the bottom of the unit. Additional air, also defined as primary for the purpose of defining a primary-to-secondary air split because of its proximity to the primary air distributor, enters through the pneumatic feed system. This system is used to recycle secondary cyclone ash and to convey fuel into the reactor. The pneumatic transport air has a consistent flow of  $8 \times 10^{-3} \text{ m}^3/\text{s}$  and is typically 20% of the total air flow. Secondary air enters through one of two levels, either a lower manifold, 1 metre above the air distributor, or an upper level, 3.1 m from the base. In all the combustion trials described in this thesis, the secondary air was introduced entirely at the upper level.

### **3.2 Fuel Feed System**

The feed system varies according to the type of fuel burned. For solid fuel feeding, the system consists of up to three sealed hoppers - one for a high reactivity start-up coal, the second for the fuel, and a third for sorbent. The capacity is 2 drums of solid fuel (approx. 360 kg) for the large hopper, 1 drum (approx. 180 kg) for the smaller start-up hopper, and approximately 20 kg for the limestone hopper. The solid fuel or fuel/sorbent mixture is fed pneumatically through a 38 mm diameter pipe entering the reactor at an angle of 15 degrees downward to the horizontal. Generally, the pneumatic conveying air is first used to pick up secondary cyclone solids; it then entrains the feed fuel and limestone. If it is not desirable to



**Figure 3.3: Primary Air Distributor**

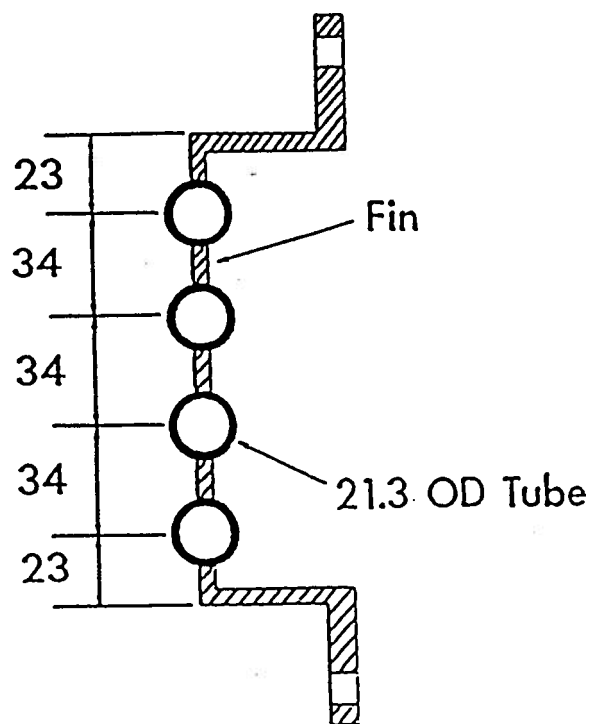
recycle the secondary cyclone catch, it is possible to take fuel feed air directly from the main compressed air line. The solids feed rates are controlled by means of rotary valves, one per hopper, with rubber impellers and a transparent front to allow visual verification that the feeder is operating properly at all times. All three hoppers are mounted on load cells which allow the feed rates to be determined by weight loss over an approximately 20 minute period. Limestone addition was not used for the experimental work reported here.

### **3.3 Heat Transfer Surfaces**

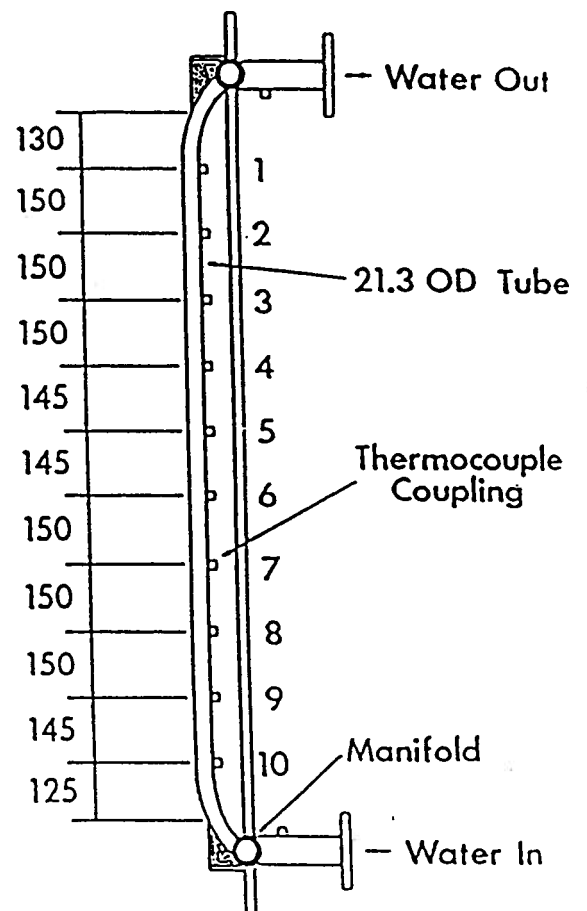
Two heat transfer configurations are available for the pilot plant CFB. These are interchangeable to allow different heat transfer problems to be studied, and to permit maximum flexibility with respect to fuels fired.

The first heat transfer configuration is a membrane wall heat transfer surface. It consists of four vertical stainless steel tubes of 21.3 mm OD and 13.9 mm ID, separated by longitudinal flat fins welded along their lengths as shown in Figure 3.4. Thermocouples are positioned at 152 mm intervals inside one of the central tubes, and at the inlet and outlet of the cooling water in order that heat transfer coefficients can be determined for different intervals. The cooling surface has a total exposed area of 0.30 m<sup>2</sup> and a projected surface area of 0.23 m<sup>2</sup>. With this surface in place, it is possible to burn high calorific fuels (HHV>28 MJ/kg) at a minimum gas superficial velocity of 6 m/s while maintaining the reactor temperature above 850°C. The surface is located in the upper section of the combustor as shown in Figure 3.2. It is interchangeable with the second surface described below.

During combustion trials of low calorific fuels with high moisture contents, such as lignite, it is not possible to run at a low load condition (6 m/s) and maintain the reactor temperature above 850°C. A second heat transfer surface is therefore utilized. This section contains three heavy-wall stainless steel tubes of 12.7 mm OD and 9.4 mm ID, in a hairpin configuration as shown in Figure 3.5. This surface reasonably approximates a superheater platen. Water or air can be used as cooling fluid in one, two or three tubes at a time. This



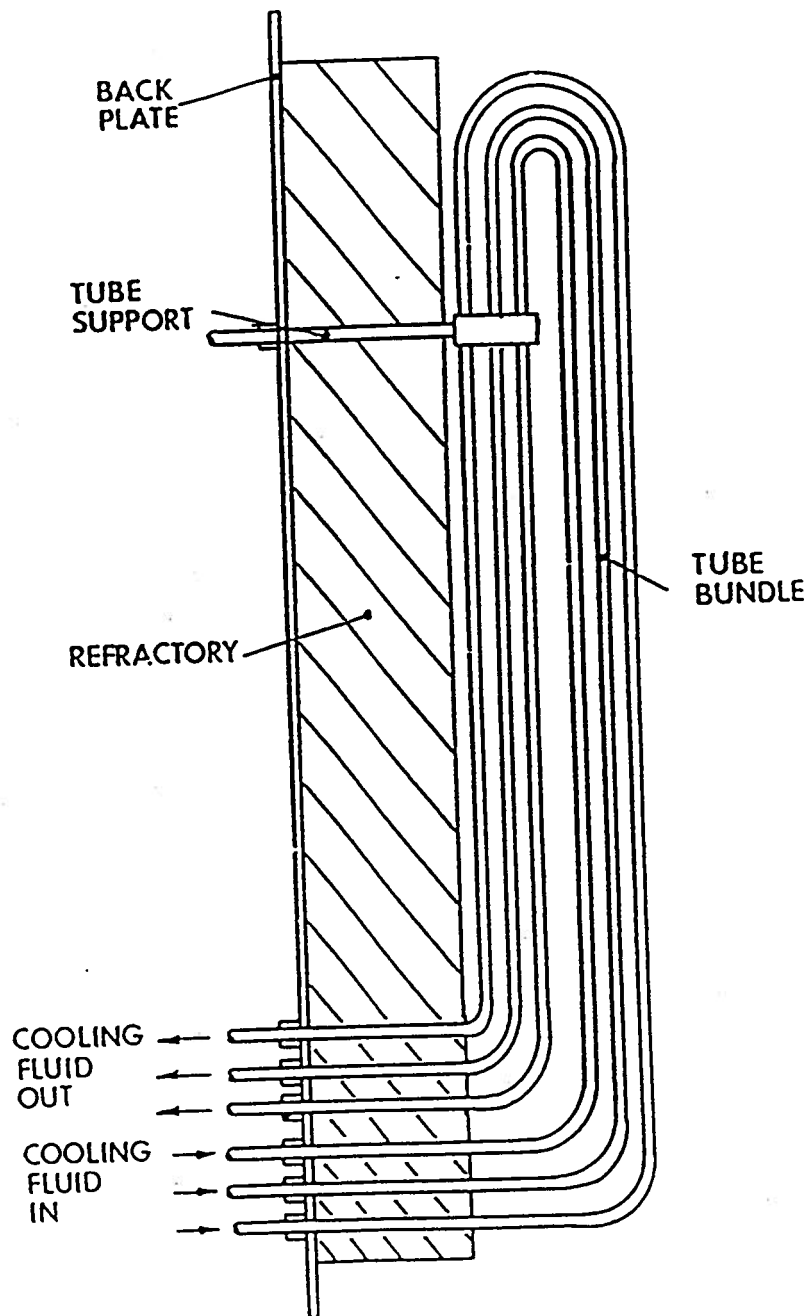
(a) Plan view



(b) section view

**Figure 3.4: Membrane Wall Heat Transfer Surface**

(all dimensions in mm)



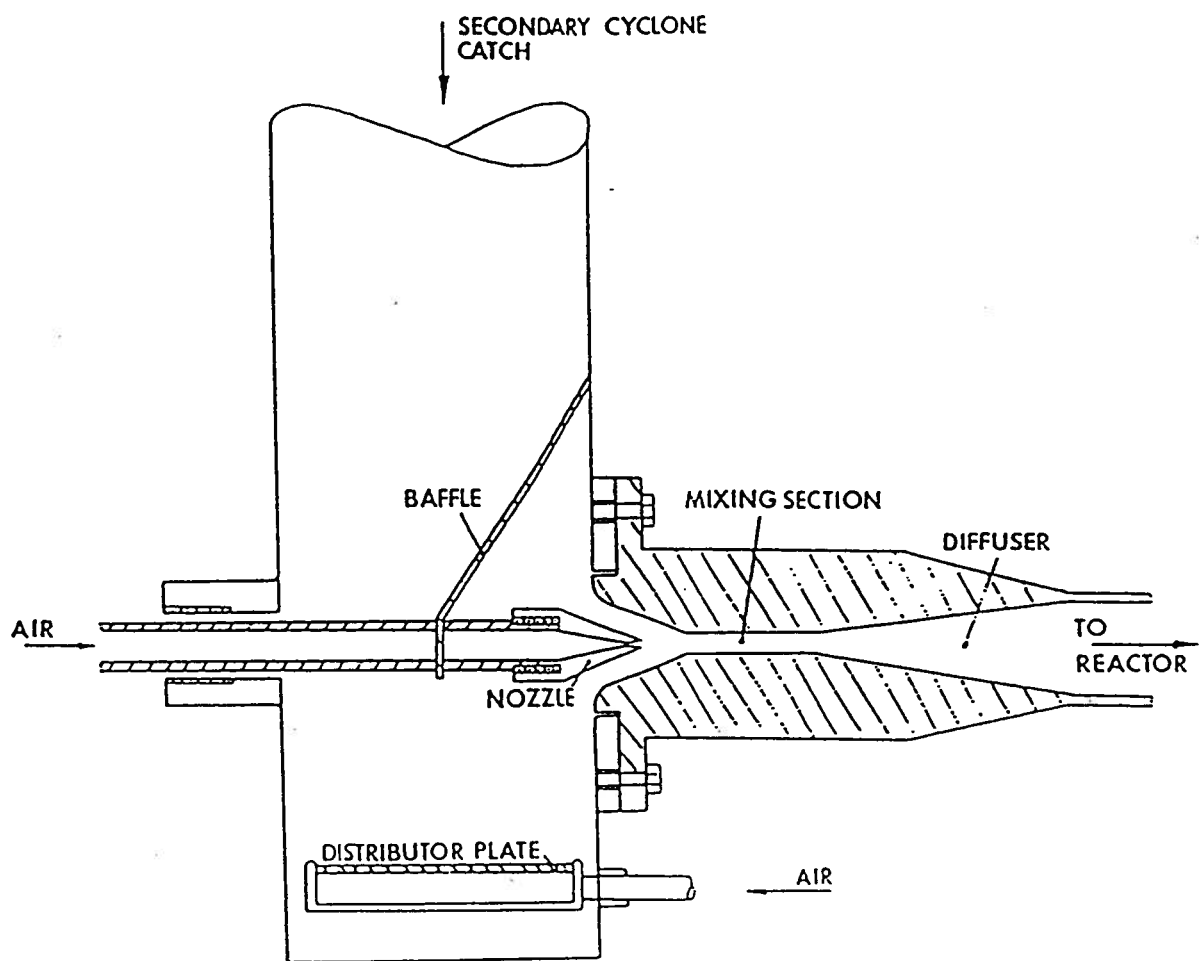
**Figure 3.5: Hairpin Heat Transfer Surface Configuration**

allows the heat removal rate to be varied from 0 to approximately 20 kW without problems due to thermal expansion, since the tube bundle can expand freely inside the reactor. The heat transfer surface has a total exposed area of 0.34 m<sup>2</sup>. This configuration was used with water flowing through one tube for all the trials described in this report.

### **3.4 Solids Recycle Systems**

Gas and entrained solids leaving the top of the reactor enter a refractory-lined medium-efficiency primary cyclone of inside diameter 305 mm. Solids captured in the primary cyclone drop into a conical recycle hopper. Make-up inert particles can also be added in this vessel from a small external hopper. From the bottom of the vessel, the solids descend in a moving packed bed or fully fluidised flow in a 102 mm ID, 4.72 m long externally insulated stainless steel standpipe, forming the vertical leg of an L-valve. There is a bellows-type expansion joint between the top of the standpipe and the bottom of the recycle hopper to allow for thermal expansion. The solids are returned to the reactor, 152 mm above the primary air distributor, through the horizontal section of the L-valve, which is 790 mm long and 102 mm ID. The circulation rate is controlled by the amount of aeration air fed to the L-valve at a single point, 102 mm above the horizontal axis of the L-valve.

Gas and entrained solids leaving the primary cyclone are directed to a 203 mm ID, high efficiency secondary cyclone. This cyclone is made of stainless steel and is insulated externally. Solids captured in the secondary cyclone fall into a 76 mm diameter dipleg. From here they are returned continuously using a venturi eductor system shown in Figure 3.6. The purpose of the eductor is to enable the solids from the secondary cyclone, together with a small amount of gas, to be returned to the reactor bottom which is at a higher pressure. A small amount of air is fed to the distributor at the bottom of the dipleg (see Figure 3.6) to keep the solids fluidized. Larger particles, which might obstruct the venturi section, fall onto the distributor. The distributor can be rotated to allow removal of solids through a ball valve located below. The air fed to the eductor nozzle, together with that which is entrained from



**Figure 3.6: Eductor Configuration for Secondary Cyclone Return**



the base of the cyclone, is then used for pneumatic solid fuel feeding. The oxygen or carbon dioxide concentration can be measured in the dipleg to ensure that the air introduced in the eductor and by the distributor is drawn downward so that the performance of the secondary cyclone is not adversely affected by air drawn from below.

### **3.5 Flue Gas Cooling and Cleaning**

Gas leaving the secondary cyclone is cooled on the inside of three double-pipe heat exchangers in series. The first of these heat exchangers may be used as the secondary air preheater, while the other two are water-cooled. The third heat exchanger offers the possibility of further cooling with three independently controlled internal water-cooled coils. Particulate solids are removed from the flue gas in the baghouse before the flue gas enters an exhaust duct connected to an induced draught (I.D.) fan. The flue gas is then directed into the building ventilation duct. The I.D. fan creates a negative pressure in the baghouse which eliminates the risk of flue gas leaking into the laboratory.

### **3.6 Gas Sampling System**

Gas may be sampled from five axial positions along the reactor, from one position between the two cyclones, and from a further point which monitors the flue gas after the first heat exchanger. The gas from any of these seven positions can be monitored continuously to determine concentrations of O<sub>2</sub>, CO<sub>2</sub>, CO, SO<sub>2</sub>, NO<sub>x</sub>, N<sub>2</sub>O and unburned hydrocarbons (as CH<sub>4</sub>). For the purposes of this work, the gases were sampled continuously from the flue gas.

The sampled gas passes through a porous stainless steel filter, in which gas and solids are separated. Gas is withdrawn through a heated sample line discharging directly into a cold trap. A drying tube filled with magnesium perchlorate (Mg(ClO<sub>4</sub>)<sub>2</sub>) removes any residual moisture before the sample gas reaches a manifold on the analyser control board. In previous work, magnesium perchlorate was found to be the optimum desiccant; while absorbing water, it does not appear to absorb any of the pollutant species of interest.

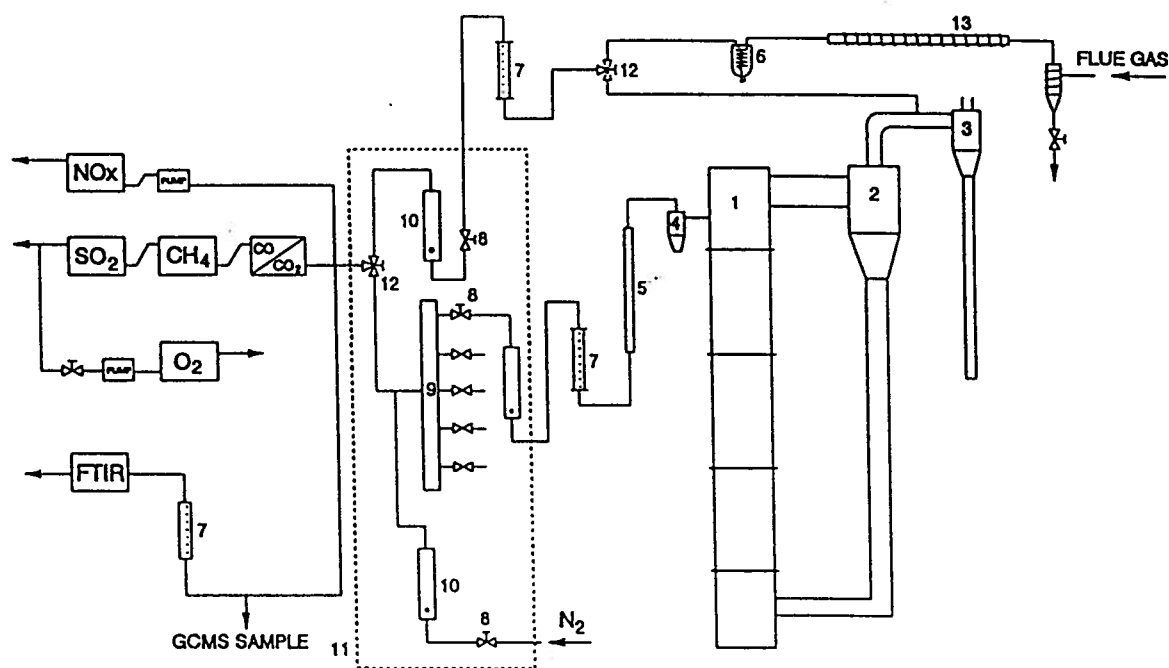
To minimize the possibility of reactions in the sampling lines, all sampling lines are stainless steel or Teflon. Air purge is used routinely to back-flush solids from the filters and sample lines. To ensure accuracy of gas sampling, the combustor is controlled at a small positive pressure using a damper on the flue gas line just upstream of the baghouse. Hence, no vacuum pump is needed on the sample line and there is no possibility of dilution due to air leakage into the sampling lines. A schematic of the complete sampling system is provided in Figure 3.7.

Continuous on-line gas analysers are used for  $O_2$ ,  $CO_2$ ,  $CO$ ,  $NO$  and  $SO_2$  concentration measurements. The details of the analysers are given in Table 3.1. For the control work reported here, the  $O_2$  analyser and pump were removed from the gas analyser train (see Figure 3.7), and were installed after a dryer in a separate line immediately following the flue gas sample filter (upstream from 13 in Figure 3.7). This configuration provided the fastest possible analysis of  $O_2$  for more up-to-date measurement and tighter control, with total response times (analyser and sample line lag) reduced from 2 to 3 minutes to less than 30 seconds.

### **3.7 Instrumentation, Data Acquisition, and Control**

The system instrumentation consists of the following:

- (1) flowmeters for measuring all air and in-bed heat exchanger water flow rates;
- (2) thermocouples throughout the entire system (Figure 3.8) to allow measurement of temperature profiles;
- (3) pressure transducers connected to pressure taps on the main reactor (see Figure 3.8 for locations) to allow measurement of pressure profiles and of the overall pressure drop;
- (4) load cells supporting all three hoppers to determine the solid and sorbent feedrates;
- (5) gas analysers (Table 3.1) to monitor the composition of the gases sampled from any of the positions described in section 3.6 above.



**Figure 3.7: Gas Sampling System**

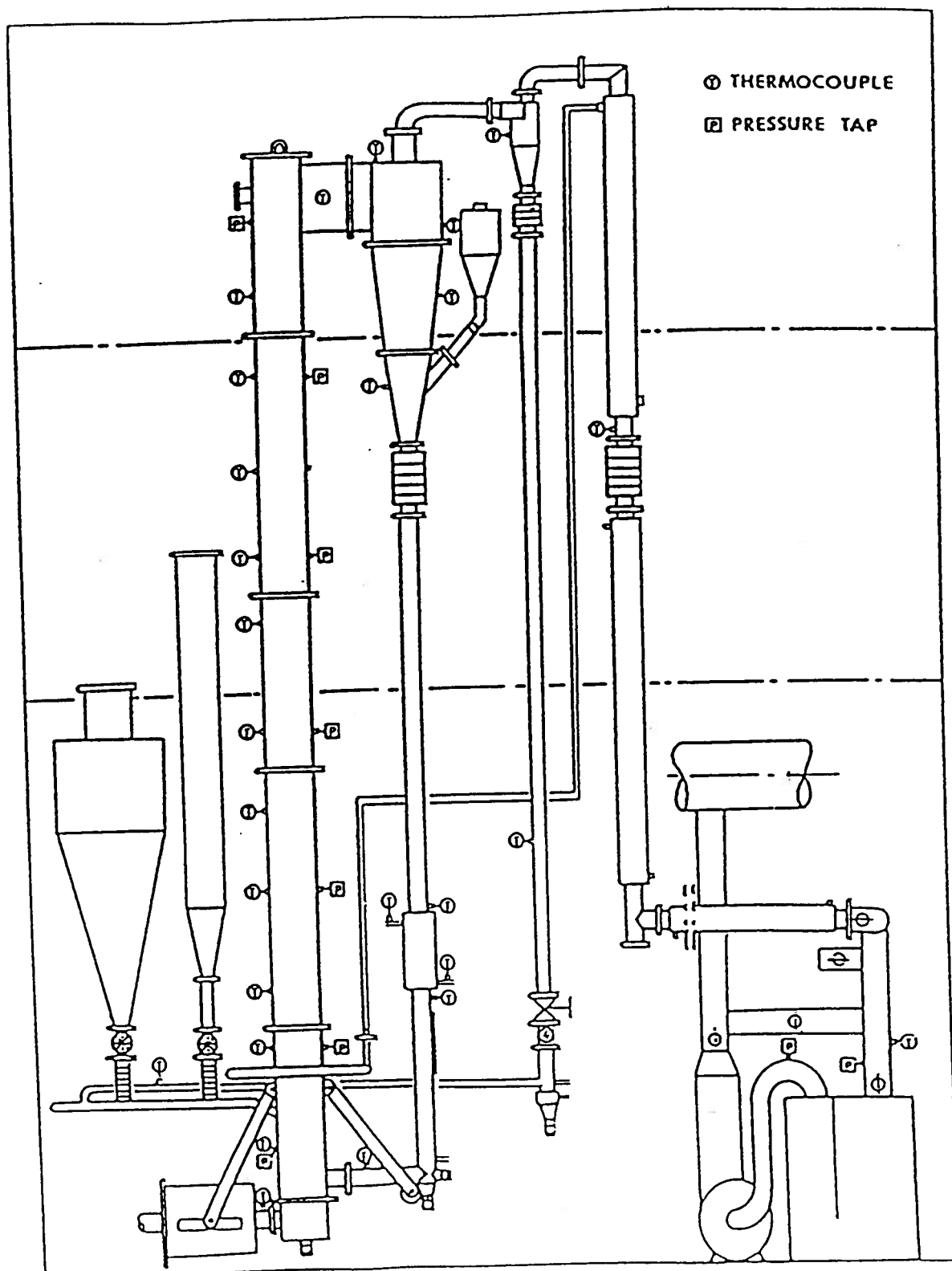
1. Reactor; 2. Primary cyclone; 3. Secondary cyclone; 4. Gas filter; 5. Heat Exchanger; 6. Water trap; 7. Dryer; 8. Flow control valve; 9. Manifold; 10. Rotameter; 11. Control panel; 12. Three-way valve; 13. Electrically traced gas sampling line

**Table 3.1: Key features of gas analysers**

Gas	Principle	Range	Response Time	Accuracy
O <sub>2</sub>	Paramagnetic	0-25% or 0-10%	20 s	1%
CO	NDIR*	0-1000 ppm	5 s	1%
CO <sub>2</sub>	NDIR	0-20%	5 s	1%
CH <sub>4</sub>	NDIR	0-0.2% or 1-0.5%	5 s	1%
NO	Chemiluminescence	0-250 ppm or 0-500 ppm	3 min.	1%
SO <sub>2</sub>	NDIR	0-1000 ppm or 0-3000 ppm or 0-5000 ppm	5 s	0.5%
N <sub>2</sub> O	FTIR**	0-1000 ppm	1 min.	+/- 5 ppm

\* Non-dispersive Infrared

\*\* Fourier Transform Infrared



**Figure 3.8: Thermocouple and Pressure Tap Locations**

An AT&T 6300 PC, programmed in BASIC, equipped with a Metrabyte interface, is used for data acquisition. The interface may be expanded with up to eight multiplexer boards with 16 channels on each board. Electrical outputs from thermocouples, pressure transducers and gas analysers can be stored and processed in the computer. The load cells are connected through summation boxes to digital readouts. Air and water flow rates are read from calibrated rotameters.

### **3.8 Modifications for Current Project**

Modifications incorporated in the pilot plant for the work reported here are summarized below.

Three control valves were installed parallel to the respective rotameters to allow either manual or computer control of primary air, secondary air, and L-valve aeration rates. Both the primary and secondary air control valves are 1" (25mm) Foxboro V1 series Stabilflo control valves, while the L-valve aeration control valve is a ½" (13mm) Foxboro V4A needle style control valve. All valves are pneumatically actuated by a 3 to 15 psi (20 to 100 kPa) nominal input pressure, and are driven by a current-to-air positioner which converts a 4 to 20 mA control signal to actuation pressure. 2" (50mm) orifice runs were installed in the primary and secondary air lines to allow the flowrates, via orifice pressure drop, to be monitored by computer for on-line logging and control of these flows.

The rotary valve controllers had facilities in place for the connection of a 4 to 20 mA control signal. These can be operated either in manual or auto mode by selection with a toggle switch.

A new data logging system and computer were dedicated to logging and control for this work. Optomux, by Opto22, was chosen for logging as its versatility allows any combination of input and output ports, each with its own predetermined gain. Logged data include 12 temperatures around the primary combustor loop; in-bed heat exchanger outlet

water temperature; pressures from the eight pressure taps along the riser; pressures from four pressure taps from the two orifice runs; and O<sub>2</sub>, CO, CO<sub>2</sub>, CH<sub>4</sub>, SO<sub>2</sub>, and NO<sub>x</sub> concentrations from the gas analysers. In addition, separate from the Optomux system, the loss in weight from the fuel and start-up hopper load cells is recorded directly from the digital display through an RS232 port on the computer. Control outputs are in the form of 4 to 20 mA signals to the primary, secondary, and L-valve valves, and the start-up and fuel hopper rotary valve controllers.

The data logging and control programs were written in Microsoft QuickBASIC and allow interactive operator interfacing for the turning-on or -off of various controllers, and changing of control parameters. The computer is able to completely log the system and perform all necessary calculations (of flowrates, solids loading, control action, etc.) in less than 2 seconds. A complete listing of the data logging and control programs written for this work are provided in Appendix C.

### **3.9 Fuels and Inerts Characteristics**

The fuels used in this work include Highvale coal, Conoco coke, and Syncrude Fluid coke. The proximate and ultimate analyses of these fuels is provided in Table 3.2. The choice of fuels was based on availability from other research on the pilot CFBC. Table 3.3 provides the size distribution of the Ottawa 50 silica sand employed as bed material throughout this work. The Sauter Mean Diameter of this sand is 259  $\mu\text{m}$ .

**Table 3.2: Proximate and Ultimate Analyses of Fuels**

Proximate Analyses wt.% (as recieved)				
	Highvale	Conoco	Syncrude	
	Coal	Delayed Coke	Fluid Coke	
Volatile Matter	30.5	10.0	7.2	
Fixed Carbon	42.1	84.0	84.6	
Ash	12.2	0.8	7.6	
Moisture	15.2	5.2	0.6	
Ultimate Analyses wt.% (dry basis)				
	Highvale	Conoco	Syncrude	
	Coal	Delayed Coke	Fluid Coke	
Carbon	62.4	87.3	80.1	
Hydrogen	3.6	3.8	1.9	
Nitrogen	0.8	1.9	2.0	
Chlorine	-	0.0	0.0	
Sulphur	0.2	4.9	7.1	
Oxygen (by difference)	18.7	1.1	1.3	
Ash	14.3	0.9	7.6	
Higher heating value				
(as received)	(MJ/kg)			
	24.0	32.9	30.0	



**Table 3.3: Screen Size Distribution of Ottawa 50 Silica Sand**

Screen Size		Mass Retained
(-)	(+)	%
TOP	35	2.7
35	45	25.5
45	50	14.6
50	660	19.7
60	80	28.1
80	120	8.1
120	PAN	1.2

## **CHAPTER 4 - MODEL DESCRIPTION**

### **4.1 Introduction**

This chapter summarizes a model which describes the dynamic behaviour of a circulating fluidized bed combustor. The purpose of the model is to provide an adequate simulator for the testing of control structures and algorithms. The model focuses on the prediction of the time variations of combustor temperature, outlet oxygen concentration, and the temperature of the water leaving the in-bed heat exchanger, arising from changes in operating parameters. Provided in this chapter are the major equations, methods, and assumptions used in this model.

The goal of research modeling is to explain observed behaviour based on first principles. In most cases it is not possible to entirely describe a process at this level. Empirical correlations and simplifying assumptions must generally be incorporated to complete the model or to make it solvable. These approximations are commonly justified by the final application for which the model is intended and by the complexity of the physical system under consideration. Often, particular aspects must be sacrificed in order to improve

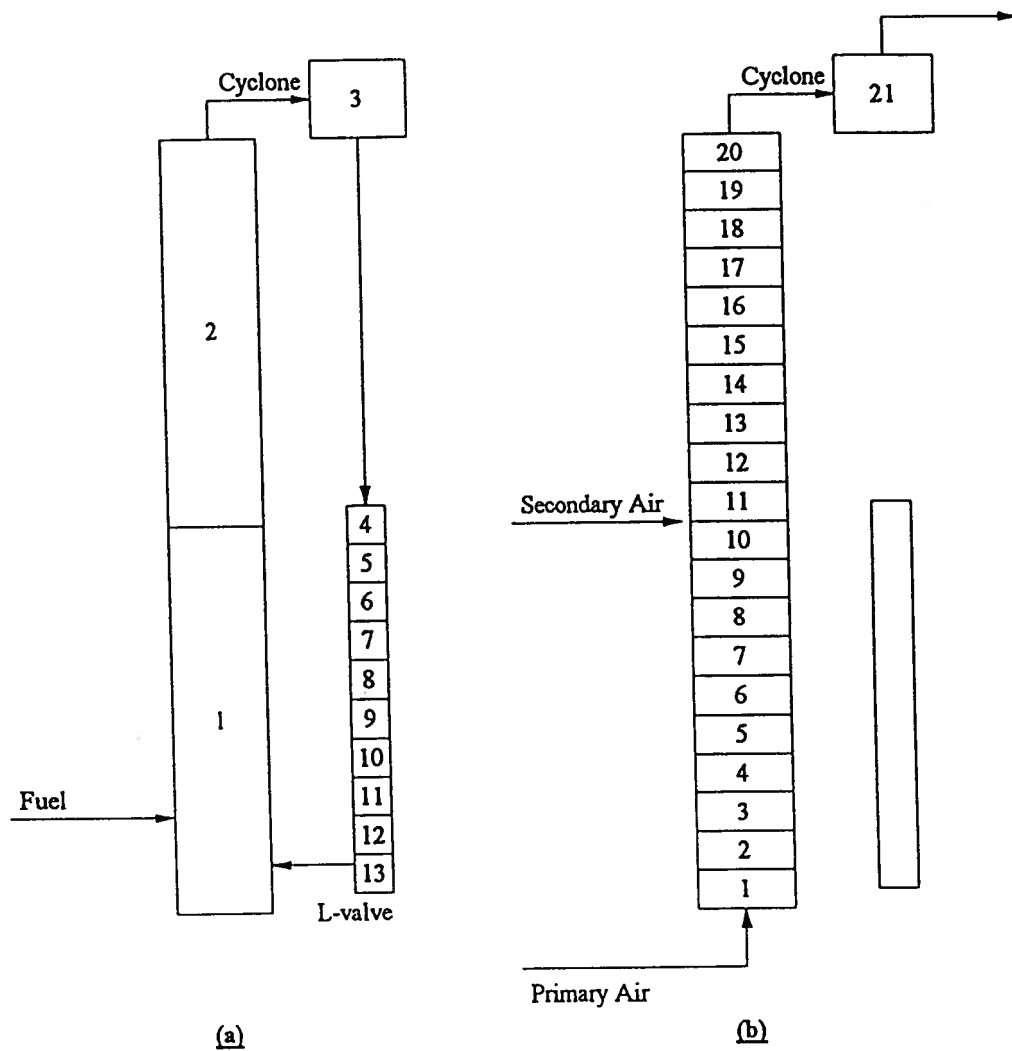
the reliability of the desired results (Hyppanen et al., 1993). Simulation of the dynamics of a circulating fluidized bed combustor is one case where simplifications are an absolute necessity.

Steady-state CFB models are far from complete. Although the qualitative behaviour is well understood, a generalized model has yet to be developed. Extensive research has been performed to investigate such aspects as core/annular structure, solids loading profile, and solids and gas residence time distributions. Results are difficult to compare. Little is known regarding the effects of parametric differences between various experimental units to allow confident comparison. Compounding this deficiency is the need to use fitting parameters in order to match models to various results. The need for fitted parameters illustrates the current state of CFB modelling. Combustion of solid particles is also still the focus of much research. Questions arise regarding such aspects as fuel physical characterization, devolatilization rates, internal/external combustion of char, transfer processes to and from the particles, and surface products of combustion.

Compounding the above complexities is the extension to the time domain. It is not generally possible, or necessary, to base transient models on complex steady state versions. Very often the objectives of the two types of models are different and assumptions, which are inappropriate for one, are both reasonable and necessary for the other. In order to maintain an acceptable level of accuracy, and yet obtain acceptable speed of solution, a number of simplifications are required.

#### **4.2 Mass Balance**

To write an overall mass balance, the unit is divided into numerous zones for both the solids and various gaseous species. This is necessary in order to avoid the computational burdens associated with a continuum approach. Figure 4.1 presents the divisions employed in this model. Optimistically, these zones represent the behaviour of the critical modeling components in as simple a form as possible while maintaining some essential characteristics of



**Figure 4.1: Modeling Zones for Material Balance in Primary Loop of the CFB**  
 (a) Solids; (b) Gas

fast fluidization and CFB combustion, such as extensive backmixing of solids and near plug flow of gas in the riser.

#### **4.2.1 Solids Distribution**

A number of advanced models are available describing gas and solids movement in a circulating fluidized bed. For the most part, these models are too cumbersome to solve on a real time, dynamic basis. As well, they often require prior knowledge (ie. measurements) from the system in order to obtain a full solution and, as such, are not suitable for predictive simulation.

These models generally attempt to describe the now well-accepted core/annulus structure predominant in circulating fluidized beds. It is this structure that makes the CFB both unique and practically advantageous, yet mathematically complex. The purpose of the proposed model is to predict the dynamics of the combustor temperature, flue gas oxygen concentration, and heat removed by the in-bed heat exchanger. In cases where the combustion time of a particle is large compared to the turnover time of the bed and the cycle time through the external loop, the specifics of the internal structure are not important to overall heat release. For these well backmixed systems the core/annulus nature need not be addressed in mathematical form, and it is sufficient to assume a uniform radial suspension density.

Critically, it has been assumed that the interchange of both gas and solids between the core and annular regions is fast enough that these two regions can be taken as well mixed at the average voidage determined at a given height. Since the empirical heat transfer correlations employed later are consistent with this assumption, and temperatures are experimentally found to be radially fairly uniform, this assumption appears to be justified in the current context.

Key to the description of the hydrodynamics of a circulating fluidized bed is the axial solids loading profile, in particular, the riser average and at the in-bed heat exchange surface,

as a function of the operating parameters (solids circulation rate, gas velocity, etc.). Senior and Brereton (1992) consider two zones in the riser - a developing zone at the bottom and a fully developed zone above. Their model concentrates on the developed core/annular region and particle interactions between the core and the annulus for the purpose of heat transfer calculations. Noting that heat transfer surfaces are typically located in the upper developed zone and that the developing region is too complex for detailed modelling, they simply estimate the developing zone solids loading profile with a fitted exponentially decreasing voidage profile correlated to experimental data. An iterative solution is employed where the height of the transition point is altered until specific "matching" criteria are satisfied. These criteria are that, at the transition, (i) the densities of the developing and developed regions are the same, and (ii) the core-to-wall mass transfer coefficients are the same.

The solution of their model is too numerically intensive for this application. As well, their core/annular solution does not satisfy the radial uniformity assumption employed in this model. Some of their basic concepts are used here in order to obtain a solution for the solids loading profile.

Senior and Brereton assume an exponentially decaying profile from the solids entry point to the top of the developing flow region given by:

$$\frac{\rho_z - \rho_f}{\rho_0 - \rho_f} = \exp(-\zeta(z - z_0)) \quad (4.1)$$

where:  $z$  = height (m)

$z_0$  = reference height (m)

$\rho_z$  = density at height  $z$  (kg/m<sup>3</sup>)

$\rho_f$  = density at top of profile (kg/m<sup>3</sup>)

$\rho_0$  = reference constant density at height  $z_0$  (kg/m<sup>3</sup>)

$\zeta$  = decay constant (m<sup>-1</sup>)

The points chosen by Senior and Brereton to fit the curve given by Equation (4.1) are the solids return point and the top of the so-called developing flow region. In the model proposed here, exponential decay is considered right to the top of the riser. This is a common assumption in current CFB modelling. It implies that there is a continuous decrease in the riser solids loading with height. Senior and Brereton (1992) define a reflection coefficient at the top of the riser to account for observed deviations from this assumption. Riser exit geometry has a strong effect on the ease with which solids can leave the riser. For example, the common abrupt (90°) exit has a tendency to cause particles to rebound back down the walls of the riser leading to the often observed increase in solids loading at the top of the riser. Their reflection coefficient stipulates what fraction of solids reaching the top of the riser return down the walls. Unfortunately, this coefficient is not only a function of unit geometry, but is also highly dependent on operating conditions. Rather than redefine this coefficient, the exponential decay to the exit is accepted and fitted to ensure appropriate solids loading over the heat exchange surface. Although creating some error in the axial distribution of solids, this assumption has very little effect on the mass balance considering the assumption of perfect mixing.

For the suspension density at the solids entry port, Senior and Brereton (1992) propose a relationship correlated to data obtained on the UBC combustor of the form:

$$(1 - \epsilon_r) = c_0 U^{*c_1} \left( \frac{G_s}{\rho_p (U - v_T)} \right)^{c_2} \quad (4.2)$$

where:  $c_0, c_1, c_2$  = fitted constants

$G_s$  = net solids flux (kg/m<sup>2</sup>.s)

$U$  = gas velocity (m/s)

$U^*$  = dimensionless gas velocity (Grace, 1982)

$v_T$  = particle terminal velocity (m/s)

$\varepsilon_r$  = voidage at solids return level  
 $\rho_p$  = particle density (kg/m<sup>3</sup>)

It has been found that, with solids circulation rates measured experimentally (Appendix B), Equation (4.2) greatly overpredicts the density at the base of the riser. Consequently, the net solids flux  $G_s$  has been replaced with an internal flux  $G_{sj}$  correlated to fit measured combustor data.

The suspension density in the upper portion of the riser is assumed to follow a continuation of the exponential curve from the developing region, with the exit solids loading experimentally correlated. This density is determined from the work of Patience et al. (1992) who propose a relationship for a slip factor (the ratio of interstitial gas velocity to particle velocity) that has been shown to be consistent with data in the hydrodynamically developed flow of the upper bed. This factor does not take into account the effects of exit geometry. It has been modified here to represent the high internal recirculation of solids resulting from the abrupt exit.

The slip factor,  $\phi$ , applied to an asymptotic limit of fully developed flow, is defined as the ratio of the superficial gas velocity,  $U_g$ , to particle velocity,  $V_p$ , i.e.:

$$\phi = \frac{U_g}{\varepsilon V_p} \quad (4.3)$$

where  $\varepsilon$  = voidage.

The particle velocity,  $V_p$ , is based on the solids circulation rate, i.e.:



$$V_p = \frac{G_s}{\rho_p(1-\varepsilon)} \quad (4.4)$$

The relationship for slip factor proposed by Patience et al. (1992) is given by:

$$\phi = 1 + \frac{5.6}{Fr} + 0.47Fr_t^{0.41} \quad (4.5)$$

where  $Fr$  is the Froude number based on superficial gas velocity:

$$Fr = \frac{U_g}{(gD_c)^{0.5}} \quad (4.6)$$

while  $Fr_t$  is the Froude number based on particle terminal velocity:

$$Fr_t = \frac{v_t}{(gD_c)^{0.5}} \quad (4.7)$$

where  $D_c$  is the riser diameter (m).

By substituting the relationship for the slip factor (Equation 4.5) and particle velocity (Equation 4.4) into the definition of slip factor (Equation 4.3), the voidage,  $\varepsilon$ , in the fully developed zone can be estimated. As noted earlier, Equation (4.5) does not take into account riser geometry. In order to account for the substantial internal mixing caused by the abrupt exit on the UBC pilot CFBC, an internal flux,  $G_{si}$ , has been substituted for  $G_s$  in Equation (4.4). This hypothetical internal flux is obtained from a correlation of data from the UBC pilot plant to provide limiting voidages appropriate to estimate upper bed voidages.

#### **4.2.2 Solids Balance Zones**

The zones used to describe the density profile do not correspond well to physical divisions in the combustor; there are better divisions from the perspective of defining solids mixing zones. For this purpose, the riser is divided into two regions in series as illustrated in Figure 4.1(a). The "primary" zone extends from the base of the riser to the secondary air injection port and includes the primary air port, fuel feed point, and L-valve solids return. The "secondary" zone continues from the secondary air injection port to the top of the riser.

It is more common to regard the riser as 3 distinct zones to portray the conventional sigmoidal profile (e.g. Wen and Chen, 1982). Three-zone models recognise a dense region at the bottom, fully developed flow at the top, and an intermediate transition zone. Typical operating conditions in the UBC pilot combustor result in no significant bottom dense bed. The tapered lower section (Chapter 3) is intended to create high velocities at the base to help move the solids up the riser (although Senior (1992) has shown that, under some conditions, it can increase the base density). Under most conditions, the tapered bottom section essentially eliminates the dense zone. Because of the lack of a significant dense region, this zone is neglected in this model, as by Senior and Brereton (1992), and only the developing and upper bed regions are considered.

The base of the riser is characterised by a relatively high solids loading and significant particle movement in all directions. Above this turbulent, or developing, region is the core/annular flow structure characterised by a large downflow of solids at the wall and a high interchange of particles between the core and the annulus. The gas velocity in the primary zone (above the tapered section) is typically 4 to 5 m/s, 1/2 to 2/3 of the velocity in the secondary zone leading to a high downward flux of solids. From a global perspective, the whole zone might be considered well mixed.

As with the primary zone, the secondary zone exhibits a relatively high degree of backmixing of solids. This region has a developed core/annulus structure throughout, with solids mixing accentuated by the presence of an abrupt exit at the top of the riser. As solids

strike the top of the riser, many particles rebound downward internally, both increasing the solids hold-up in the upper zone as well as enhancing the backmixing of solids. As with the primary zone, in view of the extensive downflow of particles, for the solids, a well mixed assumption appears to be reasonable for this part of the furnace.

To provide a full description of the solids mixing, the degree of solids mixing between the upper and lower zones must finally be described. Here, communication between the primary and secondary zone is assumed to occur in the upward direction only. Downflow across this zonal barrier is considered negligible. At the secondary air injection port there is a sudden increase in gas velocity from that of the primary zone. This addition of secondary air is considered an impediment, preventing solids downflow from the secondary to primary zone. Such a barrier has been observed in practice (Brereton and Grace, 1993), although it does not prevent some communication between the regions due to particles falling between the secondary air inlet ports.

All solids leaving the riser enter the primary cyclone, where they are separated from the gas. The cyclone is considered as having a 100% collection efficiency, thereby passing all solids to the return standpipe.

Gas and solids are assumed to be thoroughly mixed in the cyclone. Kang et al. (1989) correlated the average solids residence time in cyclones (for geometric ratios of cylinder to cone height of 1.8 to 5.0) from experimental measurements as:

$$\frac{\bar{t}_s}{t_g} = 0.032 \text{Re}_p^{0.43} \left( \frac{u_g - v_t}{v_t} \right)^{0.70} \left( \frac{\rho_p - \rho_g}{\rho_g} \right)^{0.42} \left( \frac{H}{H - h_c} \right)^{-1.76} \quad (4.8)$$

where:  $h_c$  = cyclone cylinder height (m)

$H$  = cyclone overall height (m)

$\text{Re}_p$  = particle Reynolds number  $(\rho_g u_g d_p / \mu_g)$

$t_g$  = mean gas residence time (s)

$t_s$  = mean solids residence time (s)

$u_g$  = inlet gas velocity (m/s)  
 $v_t$  = particle terminal velocity (m/s)  
 $\rho_g$  = gas density (kg/m<sup>3</sup>)  
 $\rho_p$  = solids density (kg/m<sup>3</sup>)

The mean gas residence time,  $\bar{t}_g$  was given by Leith and Licht (1972) as:

$$\bar{t}_g = \frac{1}{(ab)u_g} \left( V_s + \frac{V_{nl}}{2} \right) \quad (4.9)$$

where: (ab) = cross-sectional inlet area = height  $\times$  width (m<sup>2</sup>)  
 $V_{nl}$  = volume of cyclone at natural length (m<sup>3</sup>)  
 $V_s$  = annular volume above exit duct to middle of entrance duct (m<sup>3</sup>)

Figure 4.2 indicates the cyclone dimensions referred to in Equations (4.8) to (4.11).

$V_s$  can be calculated from:

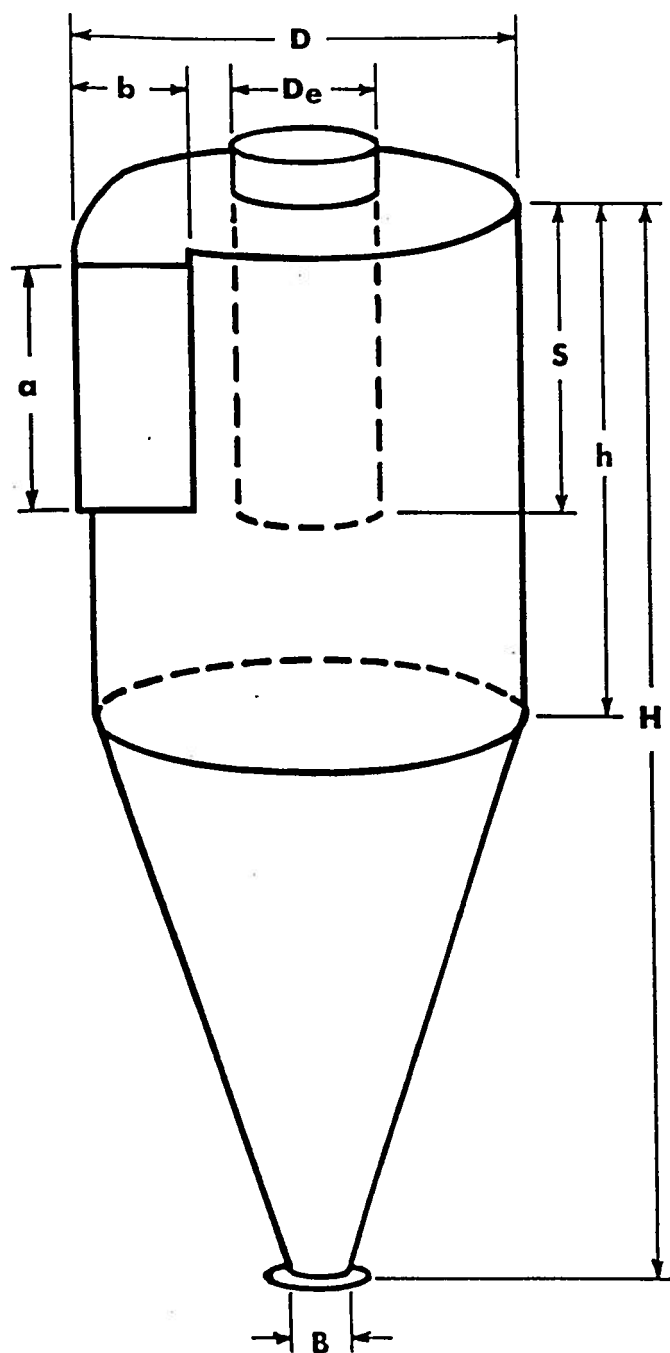
$$V_s = \frac{\pi \left( S - \frac{a}{2} \right) (D^2 - D_e^2)}{4} \quad (4.10)$$

while the volume at natural length,  $V_{nl}$ , can be determined from:

$$V_{nl} = \frac{\pi D^2}{4} (h_c - S) + \frac{\pi (1 + S - h_c) D^2}{4 \cdot 3} \left[ 1 + \frac{d}{D} + \frac{d^2}{D^2} \right] - \frac{\pi D_e^2 l}{4} \quad (4.11)$$

where  $d = D - (D - B) \left[ \frac{S + 1 - h_c}{H - h_c} \right]$ .

Assuming steady state conditions (ie. no accumulation of solids in the cyclone), the cyclone solids holdup,  $\rho_{cyc}$  (kg/m<sup>3</sup>), can be estimated from the solids residence time (Equation 4.8) and the mass flowrate of particles:



**Figure 4.2: Sketch Showing Cyclone Dimensional Parameters**  
 (from Leith and Licht, 1972)

$$\rho_{cyc} = \frac{\dot{m}_s \bar{t}_s}{V_c} \quad (4.12)$$

where the solids mass flowrate,  $\dot{m}_s$ , (kg/s), can be determined from the net solids flux as:

$$\dot{m}_s = G_s A_{ris} \quad (4.13)$$

and  $V_c$  is the cyclone volume excluding the volume inside the vortex-finder ( $m^3$ ).

The standpipe is simulated by a large number of CSTRs in series; sufficient to represent a plug flow system. The PFR is necessary to follow the gradual cooling behaviour of solids as they pass through the standpipe, as well as to dynamically represent the accumulation or consumption of the carbon in the total inventory as it is returned to the base of the combustion zone.

#### **4.2.3 Gas Balance Zones**

The gas is assumed to be in near plug flow in the riser. For simplicity, both the primary and secondary solids regions are each treated as ten well-mixed gas zones in series as shown in Figure 4.1(b). This gives a close approximation of plug flow. In practice, some gas backmixing occurs as gas is pulled downwards with the solids in the annular region (Yu et al., 1987). However, in combustors, this backmixing is small compared to the overall throughput of the system, and the model is relatively insensitive to the gas mixing assumption. The number of zones employed here is somewhat arbitrary, 10 being more than adequate to approach plug flow. The number of gas zones could be reduced in order to account for gas backmixing as a crude alternative to an axial dispersion model. In view of the structure of the dynamic model presented here, which leads to its apparent insensitivity to gas backmixing, the tanks-in-series approach satisfies the plug flow assumption while retaining the option (by

reducing the number of gas zones in series) of approximating higher degrees of axial gas dispersion.

It is assumed that there is negligible gas flow in the standpipe. Although air is introduced to the standpipe through the L-valve for solids circulation control, the amount is very small compared to the total throughput of the system (less than 1 %), and the contribution of reaction of this air with carbon in the standpipe is considered negligible.

#### **4.2.4 Combustion**

Superimposed upon the fluid mechanics are combustion reactions generating the heat release within the system. Different transient responses for different fuels must be reflected by different intrinsic combustion rates at this level of the model. The general approach here is to use small batch tests in a bench scale fluid bed reactor (Appendix A) to determine kinetic constants for a given char. Such a test is of relatively low cost and, because it is performed in a fluid bed environment, should provide combustion rates approximating those in the pilot plant. Provided that the same assumptions are applied to interpretation of the batch data as are used later in extrapolation of these data to the pilot, then appropriate variation of combustion rate with particle diameter and temperature should be achieved.

In this work, which is currently limited to solid fuels (coals and cokes), carbon combustion calculations are based on spherical char particles of a diameter representative of the system average. Sphericity is assumed for the sake of simplicity. Particle shape is not only fuel specific, but depends on fragmentation through devolatilization and collision, as well as on the mode of combustion. For example, a cubic particle burning under external diffusion control would have the highest local rate of combustion at the outer sharp edges where the boundary layer is thinnest. Eventually, in the extreme, it would burn itself into a nearly spherical particle. Therefore, initial particle shape is not necessarily indicative of the shape of the char in the bed.

Allowing for a non-uniform particle size distribution decreases the computational speed many-fold, while not necessarily improving accuracy. Rate of size reduction calculations are in any case approximate at best, being highly fuel specific and dependent on the mode of combustion. Swelling of the fuel, as well as fragmentation, creates uncertainties about the changing size distribution within the combustor. The actual combustion processes involving swelling, fragmentation and attrition, which result in a complex size distribution, are thus represented for kinetic purposes by a single effective mean diameter.

Devolatilization is treated as if it were instantaneous. It is assumed that combustion takes place on the surface of the particle and that all C consumed is converted to CO<sub>2</sub>. To reduce computational time, the particle temperature is taken as a conservative 100 °C above that of the combustion gas. Again, the unknown properties of the burning fuel (pore size distribution etc.) make any assumption regarding the extent of internal burning questionable. Measurements indicate particle burning temperatures in the range of 50 to 200 °C above that of the surroundings, the excess being a function of fuel type and, more importantly, size. In the case of the kinetic studies reported here (Appendix A), it was estimated that particle temperatures varied between 50 and 150 °C above the surroundings during the burnout period. The assumptions needed to apply a particle heat balance are not justifiable. In any case, errors in this assumption are corrected for by the choice of bed char characterisation (representative size) described above.

Following the development of Arena et al. (1991), the external diffusion constant is defined by the Sherwood number:

$$k_d = \frac{ShD}{d_c} \quad (4.14)$$

where: D = oxygen diffusivity (m<sup>2</sup>/s)  
d<sub>c</sub> = char particle diameter (m)



$k_d$  = diffusion constant (m/s)  
 $Sh$  = Sherwood number

The Sherwood number is taken as that developed for a bubbling fluidised bed by Chakraborty and Howard (1981):

$$Sh = 2\varepsilon + 0.69 \left( \frac{Re}{\varepsilon} \right)^{1/2} (Sc)^{1/3} \quad (4.15)$$

where:  $Sc$  = Schmidt number =  $\mu/\rho_g D$   
 $\varepsilon$  = suspension voidage

and  $Re$  is the particle Reynolds number, defined by Basu et al. (1987) for a circulating fluidized bed as:

$$Re = \frac{d_c \rho_g \left( \frac{U_g}{\varepsilon} - \frac{G_s}{\rho_s (1 - \varepsilon)} \right)}{\mu} \quad (4.16)$$

Equation (4.15) fully defines the external mass transfer coefficient based on the assumption that the external resistance is confined to char particles moving freely in a gaseous suspension. This is consistent with the assumption of thorough mixing, and ignores clustering or streamer formation.

The overall rate of combustion can be determined by equating the rate defined by kinetics to that defined by external diffusion to the char surface, i.e.:

$$r_c = k_c c_s^n = k_d (c_b - c_s) \quad (4.17)$$

where:  $c_b$  = bulk oxygen concentration (kmol/m<sup>3</sup>)  
 $c_s$  = surface concentration of oxygen (kmol/m<sup>3</sup>)  
 $k_c$  = kinetic rate constant based on external surface area ((kmol/m<sup>2</sup>·s)/(kmol/m<sup>3</sup>)<sup>n</sup>)  
 $n$  = order of reaction  
 $r_c$  = overall rate of combustion (kmol/m<sup>2</sup>·s)

The kinetic rate constant,  $k_c$ , can be expressed by the Arrhenius expression:

$$k_c = A e^{\frac{-E_a}{RT_p}} \quad (4.18)$$

where:  $A$  = frequency factor ((kmol/m<sup>2</sup>·s)/(kmol/m<sup>3</sup>)<sup>n</sup>)  
 $E_a$  = activation energy (kJ·mol<sup>-1</sup>)  
 $R$  = gas constant = 8.314 kJ·mol<sup>-1</sup>·K  
 $T_p$  = particle temperature (K)

Solution of Equation (4.17) for the surface concentration of oxygen,  $c_s$ , defines the current rate of combustion.

Only the reaction of O<sub>2</sub> with C is considered at this stage. Other reactions, such as those involving S or N, are ignored. Currently, of interest is the flue gas oxygen concentration and the amount of heat released - neither of which are computationally affected by the consideration of these secondary reactions. The amount of oxygen involved with these other reactions is negligible compared to the total throughput, as is their affect on the overall heat balance.

### **4.3 Heat Balance**

The hydrodynamic and combustion elements of the program combine to give the rate of heat release as a function of fluid mechanics, gas and fuel feedrates, and system temperature. To then solve for the critical system temperature and oxygen level requires simultaneous solution of the unsteady state heat transfer from the system.

The heat balance on the riser/cyclone section considers the heat of reaction (taking into account combustion and overall enthalpy change), heat removed by the in-bed heat exchanger (controllable by variation of the suspension density), heat lost to the refractory, and heat added to solids returning to the riser.

Unlike the mass balance in which the combustor is separated into numerous zones, the heat balance is made globally because, as noted experimentally, under typical steady state operation there are essentially no temperature gradients along the riser, while during periods of transients the gradients are never more than 10 to 15 °C. This is due to the high degree of solids backmixing which results in a high internal flux of heat throughout the system.

#### **4.3.1 Heat From Reaction**

The heat generated from combustion is determined from the heating value of the fuel and the amount of carbon consumed over the period  $\Delta t$  (as calculated from the mass balance). The heat leaving the reactor in the flue gas is determined from the enthalpies and mass balance for  $N_2$ ,  $O_2$ ,  $CO_2$ , and  $H_2O$ , from which the heat from the entering air is subtracted.

#### **4.3.2 Heat Removed by Heat Exchanger**

The water side heat transfer coefficient,  $h_{ti}$ , (for turbulent flow) is computed from Sleicher and Rouse (1975), i.e.:

$$Nu = \frac{2r_{ti}h_{ti}}{k_w} = 5 + 0.015Re_w^m Pr^n \quad (4.19)$$

where:

$$m = 0.88 - \frac{0.24}{4 + \text{Pr}} \quad (4.20)$$

$$n = 0.333 + 0.5e^{-0.6\text{Pr}} \quad (4.21)$$

$$\text{Re}_w = \frac{\rho_w Q_w}{2\pi\mu_w r_{ti}} \quad (4.22)$$

with:  $Q_w$  = water flowrate ( $\text{m}^3/\text{s}$ )  
 $r_{ti}$  = tube inside radius (m)  
 $\mu_w$  = water viscosity ( $\text{kg}/\text{m}\cdot\text{s}$ )  
 $\rho_w$  = water density ( $\text{kg}/\text{m}^3$ )

and:

$$\text{Pr} = \frac{C_{pw}\mu_w}{k_w} \quad (4.23)$$

where:  $C_{pw}$  = heat capacity of water ( $\text{J}/\text{kg}\cdot\text{K}$ )  
 $k_w$  = thermal conductivity of water ( $\text{W}/\text{m}\cdot\text{K}$ )  
 $\text{Pr}$  = Prandtl number

The gas-side heat transfer coefficient is determined from the work of Wu (1989) who makes the common assumption that the heat transfer coefficient is of the form:

$$h_{to} = h_{gc} + h_{pc} + h_{rad} \quad (4.24)$$

where:  $h_{gc}$  = gas convective component ( $\text{W}/\text{m}^2\cdot\text{K}$ )

$h_{pc}$  = particle convective component (W/m<sup>2</sup>·K)

$h_{rad}$  = radiative component (W/m<sup>2</sup>·K)

$h_{to}$  = heat transfer coefficient between outside of tube and combustor (W/m<sup>2</sup>·K)

Wu (1989) estimates the gas convective component,  $h_{gc}$ , from Sleicher and Rouse (1975). For this application it is taken as a constant at 13 W/m<sup>2</sup>·K.

The gas-solids suspension is treated as a grey body so that the radiative component is estimated from:

$$h_{rad} = \frac{\sigma(T_{susp}^4 - T_{surf}^4)}{(1/e_{susp} + 1/e_{surf} - 1)(T_{susp} - T_{surf})} \quad (4.25)$$

where:  $T_{surf}$  = tube surface temperature (K)

$T_{susp}$  = suspension temperature (K)

$e_{surf}$  = emissivity of tube surface

$e_{susp}$  = emissivity of suspension

$\sigma$  = Stefan-Boltzmann constant ( $5.672 \times 10^{-8}$  W/m<sup>2</sup>·K<sup>4</sup>)

Fitted values of  $e_{susp}$  and  $e_{surf}$  equal to 0.91 were used as in the work of Wu (1989).

The particle convective component,  $h_{pc}$ , can then be estimated from experimental data as a function of the average suspension density over the heat transfer surface. From the data of Wu (1989) it was determined, for the equipment used here, that:

$$h_{pc} = 2.0\rho_{ave,hx} \quad (4.26)$$

where  $\rho_{ave,hx}$  is the average suspension density around the heat exchange surface (kg/m<sup>3</sup>).

With these substitutions, Equation (4.24) becomes:

$$h_{to} = 13 + 2.0\rho_{ave,hx} + 4.735 \times 10^{-8} \left( \frac{T_g^4 - T_w^4}{T_g - T_w} \right) \quad (4.27)$$

The overall heat transfer coefficient,  $U_o$ , based on the combustion side is then:

$$\frac{1}{U_o} = \frac{r_{to}}{h_{ti}r_{ti}} + \frac{r_{to} \ln(r_{to}/r_{ti})}{k_{ss}} + \frac{1}{h_{to}} \quad (4.28)$$

If the tube wall resistance is neglected, and the inside and outside resistances together with their corresponding temperature differences are equated, it can be shown that the tube wall temperature can be estimated as:

$$T_w = T_g - \Delta T_o \quad (4.29)$$

where:

$$\Delta T_o = \frac{\Delta T_{LMTD}}{(h_{to}/U_o)} \quad (4.30)$$

with:  $\Delta T_{LMTD}$  = overall log mean temperature difference (K)  
 $\Delta T_o$  = temperature difference between gas phase and tube wall (K)

The procedure to determine the amount of heat removed by the in-bed heat exchanger is then as follows:

- (1) given water flowrate  $Q_w$ , compute  $h_{ti}$  from eq.(4.19);
- (2) guess water outlet temperature  $T_{waterout}$  and compute  $\Delta T_{LMTD}$ ;
- (3) guess tube wall temperature  $T_w$ ;
- (4) calculate  $h_{to}$  from eq. (4.27);

- (5) calculate  $\Delta T_o$  from eq. (4.30);
- (6) calculate  $T_w$  from eq. (4.29);
- (7) converged on  $T_w$ ? If not, go to (3); If yes, continue;
- (8) calculate  $U_o$  from eq. (4.28);
- (9) calculate heat transferred  $q_{hx} = U_o A_o \Delta T_{LMTD}$ ;
- (10) calculate outlet water temperature from  $q_{hx} = \dot{m} C_{pw} (T_{waterout} - T_{waterin})$ ;
- (11) converged on  $T_{waterout}$ ? If not, go to (2); If yes, calculation complete.

#### 4.3.3 Heat Loss to Refractory

The refractory comprises the riser and primary cyclone and is taken to have an equivalent hydraulic inside radius  $r_i$  and outside radius  $r_o$ . Physically, the riser is constructed (from the inside out) of 90 mm of Plicast LWI-20 Insulating Castable, 1 mm of Fiberfrax 970 J-Paper, 6 mm of carbon steel, 21 mm of Fiberfrax Durablanket S Ceramic Fiber Blanket, and 21 mm of Paroc Basalt Insulation. The thermal properties of these materials are listed in Table 4.1.

Heat loss through the refractory is treated as an unsteady state conduction problem in the radial direction with:

$$\frac{\partial T}{\partial t} = \frac{\alpha}{r} \frac{\partial}{\partial r} \left( r \frac{\partial T}{\partial r} \right) \quad (4.31)$$

where:  $T$  = material temperature at radius  $r$  and time  $t$  (K)

$r$  = radial position (m)

$\alpha$  = thermal diffusivity given by  $k/\rho C_p$  ( $m^2/s$ )

with boundary conditions:

$$-k_i \frac{\partial T}{\partial r} = h_i (T - T_g) \quad @ \quad r = r_i \quad (4.32)$$

**Table 4.1 - Thermal Properties of Materials of Construction**

<b>Material</b>	<b>Thermal Conductivity (W/m<sup>2</sup>·K)</b>	<b>Heat Capacity (J/kg·K)</b>	<b>Density (kg/m<sup>3</sup>)</b>
<b>Plicast Refractory*</b>	<b>0.3</b>	<b>920</b>	<b>961</b>
<b>Fiberfrax J-Paper*</b>	<b>0.112</b>	<b>1130</b>	<b>160</b>
<b>Carbon Steel#</b>	<b>34</b>	<b>473</b>	<b>7801</b>
<b>Fiberfrax Durablanket*</b>	<b>0.09</b>	<b>1130</b>	<b>64</b>
<b>Paroc Insulation*</b>	<b>0.08</b>	<b>920</b>	<b>128</b>
<b>Stainless Steel#</b>	<b>26</b>	<b>477</b>	<b>7900</b>

\* - from supplier

# - from R.H. Perry and D. Green (1984)



$$-k_o \frac{\partial T}{\partial r} = h_o (T - T_s) \quad @ \quad r = r_o \quad (4.33)$$

where:  $T_g$  = combustor inside temperature at time  $t$  (K)  
 $T_s$  = surroundings temperature (K)

The contact resistance between adjacent materials is ignored.

The external heat transfer coefficient,  $h_o$ , is assumed to be 12 W/m<sup>2</sup>K. Wu's expression for the inside heat transfer coefficient,  $h_i$ , as used for the in-bed heat exchanger (Equation 4.27), is assumed to apply to the refractory as well, with the temperature at the inside wall  $T_{r=ri}$  substituted for  $T_{tube}$  and the overall average solids loading used in place of  $\rho_{ave,hx}$ . Measurements of temperature at the inside wall of the combustor indicate that this expression provides an adequate estimation of the wall temperature. This value is approximately 20 °C below the combustion temperature at steady state conditions.

#### **4.3.4 Heat Added to Solids Returned to Riser**

In order to account for heat lost through the standpipe, a separate heat balance is performed to determine the temperature of the sand/ash/char mixture being returned to the base of the riser. The combustor compensates for this heat loss by transfer of heat from the gas to these solids. It is assumed that the gas and solids in the riser are at the same temperature.

Temperature variation within the sand particles is considered negligible. In order for this to be valid, Holman (1963) indicates the following condition must be met:

$$\frac{h(V/A)}{k} \leq 0.1 \quad (4.34)$$

where:  $A$  = particle surface area ( $\text{m}^2$ )  
 $h$  = convective heat transfer coefficient ( $\text{W}/\text{m}^2\cdot\text{K}$ )  
 $k$  = particle thermal conductivity ( $\text{W}/\text{m}\cdot\text{K}$ )  
 $V$  = volume of particle ( $\text{m}^3$ )

Note that the dimensionless group here is simply a Biot number.

Under typical operating conditions, a value of  $500 \text{ W}/\text{m}^2\cdot\text{K}$  has been estimated for  $h$ . The sand particles have a mean diameter of  $260 \mu\text{m}$  with a thermal conductivity,  $k$ , of  $0.33 \text{ W}/\text{m}\cdot\text{K}$  (Perry and Green, 1984). These provide an estimate of  $\frac{h(V/A)}{k}$  of 0.07 validating the assumption of temperature uniformity of the particles.

Equating convection heat loss from a particle to its decrease in internal energy:

$$hA(T - T_s) = -C_p \rho V \frac{dT}{dt} \quad (4.35)$$

with the initial condition of  $T = T_o$  @  $t = 0$ , the solution of Equation (4.35) is:

$$\frac{T - T_s}{T_o - T_s} = \exp \left( - \left( \frac{hA}{\rho C_p V} \right) t \right) \quad (4.36)$$

The density of the sand particles is  $2600 \text{ kg}/\text{m}^3$  with a specific heat,  $C_p$ , of  $820 \text{ J}/\text{kg}\cdot\text{K}$ . Considering the discretized time interval of 1 s used in this work, the right hand side of Equation (4.36) can be solved to yield  $\frac{T - T_s}{T_o - T_s} = 0.082$  indicating that 91.8% of the

temperature change of a particle takes place during one calculation period - justifying the assumption that all solids introduced to the combustor are instantaneously heated to combustion temperatures.

The same ten standpipe zones used in the mass balance are employed to estimate the heat loss to the surroundings (Figure 4.1(a)). Each individual zone is considered to be at a uniform temperature, with the particles entering the first zone (top of standpipe) at the combustor temperature. As the solids travel down the standpipe they are assumed to be well mixed in each consecutive zone in order to determine a new zonal average temperature, accounting for heat lost to the surroundings.

The standpipe heat losses are modelled in a similar fashion as the riser (section 4.3.3). In this case, the standpipe radially consists of 6 mm of stainless steel on the inside, 21 mm of Fiberfrax Durablanket S Ceramic Fiber Blanket, and 21 mm of Paroc Basalt Insulation. It is assumed that heat travels in the radial direction only, i.e. axial conduction of heat is neglected. This is consistent with the observation that the solids, having a very high heat capacity, tend to exhibit only a small drop in temperature as they pass down the standpipe (5 to 10 °C from top to bottom compared to the driving force to the atmosphere of 800 to 850 °C).

#### **4.4 Model Summary**

Computationally, the model is solved at each time step. This takes the form of a number of sequential steps. The order of computation is as follows:

(1) Given temperature  $T_g$  from previous time step, compute general properties

- (a) gas density
- (b) gas viscosity
- (c) gas thermal conductivity
- (d) gas molecular diffusivity
- (e) particle terminal velocity

(2) Perform hydrodynamic calculations. Given overall average suspension density, compute:

- (a) solids loading profile in riser (section 4.2.1).
  - (b) solids loading in cyclone (section 4.2.3).
- (3) Complete mass balance:
- (a) advance solids through standpipe and determine carbon content.
  - (b) for riser and cyclone:
    - (i) determine carbon content of solids zone i.
    - (ii) compute rate of combustion of gas zone j (section 4.2.4).
    - (iii) compute final concentrations of  $O_2$  and C in gas zone j.
    - (iv) redistribute carbon in solids zone i and repeat for next gas zone j.
- (4) Complete heat balance. This is an iterative procedure by which the heat generated is equated to the sum of the heat losses to determine a new combustor temperature  $T_g$ :
- (a) compute heat lost to refractory (section 4.3.3).
  - (b) compute heat lost through standpipe and temperature of solids re-entering riser (section 4.3.4).
  - (c) compute heat lost to solids introduced to combustor.
  - (d) compute heat generated from reaction and enthalpy balance (section 4.3.1).
  - (e) determine heat removed by heat exchanger (section 4.3.2).

The results of the above computation fully describe the condition of the combustor at time  $t$ . Most important in this application is the combustor temperature, water outlet temperature from the in-bed heat exchanger, and flue gas composition (especially the  $O_2$  concentration). This information permits the implementation of control structures through manipulation of fuel feed rate, riser solids loading and air flows.

#### **4.5 Simulation**

In order to provide full simulation capability, the model is extended to allow on-line user interface and graphic display updated at each time step. Variables which can be manipulated include:

- fuel feedrate
- total air flow
- primary/secondary air ratio
- number of tubes used in heat exchanger
- flowrate of water through heat exchanger
- setting of average riser solids loading
- setpoints for gas temperature, water temperature, and O<sub>2</sub> concentration
- various controller parameters as required

A complete program listing of the dynamic simulator is provided in Appendix D. Simulation results are presented, compared with experimental data from the UBC pilot combustor, and discussed in Chapter 7. The model is discussed in the latter part of the thesis as some of the results are in reference to the control identification and application work (Chapters 5 and 6).

## **CHAPTER 5 - PROCESS IDENTIFICATION AND CONTROL**

This chapter introduces basic concepts of control as used in this work. It is not a literature review, nor does it consider these techniques in detail. It is included to aid the reader who does not have a background in this material. It is intended to provide sufficient information for an understanding of the results presented in Chapters 6 and 7.

### **5.1 Introduction**

Being a highly competitive field, control design and philosophy tend to be proprietary. Therefore there is little public access to information on the application of process control to CFB combustion. Note also that there is growing pressure for tighter operating constraints due to expanding legislation on emissions and the consequential need to improve general overall efficiencies, and this further promotes secrecy among vendors and operators of commercial CFBC systems.

The controls portion of this research was necessary to complete the validation of the dynamic model described in Chapter 4. Dynamic tests were carried out to develop discrete time process models of the plant. As well, an implementation for control of the UBC pilot

CFB combustor was demonstrated. Discrete time models determined experimentally can be compared with those extracted through simulation for validation of the appropriateness of the dynamic model in control simulation. In addition, the application of control to the combustor provides a basis for comparison of the behaviour of a particular control system to that same system implemented through simulation. This comparison provides further qualitative validation of the simulator.

CFB combustion is a process which is highly time variant. Among the many disturbances are variations in fuel feed size and reactivity, and non-uniformities of bed material properties and composition. Added to these disturbances, in boiler application, is the continually changing load demand. The combination of these aspects leads to a process that requires continuous monitoring and incessant control action.

This chapter focuses on the control techniques used in this research. A brief overview of model-based control leads to a discussion of the control algorithm employed - Generalized Predictive Control (GPC). This algorithm is outlined to provide the reader with an understanding of its operation and its advantages. As well, the employed identification techniques are presented. Identification refers to plant testing and analysis for the development of a simplified "black box" model describing the dynamics of the process output(s) in response to changes in the process input(s).

Of the model-based approaches to control, the GPC algorithm has been proven to be the most robust for open-loop unstable or non-minimum phase plants, and varying or unknown model order or dead-time. The pilot CFB, as found through experience, is highly time variant. This is due to the deterioration of bed material over the course of an experiment as well as varying fuel properties. In addition, consideration was given to the effect of changing fuels. In this instance, the robustness to unknown model order and dead-time is considered highly advantageous. Although it is possible that other control techniques may have proven satisfactory, the use of GPC serves as (1) an applied demonstration of GPC itself

and (2) a demonstration of model-based control within the framework of a combustion control system.

Complete control of the combustor can be achieved by adjustments in total air flow, riser solids loading, and fuel feedrate to meet specifications in combustor temperature, flue gas oxygen concentration (emissions), and in-bed heat exchanger water outlet temperature (rate of heat removal). This is a  $3 \times 3$  MIMO problem in which interaction will occur. The choice of control loops was qualitative as outline by Stephanopoulos (1984):

(1) Choose the manipulation that has a direct and fast effect on a controlled variable.

(2) Choose the couplings so that there is little dead-time between every manipulation and the corresponding control variable.

(3) Select couplings so that interaction is minimal.

Applying the above criteria, it was decided to control heat exchanger outlet temperature with solids loading, flue gas oxygen with total air flow, and combustor temperature with fuel feedrate.

Interaction between these loops is somewhat minimized due to the differences in the time constants of each. The solids loading has a time constant of approximately 10 seconds, flue gas oxygen has a time constant of 3 to 5 minutes, while that of combustor temperature is 20 to 30 minutes. These differences, while not eliminating interaction, indicate that control of these loops should not lead to instability.

It is important to note that the purpose of the control portion of this research was as outlined above. It was not for determining the optimum control strategy of CFB boilers nor for optimal control of the UBC pilot CFB combustor. It was simply to define a strategy that was able to provide sufficient control for model validation as well as demonstrate the GPC model-based controller.



## **5.2 Overview of Control**

Process control has been an issue in many industries for hundreds of years. Beginning with manual control, as processes became more complex and required more attention, the natural progression to automatic control began. Notwithstanding the value of operator knowledge, computer-based control is a major facet of any project and continues to be a prime area of research.

The field of process control concerns itself with two conditions or modes of operation. These are the regulation problem or steady-state operation and the servomechanism problem or that of changing operating conditions. For regulation, the goal is to maintain the process at some given operating point and the purpose of the controller is to correct for disturbances in the process. In general, the tighter the control about a given condition, the more efficient that process becomes and the higher the quality of the product. The servo problem concerns itself with moving the process from one operating condition to another. For efficiency, it is generally desired that these changes take place as quickly as possible without imparting excessive disturbances to the process or producing excessive off-spec product.

Numerous control methods are under both extensive application and research. These range from simple proportional controllers to highly advanced artificially intelligent or fuzzy logic type controllers. The majority of modern control methods can be considered as being model-based from some point of view. Models are employed in the initial controller design and/or employed within the control structure. This allows complicated control problems to be solved faster and more easily, while such questions as stability and robustness can be addressed directly (Fisher, 1991).

In this thesis, the term "advanced" control refers to model-based control. More specifically, this is in reference to internal model type control. Internal Model Control (IMC) is a general class of control structure which includes Minimum Variance Control (MVC), Extended Horizon Control (EHC), Linear Quadratic Control (LQ), and Dynamic Matrix control (DMC).

IMC is based on a simplified model of the process. A block diagram illustrating IMC is presented in Figure 5.1, where  $G$  represents the actual process,  $G^m$  is the model describing the process dynamics, and  $G^*$  is the controller for the given loop.

The block diagram of Figure 5.1 can be reduced to equation form:

$$y = \left[ \frac{G^*G}{1 + G^*(G - G^m)} \right] y_{sp} + \left[ \frac{1 - G^*G^m}{1 + G^*(G - G^m)} \right] N \quad (5.1)$$

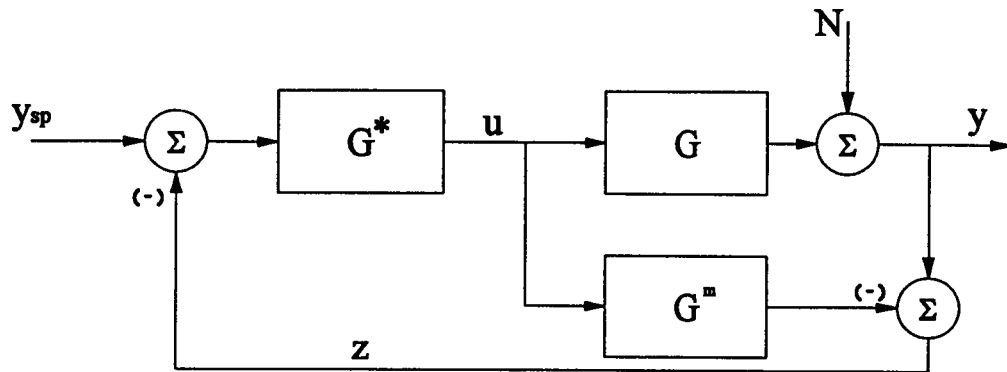
The error between the model and the actual process is given by  $(G - G^m)$ . If the model is perfect, this error is 0 and Equation (5.1) reduces to:

$$y = G^*G y_{sp} + (1 - G^*G^m)N \quad (5.2)$$

Ideally, one wishes to design a controller that can perfectly track the setpoint and completely reject all disturbances (so that  $y = y_{sp}$ ). This is accomplished by designing the controller such that  $G^*G = 1$  in Equation (5.2). Based on the assumption that the model is perfect ( $G^m = G$ ), the resulting controller is given by:

$$G^* = G^{m-1} \quad (5.3)$$

Equation (5.2) represents the loop of Figure 5.1 as an open loop relationship (since the assumption that  $G - G^m = 0$  eliminates the feedback loop). The only condition for closed-loop stability in this case is that  $G^*$  and  $G$  be open-loop stable. This is considerably simpler and more straightforward than methods required to analyse conventional control schemes (e.g. Routh, root locus, Bode, and Nyquist).



**Figure 5.1: Simplified Block Diagram Showing Internal Model Control (IMC) Structure**

In practice, this idealized controller is often impractical. Inevitably, there is some model mismatch (or modeling error), and the controller is extremely sensitive to this, as in the case with feedforward control. As well, the presence of noise or unmodeled disturbances can lead to excessive control action.

These two problems can be solved by the inclusion of a filter. A properly designed filter removes much of the noise and reduces controller activity. This is the equivalent of "de-tuning" in conventional control. It slows down the response of the controller, but can make the loop stable for any level of model error.

Another problem that can arise with this idealized controller is with the inversion of the process model ( $G^* = G^m{}^{-1}$ ). Time delays in the process appear as future values in the controller, and are impossible to realize in application. As well, right hand plane (RHP) zeros (in the s-domain) in the process model result in an unstable controller. When the model is inverted, the RHP zeros become RHP poles. From theory, RHP poles (or poles outside the unit circle in the z-domain) indicate instability. These two problems are solved by factoring the model and only inverting that part which does not contain RHP zeros or pure time delay.

An extension to the basic IMC algorithm is the inclusion of a predictor (Model Predictive Control (MPC)). The predictor estimates the trajectory of the process output to determine the next appropriate control action based on some cost function. This allows on-line optimization by minimization of the cost function while meeting constraints on the input and output variables.

Adaptive control is the technique by which process modeling and/or controller design is performed on-line. "In principle, any model-based design or control technique can be made into an adaptive system by adding a suitable on-line model identification algorithm" (Fisher, 1991). Self-tuning regulators (Aström and Wittenmark, 1973), self-tuning controllers and generalized minimum variance controllers (Clarke and Gawthrop, 1975, 1979), and pole placement controllers are examples of the implementation of adaptive control.

The combination of internal model, trajectory prediction, and adaptation, results in some of the most practical controllers. These are Long Range Predictive Control (LRPC) systems. "LRPC has been used in a number of successful applications and is recommended for those that require an adaptive controller for non-linear and/or time-varying processes" (Fisher, 1991). Generalized Predictive Control (GPC) is one of the more recent developments and belongs to this class of LRPC. It is rapidly gaining interest both in research and for industrial application.

### 5.3 Generalized Predictive Control

Generalized Predictive Control (GPC) (Clarke et al., 1987a, b) is an algorithm which is considered to be a "generalization of the previous multistep predictive algorithms" (McIntosh et al., 1991). It is based on a receding-horizon method, predicting the process output over several future time steps. This algorithm is considered "general" because of its ability to handle non-minimum phase, open loop unstable, variable or unknown dead time, and unknown order processes. A complete detailed description of GPC is provided by Clarke et al. (1987a, b).

In discrete time, the condition of the process is known only at the instant of sampling. Commonly in digital control, the output,  $y_t$ , is sampled at equispaced time intervals, the same points as those where control adjustments,  $u_t$ , are made. The linear difference equation to model the process is:

$$y_t - \delta_1 y_{t-1} - \dots - \delta_r y_{t-r} = \omega_0 u_{t-b} - \omega_1 u_{t-b-1} - \dots - \omega_s u_{t-b-s} \quad (5.4)$$

which describes the output,  $y_t$ , as some function,  $\delta(y)$ , of the previous known outputs and  $\omega(u)$  of the previous inputs.  $b$  refers to the number of periods of delay representing the plant dead-time from when a control action is performed and a response is seen in the output.

By definition the backward shift operator is given by:

$$q^{-1}y_t = y_{t-1} \quad (5.5)$$

Substituting the backward shift operator into Equation (5.4) yields:

$$y_t = \frac{(\omega_0 - \omega_1 q^{-1} - \dots - \omega_s q^{-s}) q^{-b}}{(1 - \delta_1 q^{-1} - \dots - \delta_r q^{-r})} u_t \quad (5.6)$$

or

$$y_t = \frac{\omega(q^{-1}) q^{-b}}{\delta(q^{-1})} u_t \quad (5.7)$$

which is the discrete form of the process transfer function. It is important to note that the orders and constants of the polynomials  $\omega(q^{-1})$  and  $\delta(q^{-1})$  are determined not only by the process dynamics, but also by the chosen sample/control interval.

The original formulation of GPC is based on a CARIMA (Controlled Auto-Regressive and Integrated Moving-Average) plant model:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \frac{C(q^{-1})\xi(t)}{\Delta} \quad (5.8)$$

where  $A(q^{-1})$ ,  $B(q^{-1})$ , and  $C(q^{-1})$  are polynomials of order  $n_a$ ,  $n_b$ , and  $n_c$  respectively, i.e.

$$A(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a} \quad (5.9)$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \dots + b_{n_b} q^{-n_b} \quad (5.10)$$

$$C(q^{-1}) = 1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc} \quad (5.11)$$

while  $y$  is the plant output and  $u$  is the plant input or manipulated variable;  $\xi$  is a normally distributed noise with variance  $\sigma_\xi^2$  (white noise).  $C(q^{-1})$  is in fact a model combination of all of the noise components within the process. If  $C(q^{-1})$  is known for the process, the predictor has optimality properties. Alternatively, as in this work,  $C(q^{-1})$  can be a fixed design polynomial,  $T(q^{-1})$ , becoming a filter for estimation to provide other properties such as robustness to unmodeled dynamics.

In Equation (5.8),  $\Delta$  is a differencing operator given by:

$$\Delta = 1 - q^{-1} \quad (5.12)$$

to account for the non-stationarity of disturbances often encountered in industry and inherently providing integral action to the controller to remove offset.

In order to derive a  $j$ -step ahead predictor for  $y(t+j)$ , one must define the Diophantine identity. Consider the stochastic process:

$$A(q^{-1})y(t) = \frac{C(q^{-1})\xi(t)}{\Delta} \quad (5.13)$$

The prediction of  $y(t+j)$  at time  $t$  is given by:

$$y(t+j) = \frac{C(q^{-1})\xi(t+j)}{\Delta A(q^{-1})} \quad (5.14)$$

The right hand side of Equation (5.14) contains terms  $\xi(t+j)$ ,  $\xi(t+j-1)$ , ...,  $\xi(t)$ ,  $\xi(t-1)$ , .... These terms can be separated into those known (past) values,  $\xi(t)$ ,  $\xi(t-1)$ , ..., and those unknown (future) values,  $\xi(t+1)$ ,  $\xi(t+2)$ , ...,  $\xi(t+j)$ :

$$\frac{C(q^{-1})}{\Delta A(q^{-1})} = E_j(q^{-1}) + \frac{q^{-j}F_j(q^{-1})}{\Delta A(q^{-1})} \quad (5.15)$$

Rearranging and substituting the fixed design polynomial,  $T(q^{-1})$ , for the process noise,  $C(q^{-1})$ , yields the Diophantine identity:

$$T(q^{-1}) = E_j(q^{-1})A(q^{-1})\Delta + q^{-j}F_j(q^{-1}) \quad (5.16)$$

where  $E_j$  and  $F_j$  are polynomials defined by  $A(q^{-1})$  and  $C(q^{-1})$  and the prediction interval  $j$ .  $F_j$  is the known part of the disturbance forecast and  $E_j$  represents those components in the future or the forecast error.

Multiplying the model (Equation 5.8) by  $E_j\Delta q^j$  results in:

$$E_jA\Delta y(t+j) = E_jB\Delta u(t+j-1) + E_jC\xi(t+j) \quad (5.17)$$

Replacing  $E_jA\Delta$  with the aid of Equation (5.16) yields:

$$y(t+j) = E_jB\Delta u^f(t+j-1) + E_jC\xi^f(t+j) + F_jy^f(t) \quad (5.18)$$

where the superscript,  $f$ , indicates filtering by the quantity  $1/T(q^{-1})$ .

The polynomial  $E_j$  is of degree  $(j-1)$ . Multiplying  $E_j$  by the noise term  $\xi^f(t+j)$  involves only future values of noise. By definition, the best prediction of white noise is 0 allowing this



term to be dropped for predictive purposes. In addition, replacing  $E_j B$  with  $G_j$  yields the prediction of  $y(t+j)$  at time  $t$ :

$$\hat{y}(t+j) = G_j \Delta u^f(t+j-1) + F_j y^f(t) \quad (5.19)$$

where  $G_j$  is the step response of the plant to time  $t+j$ .

The GPC approach to prediction is to consider a number of horizons over which to perform prediction. This is to say that a set of predictions is produced over a range of  $j$ . For  $j \geq b$ , the plant dead-time, certain assumptions are required about future control actions. It is these assumptions that provide the basis for GPC.

The common objective or cost function is of the form:

$$J = E \left\{ \sum_{j=N_1}^{N_2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_2} \lambda_j [\Delta u(t+j-1)]^2 \right\} \quad (5.20)$$

where:  $N_1$  = minimum horizon  
 $N_2$  = maximum horizon  
 $\lambda_j$  = control action weighting factor  
 $w$  = set-point

$N_1$  can be, and often is, taken as 1. If the plant dead-time,  $b$ , is known with confidence, then  $N_1$  can be taken as  $b$  in order to minimize computation.  $N_2$  is typically taken to contain the rise time of the plant or, more generally, to contain all of the significant response generated by the current control action. The control weighting sequence,  $\lambda_j$ , allows for a user specified weighting of the control actions to be performed.

Since the minimization of the objective function (Equation 5.20) is in terms of unfiltered inputs,  $\Delta u$ , and not the filtered inputs,  $\Delta u^f$ , of the output prediction (Equation 5.19),

modification of the output prediction is required. Clarke et al. (1987b) define the following identity:

$$G_j(q^{-1}) = G'_j(q^{-1})T(q^{-1}) + q^{-j}\Gamma_j(q^{-1}) \quad (5.21)$$

where  $G'_j$ , and  $\Gamma_j$  are uniquely defined given  $G_j$ ,  $T$ , and  $j$ .  $G'_j$  contains the coefficients of  $G_j$  for which the initial identity of Equation (5.16) has  $T = 1$ .

Substituting for  $G_j$  in the output prediction, Equation (5.19), yields:

$$\hat{y}(t+j) = G'_j\Delta u(t+j-1) + \Gamma_j\Delta u^f(t-1) + F_jy^f(t) \quad (5.22)$$

In vector form, Equation (5.22) for each prediction interval,  $j$ , becomes:

$$\hat{y} = G'\tilde{u} + f \quad (5.23)$$

where:

$$\hat{y} = \begin{bmatrix} \hat{y}(t+1) \\ \hat{y}(t+2) \\ \vdots \\ \hat{y}(t+N) \end{bmatrix}, \quad \tilde{u} = \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N-1) \end{bmatrix}, \quad f = \begin{bmatrix} F_1y^f(t) + \Gamma_1\Delta u^f(t-1) \\ F_2y^f(t) + \Gamma_2\Delta u^f(t-1) \\ \vdots \\ F_Ny^f(t) + \Gamma_N\Delta u^f(t-1) \end{bmatrix}, \quad \text{and}$$

$$G' = \begin{bmatrix} g'_0 & 0 & \cdots & 0 \\ g'_1 & g'_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g'_{N-1} & g'_{N-2} & \cdots & g'_0 \end{bmatrix}$$

The expectation of Equation (5.20) can be written:

$$J = E\{(y - w)^T(y - w) + \lambda \tilde{u}^T \tilde{u}\} \quad (5.24)$$

where  $w$  is the setpoint vector:

$$w = \begin{bmatrix} w(t+1) \\ w(t+2) \\ \vdots \\ w(t+N) \end{bmatrix}$$

or:

$$J = E\{(G'\tilde{u} + f - w)^T(G'\tilde{u} + f - w) + \lambda \tilde{u}^T \tilde{u}\} \quad (5.25)$$

The minimization of  $J$  results in the projected control vector:

$$\tilde{u} = (G'^T G' + \lambda I)^{-1} G'^T (w - f) \quad (5.26)$$

The first element of  $\tilde{u}$  is applied to the process and the vector is re-calculated at each sampling instant.

The originators of this control algorithm (Clarke et al., 1987a, b) provide a comprehensive description of the controller including a complete simulation study, effect of controller settings, effect of model parameter assumptions, and comparison with other control approaches. This single paper is sufficient to fully comprehend and implement this control

scheme as well as overwhelm the reader with the adroitness of applicability and significance of the scheme. Further references are given by Fisher (1991).

#### **5.4 Adaptive Control**

One key feature of a CARIMA model in the process controller is the ease with which this form of model may be updated on-line. Adaptation involves the measurement of process input and output and recursive updating of the model parameters as information becomes available. The use of a CARIMA model minimizes the number of parameters and therefore reduces the complexity of computations required to perform the control task.

Inclusion of adaptability into a control scheme should be considered for processes which exhibit non-linearity or are time-variant. Ideally, as the process changes over time or as new operating conditions are imposed, the self-tuner adjusts the model parameters to maintain improved control. Although not always necessary for robustness, proper adaptation ensures the best possible tuning of the controller at any given point in time.

Recursive Least Squares (RLS) is the most common form of parameter estimation in use. This recursive form of the well known least squares method commonly updates the model parameters at each time instant the system is sampled (some versions provide decision criteria for whether or not to update the parameters at any given time).

Consider the CARIMA model used in the GPC controller:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \frac{C(q^{-1})\xi(t)}{\Delta} \quad (5.27)$$

Taking a best approximation of the white noise to be zero, Equation (5.27) becomes:

$$A(q^{-1})\Delta y^f(t) = B(q^{-1})\Delta u^f(t-1) \quad (5.28)$$

Equation (5.28) can be rewritten to express the model in terms of parameter and data vectors as:

$$\Delta y^f(t) = \theta^T \phi(t) \quad (5.29)$$

where:

$$\theta^T = [a_1, a_2, \dots, b_0, b_1, \dots] \quad (5.30)$$

and

$$\phi(t)^T = [\Delta y^f(t-1), \Delta y^f(t-2), \dots, \Delta u^f(t-1), \Delta u^f(t-2), \dots] \quad (5.31)$$

This expression separates the model parameters and the input/output information, and is suitable for identification (estimation of  $\theta^T$ ).

RLS employs the cost function:

$$J_N = \frac{1}{N} \sum_{t=1}^N \lambda^{N-t} \left\{ \Delta y^f(t) - \hat{\theta}^T \phi(t) \right\}^2 \quad (5.32)$$

where  $\lambda$  is a forgetting factor which takes a value between 0 and 1. This allows for weighting of the data with more emphasis placed on recent information as  $\lambda \rightarrow 0$ .  $\hat{\theta}^T$  is the model estimate of the true process. Minimization of Equation (5.32) produces the parameter estimates which minimize the difference between the process output  $y(t)$  and that estimated by the model  $\hat{\theta}^T \phi(t)$ .

The basic RLS is a minimization of Equation (5.32) with the forgetting factor,  $\lambda$ , equal to 1. Recursively:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)[y(t) - \hat{\theta}(t-1)^T \phi(t)] \quad (5.33)$$

$$K(t) = P(t-1)\phi(t) / [\lambda + \phi(t)^T P(t-1)\phi(t)] \quad (5.34)$$

$$P(t) = \frac{1}{\lambda} \left[ P(t-1) - \frac{P(t-1)\phi(t)\phi(t)^T P(t-1)}{\lambda + \phi(t)^T P(t-1)\phi(t)} \right] \quad (5.35)$$

The basic RLS as outlined above suffers from its inability to remain alert. Alertness refers to the speed with which an algorithm is able to adapt. The inability to remain alert results from the tendency of the covariance matrix,  $P(t)$ , to become small. Note that  $P(t)$  is always positive or zero. Referring to Equation (5.35), the updated covariance matrix is always less than or equal to the previous  $P(t-1)$  (remembering that the basic RLS employs  $\lambda = 1$ ). As this covariance matrix becomes smaller, so also does the gain matrix, Equation (5.34), resulting in less sensitivity of the update law, Equation (5.33), to modeling errors ( $\Delta y^f(t) - \hat{\theta}^T \phi(t)$ ).

Use of the forgetting factor,  $\lambda$ , is one method of improving the alertness of the algorithm. However, the choice of  $\lambda$  does not just affect the reliability of the estimated parameters. A small  $\lambda$  leads to heavy weighting on recent data resulting in a large  $P$  and  $K$ , leading to uncertainty in the parameter estimates. Large values of  $\lambda$  lead to slow tracking (as illustrated previously) and make fast changing parameters difficult to follow.

Fortescue et al. (1981) modified RLS to employ a variable forgetting factor and maintain boundedness on the covariance matrix  $P$ . They proposed a time-variable forgetting factor such that  $\lambda(t)$  approaches 1 when the prediction error is small and approaches a small value ( $< 1$ ) for a large prediction error, i.e.:

$$K(t) = P(t-1)\phi(t) / [1 + \phi(t)^T P(t-1)\phi(t)] \quad (5.36)$$

$$\lambda(t) = 1 - \frac{[y(t) - \hat{\theta}(t-1)^T \phi(t)]^2}{\sigma(1 + \phi(t)^T P(t-1)\phi(t))} \quad (5.37)$$

To avoid exponential growth of the covariance matrix (Cordero and Mayne, 1981), bounds must be imposed on  $P(t)$ :

$$P(t) = W(t)/\lambda(t) \quad (5.38)$$

if  $W(t)/\lambda(t) \leq C$ . Otherwise:

$$P(t) = W(t) \quad (5.39)$$

Here:

$$W(t) = P(t-1) - K(t)\phi(t)^T P(t-1) \quad (5.40)$$

The program written to perform the GPC calculations and the adaptive algorithm as outlined above is listed in Appendix C.

## **5.5 Process Identification**

Process identification is the procedure of determining an appropriate model for use within the controller. The model is a dynamical representation of the process in terms of its input and output variables. In the present thesis, the GPC controller employs a CARIMA

model (Equation 5.8). It is the goal of identification to determine the orders of polynomials  $A(q^{-1})$ ,  $B(q^{-1})$ , and  $C(q^{-1})$  as well as the corresponding coefficients.

Two modeling approaches can be taken for process identification - mechanistic or empirical modeling (see Figure 1.1). As one of the purposes of this research was the development of a mechanistic model, process identification for the control aspects was carried out using the empirical approach to provide further validation of the mechanistic model used in the simulator. The empirical approach uses data collected directly from the process both to determine the form of the model and to estimate the parameters. This approach results directly in a linear dynamic model of the plant in discrete time.

The subgroup of models referred to as empirical models may be further subdivided into parametric and non-parametric models. The empirical approach to modeling involves identification of the dynamic characteristics directly from plant data. The non-parametric model is presented as a plot or tabulation of the data (Bode, Nyquist, etc.), while the parametric model is a black box representation of the process (CARIMA model, transfer function), with parameters estimated from the data.

Non-parametric modeling is classical in nature. The response, in its chosen form, is compared with known response order shapes, and the appropriate parameters are estimated from the response curve to provide a crude estimation of a parametric model. A simple example is that of step testing. One perturbs the plant with a step change in the input and compares the output to a first order, first order plus dead-time, and second order response. Having determined the form of model from the above test, one estimates the dead-time, time constant, and/or damping factor as appropriate for the chosen model form. This example is one of the crudest methods of obtaining a model from plant data, but illustrates the use of the information collected.

Most non-parametric methods have resulted from continuous time analysis. However, the parametric methods discussed here have arisen from discrete time data analysis. These modern parametric methods are much more efficient in their use of available information in



that all data are taken into account. This is because the model not only fits a process transfer function but also identifies a process disturbance model. By modeling the noise in the output, disturbances are removed from the data and a more accurate estimation of the process is possible. These approaches are more recent and emerge from statistical time series literature (e.g. Box and Jenkins, 1970; Aström and Bohlin, 1966).

There are numerous techniques for both testing and analyzing results. In this work, process identification was achieved through open loop PRBS testing and time series analysis.

### **5.5.1 Plant Testing**

The method of plant testing used in this work involved dynamic stimulation of the open-loop process through the use of a pseudo-random binary sequence (PRBS). An example of a PRBS signal can be found in Figure 6.31. Upper and lower limits are placed on the input variable so that, at any given point in time, the input is either "high" or "low". At pre-defined instances, a decision is made whether to change the value of the input variable or to have it remain at the current value. The period between these decisions is defined as the switching period ( $T_{\text{switch}}$ ). At every  $T_{\text{switch}}$ , a random number generator supplies a 50% probability of a switch occurring.  $T_{\text{switch}}$  is chosen by rule of thumb as  $0.5\tau \leq T_{\text{switch}} \leq \tau$  where  $\tau$  is the effective time constant of the process (MacGregor and Taylor, 1990). This choice of switching interval helps to ensure that the frequency of the input is close to the break frequency of the process.

### **5.5.2 Data Analysis**

The steps involved in identification can be summarized (MacGregor and Taylor, 1990) as:

- (1) Auto- and cross-correlate input and output.
- (2) Estimate impulse response function.
- (3) Determine order of transfer function.

- (4) Fit transfer function.
- (5) Auto- and partial-correlate residuals.
- (6) Determine order of noise model.
- (7) Simultaneously fit transfer function and noise models.
- (8) Perform diagnostic checking.

Details on these steps are provided below.

One discrete representation of a process is by a weighted function of all past inputs:

$$y_t = v(Z^{-1})u_t \quad (5.41)$$

where  $v(q^{-1})$  is the impulse response which may be estimated directly from the process by forcing the process with an impulse input and measuring the output. This approach is not taken in this work.

The disadvantage of employing the impulse response function as the control model is that it tends to be of relatively high order. This makes it computationally cumbersome both in control calculations and for adaptive updating of coefficients.

A more common representation of the process is via the discrete transfer function model as shown in Equation (5.7). As many processes can be approximated by low-order transfer function models, the orders of the polynomials  $\omega(q^{-1})$  and  $\delta(q^{-1})$  are most often  $\leq 3$ . Note that the relationship between the impulse response function and the transfer function is:

$$v(q^{-1}) = \frac{\omega(q^{-1})q^{-b}}{\delta(q^{-1})} \quad (5.42)$$

As indicated by the CARIMA model used in the GPC controller (Equation 5.8), and as stated previously when discussing time series analysis in process identification, modeling of

the process noise is also of concern. This noise takes the form of an additive term appended to the process output. In transfer function form:

$$y_t = \frac{\omega(q^{-1})q^{-b}}{\delta(q^{-1})}u_t + N_t \quad (5.43)$$

where  $N_t$  is the noise term which is a combination of all noise components within the process.

The noise is commonly modeled as an Autoregressive - Integrated - Moving Average (ARIMA) process, so that:

$$N_t = \frac{\theta(q^{-1})}{\phi(q^{-1})\Delta^d}a_t \quad (5.44)$$

where  $\theta(q^{-1})$  is the moving average polynomial of order  $q$ ,  $\phi(q^{-1})$  is the autoregressive polynomial of order  $p$ , and  $\Delta^d$  is the backward difference operator which accounts for non-stationarity in the process;  $a_t$  is white noise and is the driving force of the system noise. A complete discussion of ARIMA models is provided by Box and Jenkins (1970).

#### **5.5.2.1 Transfer Function Structure**

As noted above, a necessary step of the fitting process is to determine an appropriate transfer function model order. An initial estimate of the order can be obtained by using correlation analysis of the data to obtain an impulse response function of the process. This function can be compared to transfer functions of known orders to select the appropriate model structure.

Consider the impulse response function model:

$$y_t = v(q^{-1})u_t + N_t \quad (5.45)$$

or:

$$y_t = v_0 u_t + v_1 u_{t-1} + v_2 u_{t-2} + \dots + N_t \quad (5.46)$$

Multiplying Equation (5.46) by  $u_{t-k}$  yields:

$$u_{t-k} y_t = v_0 u_{t-k} u_t + v_1 u_{t-k} u_{t-1} + v_2 u_{t-k} u_{t-2} + \dots + u_{t-k} N_t \quad (5.47)$$

The autocovariance of a given series,  $z_t$ , at lag  $k$  is given by:

$$C_{zz}(k) = \frac{1}{N} \sum_{t=1}^{N-k} (z_t - \bar{z})(z_{t+k} - \bar{z}) \quad k=0, 1, 2, \dots, K \quad (5.48)$$

where:  $C_{zz}(k)$  = autocovariance of series  $z$  at lag  $k$

$N$  = number of data points in the series

$\bar{z}$  = mean of series  $z_t$

It is more convenient to work with a normalized autocovariance - or autocorrelation function:

$$r_{zz}(k) = \frac{C_{zz}(k)}{C_{zz}(0)} \quad (5.49)$$

where:  $r_{zz}(k)$  = autocorrelation of series  $z$  ( $-1 \leq r_{zz}(k) \leq 1$ )

$C_{zz}(0)$  = variance of series  $z_t$

$$\begin{aligned}
C_{wz}(k) &= \frac{1}{N} \sum_{t=1}^{N-k} (w_t - \bar{w})(z_{t+k} - \bar{z}) \\
&= \frac{1}{N} \sum_{t=1}^{N+k} (w_{t+k} - \bar{w})(z_t - \bar{z})
\end{aligned} \tag{5.50}$$

Normalization of Equation (5.50) gives the crosscorrelation function:

$$r_{wz}(k) = \frac{C_{wz}(k)}{S_w S_z} \tag{5.51}$$

where  $S_w, S_z$  = standard deviation of series  $w_t$  and  $z_t$  ( $S_x = \sqrt{C_{xx}(0)}$ )

Considering the data set over  $t = 0, 1, 2, \dots, N$ , we may write Equation (5.47) in terms of the variance functions as:

$$C_{xy}(k) = v_0 C_{xx}(k) + v_1 C_{xx}(k-1) + v_2 C_{xx}(k-2) + \dots + C_{xN}(k) \quad k = 0, 1, 2, \dots, K \tag{5.52}$$

By definition, the input,  $x_b$ , and the noise,  $N_b$ , are uncorrelated so that  $C_{xN}(k) = 0$  for all  $k$ .

In matrix form, Equation (5.52) becomes:

$$C_{xy} = T_{xx} V \tag{5.53}$$

$$\text{where: } C_{xy} = \begin{bmatrix} C_{xy}(0) \\ C_{xy}(1) \\ \vdots \\ C_{xy}(K) \end{bmatrix}, \quad T_{xx} = \begin{bmatrix} C_{xx}(0) & C_{xx}(1) & \dots & C_{xx}(K) \\ C_{xx}(1) & C_{xx}(0) & \dots & C_{xx}(K-1) \\ \vdots & \vdots & \ddots & \vdots \\ C_{xx}(K) & C_{xx}(K-1) & \dots & C_{xx}(0) \end{bmatrix}, \quad \text{and } V = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_K \end{bmatrix}$$

One of the reasons that a PRBS input sequence is advantageous is that randomness guarantees that it is not autocorrelated so that:

$$C_{xx}(0) = S_x^2 \quad (\text{variance of } x) \quad (5.54)$$

and:

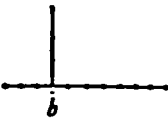
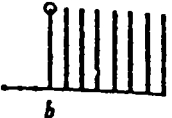
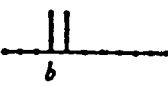
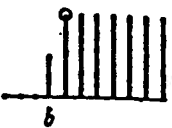
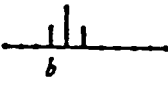
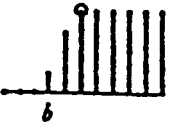
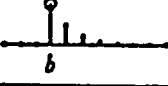
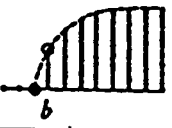
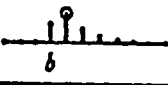
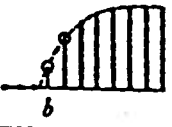
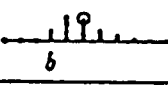
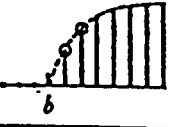
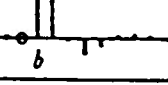
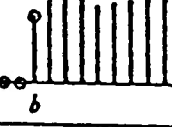
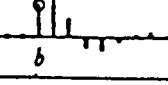
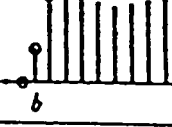
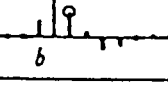
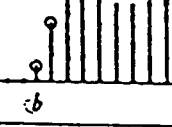
$$C_{xx}(k = 1 \dots K) = 0 \quad (5.55)$$

Thus the impulse weights can be determined by:

$$v_k = r_{xy}(k) \frac{S_y}{S_x} \quad (5.56)$$

Note that for Equation (5.56) to be valid, necessary conditions are that  $x_t$ ,  $y_t$ , and  $N_t$  are stationary with zero mean, that  $x_t$  is uncorrelated with  $N_t$ , and that  $x_t$  is uncorrelated (random).

Figure 5.2 presents a table from which the impulse responses of Equation (5.56) can be compared with transfer function models to choose an appropriate transfer function structure.

$r, s, b$	$\nabla$ Form	$B$ Form	Impulse Response $v_j$	Step Response $V_j = \sum_{i=0}^j v_i$
003	$Y_t = X_{t-3}$	$Y_t = B^3 X_t$		
013	$Y_t = (1 - .5\nabla) X_{t-3}$	$Y_t = (.5 + .5B) B^3 X_t$		
023	$Y_t = (1 - \nabla + .25\nabla^2) X_{t-3}$	$Y_t = (.25 + .50B + .25B^2) B^3 X_t$		
103	$(1 + \nabla) Y_t = X_{t-3}$	$(1 - .5B) Y_t = .5B^3 X_t$		
113	$(1 + \nabla) Y_t = (1 - .5\nabla) X_{t-3}$	$(1 - .5B) Y_t = (.25 + .25B) B^3 X_t$		
123	$(1 + \nabla) Y_t = (1 - \nabla + .25\nabla^2) X_{t-3}$	$(1 - .5B) Y_t = (.125 + .25B + .125B^2) B^3 X_t$		
203	$(1 - .25\nabla + .5\nabla^2) Y_t = X_{t-3}$	$(1 - .6B + .4B^2) Y_t = .8B^3 X_t$		
213	$(1 - .25\nabla + .5\nabla^2) Y_t = (1 - .5\nabla) X_{t-3}$	$(1 - .6B + .4B^2) Y_t = (.4 + .4B) B^3 X_t$		
223	$(1 - .25\nabla + .5\nabla^2) Y_t = (1 - \nabla + .25\nabla^2) X_{t-3}$	$(1 - .6B + .4B^2) Y_t = (.2 + .4B + .2B^2) B^3 X_t$		

**Figure 5.2: Impulse Response Functions of Various Transfer Function Models**

(from MacGregor and Taylor, 1990)

Having identified a tentative structure for the transfer function, an initial estimate of the transfer function coefficients can be obtained from Equation (5.42). By equating coefficients of  $B^j$ , we obtain:

$$\begin{aligned}
 v_j &= 0 & j < b \\
 v_j &= \delta_1 v_{j-1} + \delta_2 v_{j-2} + \cdots + \delta_r v_{j-r} + w_0 & j = b \\
 v_j &= \delta_1 v_{j-1} + \delta_2 v_{j-2} + \cdots + \delta_r v_{j-r} - w_{j-b} & j = b+1, b+2, \dots, b+s \\
 v_j &= \delta_1 v_{j-1} + \delta_2 v_{j-2} + \cdots + \delta_r v_{j-r} & j > b+s
 \end{aligned} \tag{5.57}$$

### 5.5.2.2 Noise Model Structure

A similar approach provides an initial structure for the noise model (Equation 5.44). The process noise can be estimated from Equation (5.43):

$$N_t = y_t - \frac{\omega(q^{-1})q^{-b}}{\delta(q^{-1})} u_t \tag{5.58}$$

The autocorrelation and partial autocorrelation of these residuals can be compared with known noise model structures in a similar fashion as the comparison of the impulse response to transfer function models in Section 5.5.2.1.

The partial autocorrelation is a correlation between points at lag  $k$  after accounting for all correlation effects for lags  $< k$ . Box and Jenkins (1970) fully describe the determination of the partial autocorrelation of a data set concluding with the general recursive formulae developed by Durbin (1960):

$$\varphi_{p+1,j} = \varphi_{p,j} - \varphi_{p+1,p+1} \varphi_{p,p-j+1} \quad j = 1, 2, \dots, p \tag{5.59}$$



$$\varphi_{p+1,p+1} = \frac{r_{p+1} - \sum_{j=1}^p \varphi_{p,j} r_{p+1-j}}{1 - \sum_{j=1}^p \varphi_{p,j} r_j} \quad (5.60)$$

### 5.5.2.3 Parameter Fitting

Having estimated the structures of the transfer function and noise model, the parameters must be determined. Consider the model equation:

$$y_t = \frac{\omega(q^{-1})}{\delta(q^{-1})} u_{t-b} + \frac{\theta(q^{-1})}{\varphi(q^{-1}) \Delta^d} a_t \quad (5.61)$$

In fitting the model parameters, the goal is to minimize the residuals (unfit data). This is accomplished by minimizing the sum of squares of the driving force of the noise (minimize  $\sum a_t^2$ ). Rearranging Equation (5.61), we obtain:

$$a_t = \frac{\varphi(q^{-1}) \Delta^d}{\theta(q^{-1})} \left[ y_t - \frac{\omega(q^{-1})}{\delta(q^{-1})} u_{t-b} \right] \quad (5.62)$$

Minimization of  $\sum a_t^2$  is essentially a non-linear least squares estimation of  $a_t$  in Equation (5.62). This is easily achieved with available routines.

#### **5.5.2.4 Diagnostic Checking**

Diagnostic checking of the model must be carried out to determine if the chosen form is adequate and, if not, to identify the source of the structural deficiency. The most basic checks involve the residuals of the model prediction ( $y_{\text{measured}} - y_{\text{predicted}}$ ). As discussed above, if the model is correct, these residuals should take the form of white noise (random with stationary mean of 0). An autocorrelation of the residuals is used to test this condition. If there is no autocorrelation, the model can be assumed adequate in accounting for all data. On the other hand, if there is residual autocorrelation, modification of the structure is necessary. If the input is crosscorrelated with the residuals, the transfer function structure is incorrect and must be modified to remove this crosscorrelation. If there is no crosscorrelation between the input and the residuals, the noise model must be modified to remove the residual autocorrelation.

The program written to perform the identification through time series analysis is listed in Appendix E.

## **CHAPTER 6: RESULTS AND DISCUSSION - CONTROL IMPLEMENTATION**

### **6.1 Introduction**

This chapter presents the results of the research conducted for this thesis concerning the implementation of various control loops on the UBC pilot CFBC. The underlying controllers, necessary for plant upgrading from manual to combustor control, are first discussed. These controllers are key to enabling the setting and maintaining of air flowrates and providing control over the solids circulation rate. The methods outlined in Chapter 5 are employed to implement a single loop controller for combustor temperature with fuel feedrate. The GPC algorithm was used for this loop and is shown to behave exceptionally well. As well, the identification techniques were applied with limited success. Difficulties encountered are discussed. Finally, a complete control structure is presented providing control over the combustor temperature, flue gas oxygen concentration, and the amount of heat removed by the in-bed heat exchanger. Although a multivariable process, it was found that the time constants for the various loops are such that each may be treated as a single loop controller. The solids circulation rate is used to control the amount of heat removed by varying the solids loading around the heat exchange surface of the in-bed heat exchanger. The flue gas oxygen

concentration is controlled by adjusting the total air throughput of the system. Both of these controllers employed discrete PID algorithms which sampled the process at 1 second intervals and were tuned on-line. The combustor temperature was again controlled with fuel feedrate by the GPC algorithm. A sample/control interval of 2 minutes was used for this loop. Results indicate exceptional behaviour of the overall control structure and, although probably not appropriate for a larger unit, demonstrate the general applicability of the GPC algorithm in the combustion process and the ease and reliability with which it can be implemented.

The purpose of the applied controls in this thesis was to provide further validation of the dynamic model described in Chapter 4 as well as to demonstrate GPC within a combustion system control structure. Optimization of the control system was not carried out. It was simply desired to attain some level of control to achieve the goals of this work. This level of control was assessed qualitatively. Compared to the years under manual control, the automatic control loops provide much faster responses to setpoint changes in addition to maintaining much tighter control around a given setpoint. Of issue here is that the controller maintain stability under all circumstances.

## **6.2 Basic Control Loops**

The underlying controllers are key to the successful implementation of any control scheme. Prior to the start of the experimental research, the UBC pilot plant had to be upgraded from manual to computer interactive control. The base control loops are described in this section together with an explanation of their purpose. The principal problems which had to be addressed were the tendency for the combustor to cycle while ostensibly being at conditions of steady state, and the need to control riser solids loading.

### **6.2.1 Air Flow Control**

Originally, two rotameters were used to monitor the air flows (primary and secondary air rates) and for manually setting the total air flow. Although adequate for prior research,

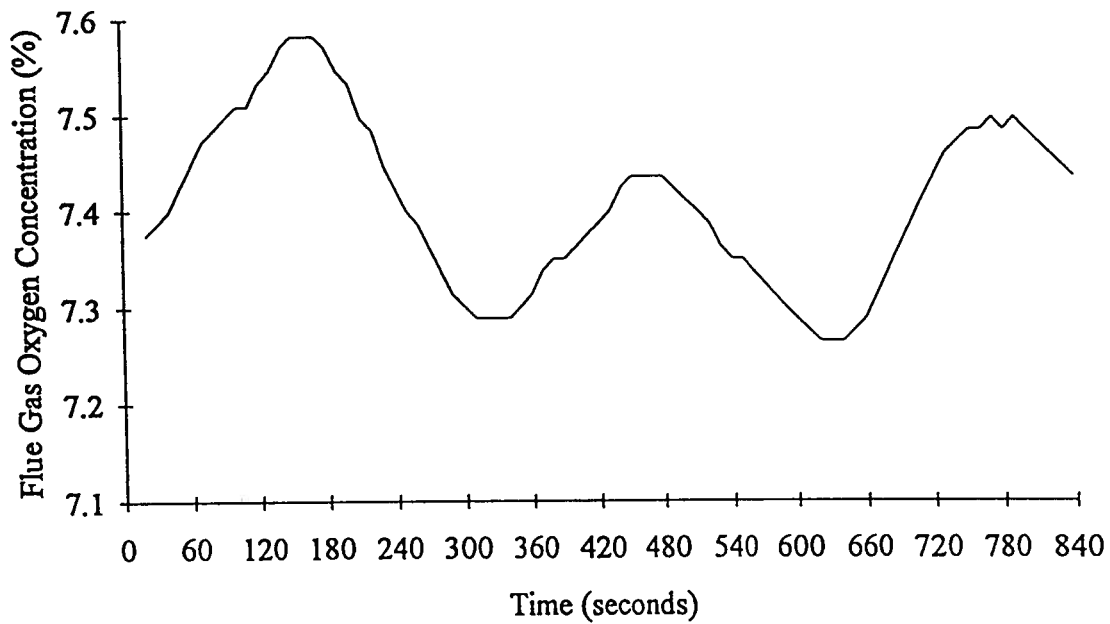
this system required modifications for dynamic and control studies. The need for modifications became apparent after initial experiments, involving step tests of various variables, were carried out to obtain a "feel" for the dynamic response of the unit. Results showed cycling of temperature, oxygen, and riser solids loading. This cycling was eventually traced back to the building air supply. Figure 6.1 presents an example of the effect of cycles in the air feed flow on the flue gas oxygen concentration.

Air to the rotameters was taken directly from the building main air compressor, which has an output design of 80 to 100 psig (550 to 690 kPa). As air from the source is used, the output pressure drops from 100 to 80 psig (690 to 550 kPa) at which time the compressor is activated, and replenishes the tank until the pressure reaches 100 psig (690 kPa). This leads to cycles which are approximately of 5 minutes duration. This pressure fluctuation causes fluctuations in the air flowrate to the combustor and consequent cycling in oxygen concentration (due to oxygen supply), riser solids loading (due to velocity fluctuations with no loading controller in place), and temperature (resulting both from oxygen available for combustion and changes in riser solids loading).

The initial solution to this problem was the insertion of a pressure regulator in the air supply line to drop the supply pressure to 275 kPag. This helped to filter out the compressor pressure fluctuations.

In order to facilitate computer control, pneumatic control valves were installed parallel to the air flow rotameters. These were conventional, pneumatically operated, equal percentage control valves responding to a 4 to 20 mA control signal.

To obtain digital flow measurement, orifice meters were placed downstream of the control valves. Pressure measurements were taken upstream and downstream of each orifice plate and converted through the data logger to digital pressure measurements. From this input, the computer performed conventional orifice calculations to determine the corresponding air flowrates. The inclusion of digital proportional controllers in the data logging algorithm, both employing a gain of  $0.02 \text{ mA/kg}\cdot\text{hr}^{-1}$ , accommodated the base



**Figure 6.1: Effect of Compressor Fluctuations on Flue Gas Oxygen Concentration with Manual Setting of Air Flowrate (Highvale coal, combustor temperature = 835 °C, gas velocity = 7 m/s, P/S air ratio = 3)**

feedback loop between each orifice run (flow measurement) and the corresponding control valves (flow controller). Continual operation of these loops (on a one second interval) enabled complete removal of flow cycling caused by the main air compressor.

Individual primary and secondary air flowrate specifications could be set through the computer by the operator. Further program algorithmic modifications were later incorporated which allowed the operator to specify the primary-to-secondary air ratio as well as the total air supply or gas velocity.

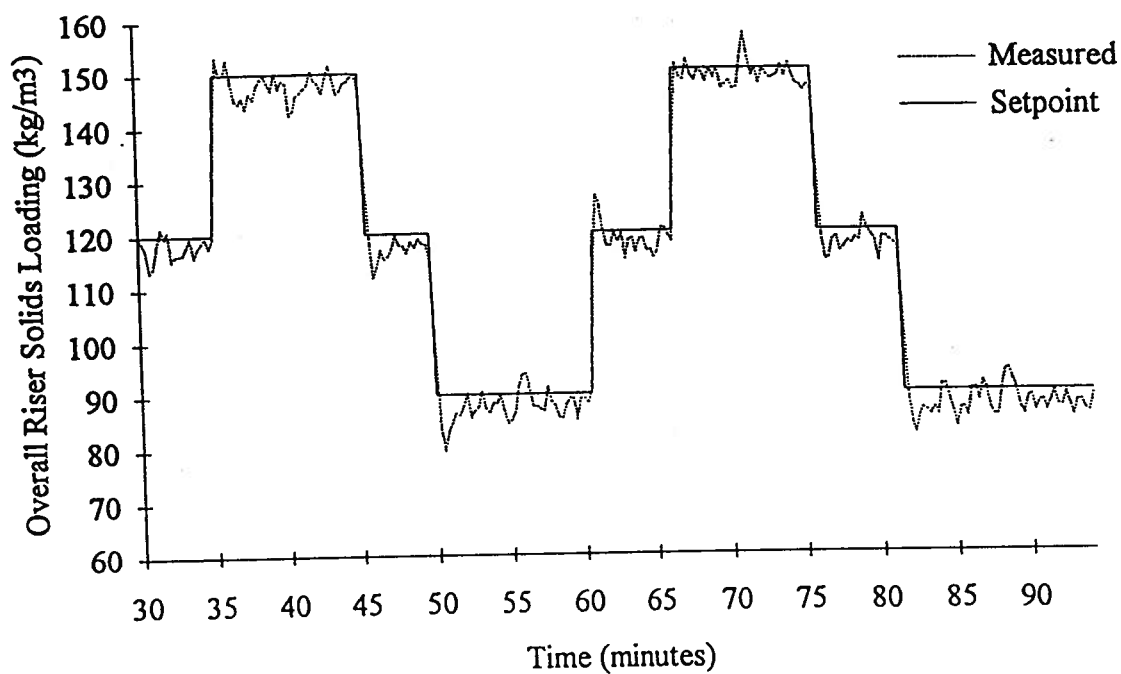
### **6.2.2 Solids Circulation Control**

Riser solids loading is varied through adjustment of the L-valve aeration rate. Earlier experiments simply dictated "high" or "low" solids loadings. During these experimental runs, the overall riser pressure drop was monitored by the operator using a water manometer and occasional manual adjustments were made through the L-valve air rotameter to maintain "high" or "low" solids loadings as required.

In order to automate the control, a computer actuated pneumatic valve was installed parallel to the L-valve air rotameter. A digital PID controller could then control riser solids loading by adjusting this valve. A setpoint for the riser solids loading is input by the operator and the controller meets this by adjusting the opening of the L-valve aeration valve.

The above control loop enabled experiments to be carried out at specified riser solids loadings. This was of particular importance in dynamic testing since it had been previously observed that uncontrolled variations in loading had a direct effect on temperature and flue gas emissions. Figure 6.2 demonstrates the application of this controller.

The typical operating range of riser solids loadings in the UBC pilot CFBC is from 50 kg/m<sup>3</sup> to 200 kg/m<sup>3</sup>. The solids loading controller, employing a gain of 0.02 mA/kg·m<sup>-3</sup>, an integral time of 20 s, and a derivative time of 0.12 s, is able to facilitate a step increase in the setpoint as a first order response with a time constant of approximately 7 seconds. A step decrease in the setpoint takes the appearance of a slightly underdamped second order



**Figure 6.2: Operation of Solids Loading Controller Showing Its Success in Following Step Changes in Setpoint of Solids Loading in Riser (Highvale coal, combustor temperature = 855 °C, gas velocity = 7.3 m/s, P/S air ratio = 2)**



response. The overshoot is less than  $10 \text{ kg/m}^3$ , and the change is achieved in approximately 10 seconds. Some difficulties with the solids loading control were associated with the fuel and could not be completely resolved. A case in point was a problem encountered when running for extended periods of time with Highvale coal. It was found that the response gradually became sluggish and eventually cyclic. When this occurred, the controller would be taken off-line. Although it is difficult to manually maintain a set-point, the cycling in solids loading was abated. As these experiments proceeded, there was a tendency for the loading profile to shift slowly such that a greater proportion of particles were at the bottom of the riser. In a few instances, the extent of this shift was such that L-valve aeration could be shut off with no solids at the riser exit, although a loading greater than  $100 \text{ kg/m}^3$  could still be maintained. This phenomenon is directly related to the nature of the particular fuel (Highvale coal), which is noted for a high shale content. The shale gives rise to inert "stones" of approximately 5 mm in diameter which build up in the bed. These stones tend to accumulate in the riser and do not become part of the circulating inventory. Although part of the bed, they are independent of the circulating solids and contribute uncontrollably to riser solids loading. Over time, these solids predominate the riser pressure drop, making control of riser loading increasingly difficult.

This pilot plant problem is not a significant issue for industrial scale CFB units where there are bed drains. Bed draining and replenishment were found to alleviate the problem on the pilot unit. For example, draining 20 to 30 % of the inventory and replacing it with fresh sand returned controllability for a few hours. However, this is not an acceptable procedure during testing as it can affect results which depend on bed material build-up, loss of carbon, limestone for  $\text{SO}_2$  removal, and a shift in the solids loading profile.

The solids circulation problem had implications beyond basic controllability. The nature of the research required long term testing for identification and adaptation. Because of the solids loading difficulties, a single dynamic test could rarely exceed 10 hours. This restriction meant an undesirable, but unavoidable, limitation on the work. The results of the

research are definitely promising and indicate the applicability of these techniques in industry - the limitations being imposed by the particular experiment in question, not by the technology or concepts.

### **6.3 Temperature Control**

The temperature controller is the major focus of this work and is discussed in detail in following sections.

Unlike solids loading and air flow, combustor temperature may be represented by values taken at a number of alternative locations for the purpose of control. For example, one may consider the average system temperature, maximum system temperature (usually in the primary cyclone), or temperature at the heat transfer surfaces (here the in-bed heat exchanger located in the upper portion of the riser, although industrially a large portion of the combustor is surrounded by heat transfer surfaces).

The most appropriate temperature would be a system average. This would allow the combustor temperature to be defined by a mean value while filtering out localized anomalies. Within the pilot plant, however, there is a practical difficulty in maintaining measurement of a system average temperature. Malfunction of any of the many thermocouples may not be detected or cannot be corrected "on the fly". Fortunately, with the fuels used in this work, uniformity of temperature around the solids loop was observed - within  $\pm 5$  °C. This allowed a single point measurement to provide an adequate approximation of the combustor temperature for control purposes. This measurement was taken from one of the upper three thermocouples located in the primary zone of the riser (Figure 3.8).

#### **6.3.1 Single Loop Temperature Control**

This section presents work concerned only with the control of combustor temperature. The purpose of this work was to gain experience with combustion control and to explore the

effect of GPC parameters. The results provide a demonstration of GPC - both its capabilities and its ease of application.

Note that the work reported in this section does deviate from methods outlined in Chapter 5. Specifically, identification was not carried out prior to these control tests. Identification is a necessary step in the implementation of any control structure and is key for the determination of appropriate models for model based controllers. Initially, the primary goal of this control implementation was to perform final debugging of the control and adaptive algorithms after their incorporation into the data logging program. After successful debugging, it was decided to continue and investigate the behaviour of GPC. Identification at this point in time was deemed impractical as plans were underway to alter the insulation of the pilot plant in order to reduce heat losses. Identification prior to re-insulation would provide inaccurate control models as the increased insulation would increase the thermal capacity of the unit, thus slowing the temperature response.

During the experiments described in this section, the riser solids loading and the total air throughput were maintained constant using the controllers discussed in Section 6.2. At the time of these tests, only approximately 10 kW of heat was being removed by the in-bed heat exchanger using air as the cooling fluid. Heat removal control was not considered feasible under these conditions. Hence, the riser solids loading was set by the operator and not considered part of the temperature control structure.

In addition, at the time of these experiments, the flue gas oxygen measurement was considered very slow as well as of doubtful accuracy. Because flue gas oxygen was considered somewhat secondary in terms of heat generation, it was not controlled. Instead, the total air throughput was determined by the operator and changed at his/her discretion, independent of the control structure under investigation.

*Throughout this work experiments are referred to by run numbers and test numbers. A run refers to the total period of an experiment from combustor start-up to shut-down. The runs are*

*sequentially numbered. There are gaps in the numbers because the reported runs were carried out between runs of the CFBC pilot plant for other purposes. Most of the unreported runs are not related to this work or, for any number of reasons, shut-down occurred before any relevant results to this thesis were obtained. A test designates a period of time during which one particular aspect or more was investigated.*

#### **6.3.1.1 Run #12**

The first implementation of Generalized Predictive Control (GPC) was during Run #12 in May, 1992. The GPC algorithm had been incorporated into the data logging routines and it was necessary to conduct an on-line test for final debugging. Since this was successful, it was decided to investigate the performance of the GPC controller, although identification is always recommended prior to implementation.

This experiment was kept relatively short because the secondary return became blocked, rendering the secondary cyclone inoperable. As a result, fines carried over from the primary cyclone passed directly to the baghouse and were not returned to the CFB. It was recognized that this loss would greatly reduce the combustion efficiency as well as affect the dynamic behaviour of the system.

Throughout the work described in this thesis, every effort was made to operate the pilot plant as consistently as possible from experiment to experiment. However, this was difficult to accomplish in practice because of a range of upsets and interfering factors - some unknown, some that could not be monitored, some unanticipated. More often than not, as in the above case, the experiment was continued while recognizing that the desired degree of consistency was being sacrificed.

A general rule of thumb for the selection of the sample/control interval is to choose it such that is  $1/15^{\text{th}}$  to  $1/4^{\text{th}}$  of the settling time of the plant. Best estimates of the temperature in response to a step change in fuel feedrate indicated rise times of 20 to 30 minutes. Based on this rule of thumb, this places the sample/control interval from 1 to 6 minutes. A shorter

interval is used in this section as discussed below while the 2 minute interval used in Section 6.4 is within the estimated limits.

Implementation of the controller at a 5 and then a 2 minute sample/control interval revealed that it must "warm-up" to current operating conditions before it can perform a proper controlling function. This is because the data logging and control routines were written as two completely independent programs (due to executable code size limitation). The controller only has access to information passed to it from the data logging program at the time it is called. At the time of this test, the control program was not accessed until the controller was turned on.

The warm-up period is that time necessary for the controller to collect sufficient process information to enable it to perform its function. The consequences of allowing no warm-up can be illustrated by considering the controlled variable,  $y_t$  (in this case temperature). On the first call to the control routine, the current value of  $y_t$  is supplied. But the control algorithm requires knowledge of  $y_{t-1}$ ,  $y_{t-2}$ , etc. as well. Any control action performed without the knowledge of these previous values cannot be correct in that calculations are performed based on inaccurate information (0, as initialized by default). The same is true for the controlled variable (fuel feedrate). The sample/control intervals of 5 and 2 minutes means that the length of time before the controller has sufficient prior data to behave correctly is such that the unit is usually too far out of operable conditions for control to be regained. An example would be the controller shutting off the fuel feed resulting in too great a temperature loss before feeding resumes. This occurs because the controller has a value of, say, 850 °C for  $y_t$  and 0 for  $y_{t-1}$ ,  $y_{t-2}$ , etc. Consequently, it "sees" the temperature shooting upwards from 0 to 850 °C in one sample period and shuts off the fuel to stop the "apparent" temperature rise. This problem can be eliminated by allowing calls to the control program, providing current measurements and manipulated variable settings, at the appropriate sample interval without performing control calculations.

Successful implementation was achieved in this same run using a sample/control interval of 15 seconds. In this case, the warm-up was short enough that the combustion conditions did not change significantly before the controller was fully operational.

For this test some simplifying assumptions were made. It was assumed that no noise was present and that the  $A(q^{-1})$  polynomial acting in the backwards shift was constant and of the form:

$$A(q^{-1}) = 1 - 0.9q^{-1} \quad (6.1)$$

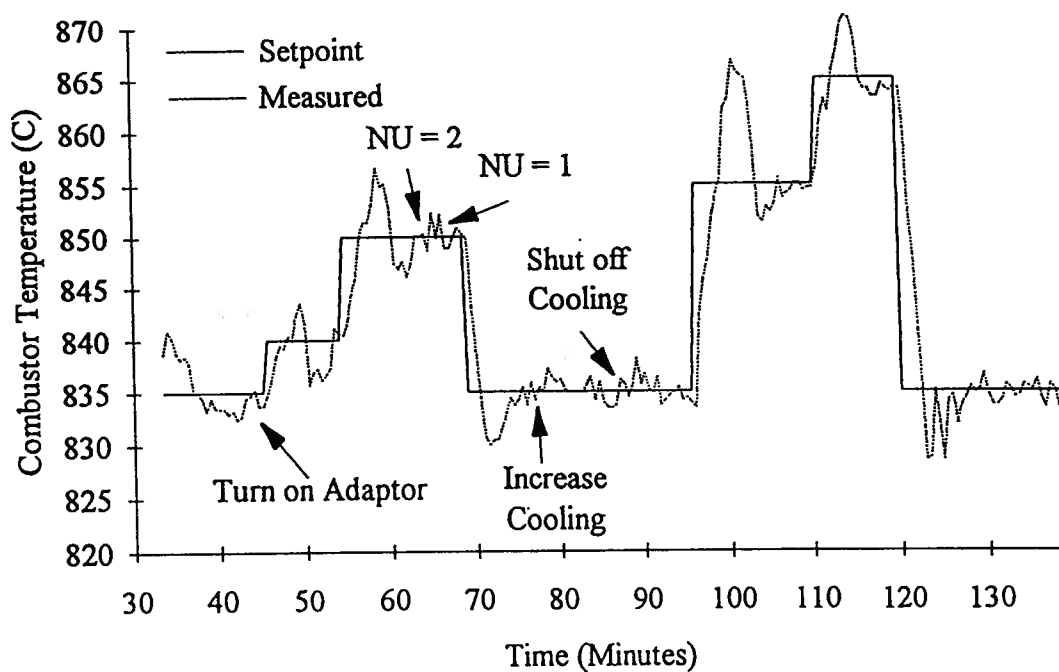
During initialization, the transfer function was assumed to be:

$$\frac{y(q^{-1})}{u(q^{-1})} = \frac{1q^{-1}}{1 - 0.9q^{-1}} \quad (6.2)$$

When the adaptive portion of the controller was turned on, it was designed to fit a 3<sup>rd</sup> order numerator:

$$\frac{y(q^{-1})}{u(q^{-1})} = \frac{(b_0 + b_1q^{-1} + b_2q^{-2} + b_3q^{-3})q^{-1}}{(1 - 0.9q^{-1})} \quad (6.3)$$

The controlled temperature during this test is shown in Figure 6.3. At time 33 minutes, the controller was turned on using Equation (6.2) to bring the temperature to the initial setpoint of 835 °C. At time 45 minutes, the adaptive portion of the controller was turned on and allowed to update the model, Equation (6.3), at each sample/control interval. The adapter was left on for the the remainder of this test.



**Figure 6.3: Temperature Response - Run #12 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S=2, average solids loading = 120 kg/m<sup>3</sup>)**

It is apparent in Figure 6.3 that each change in the setpoint was accompanied by an overshoot of approximately 50%. This was due to a combination of three important factors. Firstly, the denominator of the transfer function was fixed. This particular controller was intended as a multivariable controller for which on-line adaptation of the  $A(q^{-1})$  polynomial is difficult. Identification of this polynomial was planned at a later date. In the meantime, by fixing this polynomial, the true dynamics of the temperature were not known by the controller.

Secondly, the sample/control interval was much shorter than needed for this application. One of the advantages of model-based predictive control is the ability to use longer sampling intervals, leading to fewer and lower amplitude control moves. Shorter control intervals result in stronger control actions. This leads to the third factor.

Constraints must be placed on the controlled variable (rotary valve signal). Such constraints do not allow the controller to demand either unreasonable or unrealizable control settings. The rotary valve motor controller reacts to a 4 to 20 mA input signal. Likewise the data logging module is a 4 to 20 mA output block. A request from the program (be it calculated by the controller or input by an operator) for a current greater than 20 mA or less than 4 mA results in a hardware error and subsequent shut-down of the system. To ensure appropriate operator inputs, one simply incorporates a check loop into the input routine. If the input is outside an acceptable range, the routine requests the operator to enter a new value. Similarly, if the controller demands an unacceptable setting, a maximum or minimum value is automatically employed. For example, if the controller determines that a 2 mA signal is required, a 4 mA signal is generated. Throughout the course of this research, a working range of 4 to 16 mA was employed, 4 mA being the minimum output possible and 16 mA considered a safe maximum.

A more theoretical approach to employing constraints is to include them in the cost function of the controller (Dion et al., 1988). This allows the controller to optimize the setting with foreknowledge of the upper and lower limits of the manipulated variable. Since this was an initial test implementing the basic GPC algorithm, this theory was not applied here.



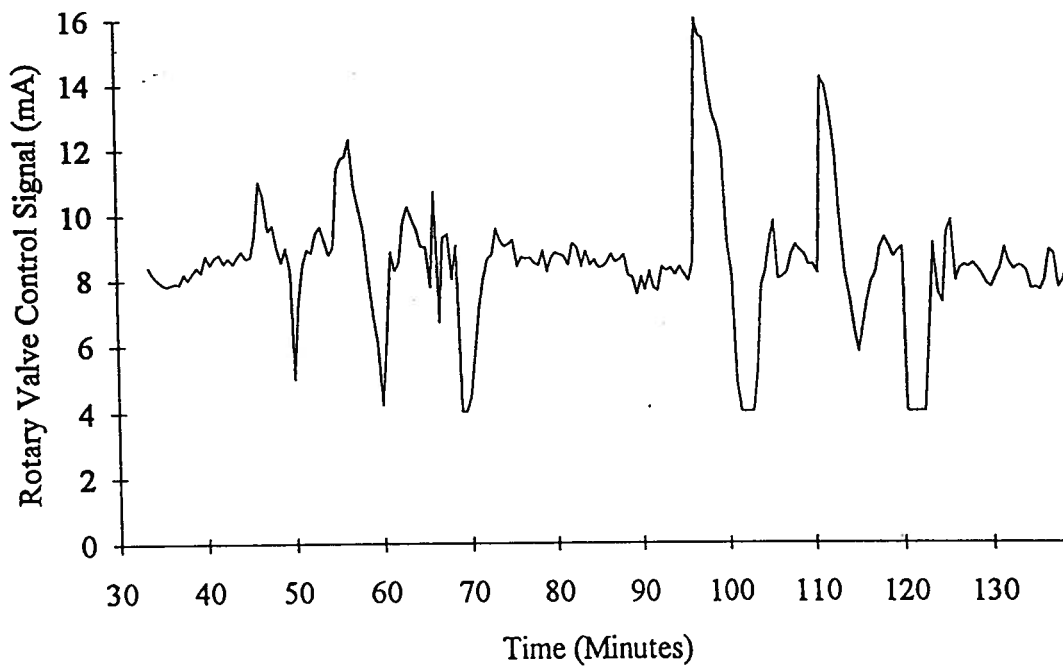
In any event, the 4 to 16 mA allowable range of control action far exceeds any reasonable range of feedrate settings so that any adequately designed controller would stay well within this operating range.

The consequence of these three contributions to overshoot can be seen more clearly in Figure 6.4, which presents the fuel feedrate (as a mA signal to the rotary valve) during this test. Comparing this figure with the temperature response (Figure 6.3), it is obvious that each change in setpoint leads to substantial control action.

After each change, the feedrate increases or decreases for some period of time (usually until overshoot occurs) and then undergoes a large correction in the opposite direction to counterbalance the over- or under-feeding that has taken place. The plateaus indicate that polynomial  $A(q^{-1})$  in the model equation is incorrect. The controller makes the initial adjustment upon a setpoint change to approximately where it should be, based on the model. By staying near that point, it appears that, based on the fixed model, the unit is reacting more slowly than predicted. The excessive control action is then maintained by the controller in an attempt to force the faster, expected, response. It is not until overshoot occurs that the controller "realizes" that a counter correction is required to bring the temperature back.

The size of control actions in Figure 6.4 are considered excessive as swings in the range of 6 to 8 mA occur, while even a 3 mA change is considered substantial. Under manual control, changes greater than 1 mA have rarely been employed over the 8 year life of this plant. As stated earlier, a longer control interval would significantly reduce the size of control actions.

Also of note in Figure 6.4 is the differing behaviour of the fuel feed under increases and decreases in the temperature setpoint. Increases occur at 46, 55, 97, and 111 minutes. The mA signal undergoes a three stage adjustment - a sharp increase followed by a large decrease and a return to appropriate settings in fuel feedrate. In contrast, the step decreases in setpoint at 68 and 120 minutes lead to only a two-stage adjustment - a large initial decrease followed by a return to acceptable levels. Although results are insufficient from this test to



**Figure 6.4: Rotary Valve Speed - Run #12 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**

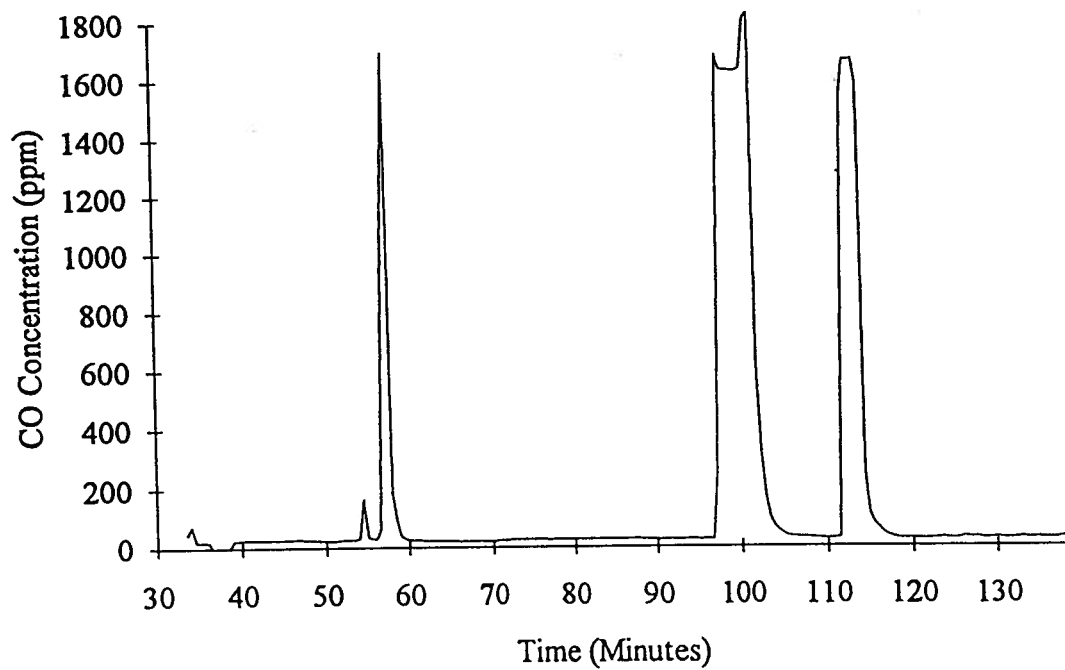
state unequivocally that the system is non-linear, it is obvious that two different responses are encountered depending on whether one is increasing or decreasing the fuel feedrate.

Figure 6.5 presents the CO measurements for the flue gas in this experiment. The spikes correspond to the increases in temperature setpoint and are significant in that they indicate that the unit is being taken into poor operating regions by the controller. These spikes occur during periods where overfeeding has occurred. When there is too much fuel and not enough oxygen, incomplete combustion results, leading to the sharp increases in CO leaving the system.

Over-action of the feeder, as shown in Figure 6.4, is the source of this problem. Likely ways of preventing the CO peaks include: (1) reducing the maximum allowable fuel feedrate setting from 16 mA to some lower value, (2) implementing some form of controller for the air flow that takes action based on a feed-forward system connected to fuel feedrate or feedback from oxygen availability, or (3) implementing a controller based on CO production to cut the fuel feedrate when incomplete combustion is indicated. However, each of these measures has its own particular limitation or drawback.

Changing the operable range of the fuel feedrate is not truly an acceptable solution. This range is only in place for hardware limits and safety. As previously stated, these limits should never be reached by a properly designed controller, and therefore should not be expected to be employed by the controller.

Feed-forward control of oxygen is often used in industry. Any increase or decrease in fuel feedrate is accompanied by an appropriately proportional change in the air flows, augmented by a feedback controller to correct for any errors in oxygen concentration. Unfortunately, physical aspects of the UBC pilot plant made implementation of this type of control impossible. No method of measuring the fuel feedrate on the time scale required was available. Steady-state rates are determined by averaging the change in weight of the feed hopper over a period of time - at least thirty minutes for an accuracy of  $\pm 10\%$ . Instantaneous measurements of feedrate for control purposes could not be obtained. This is why the fuel



**Figure 6.5: CO Response during Run #12 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**

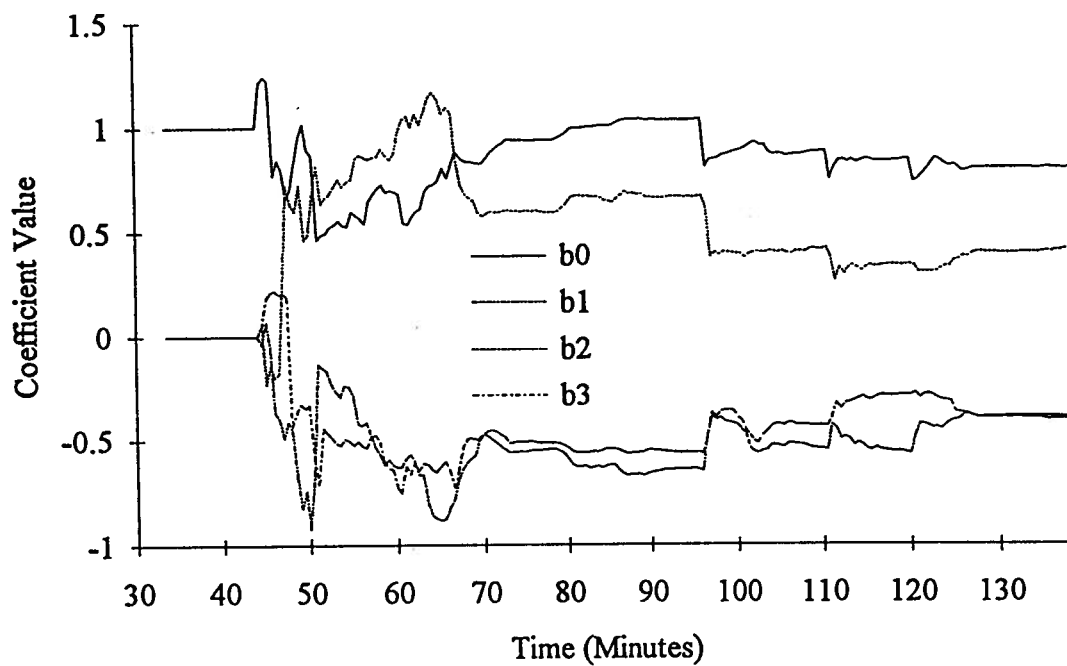
controller was designed around the mA control signal to the rotary valve feeder. This signal would not work in conjunction with a feed forward controller as a constant mA signal does not ensure a constant feedrate, which can be influenced by such factors as particle size, mass of solids in feed hopper, and both hopper pressure and system back-pressure. This, added to the fact that one of the aims was to maintain control during changes in fuel type, made any form of feedforward control between fuel and air impractical.

At the time of these tests, it was not feasible to base an air controller on oxygen concentration. The oxygen measurement was extremely slow (requiring approximately 5 minutes) and somewhat unreliable due to difficulties in maintaining constant flue gas sample flows. The third solution, reducing fuel flow upon the occurrence of CO spikes, appeared to be the simplest solution and is discussed below with reference to the results for Run #13.

The adaptor during this test appeared to work acceptably well. The results of the adaptation of the B coefficients is shown in Figure 6.6. For the first 25 minutes or so after initialization at  $t = 43$  min, there was considerable fluctuation in these values. This is attributable to a lack of prior data at the time of implementation. The adaptor is "searching" for appropriate values.

As can be seen from the latter part of this test, variable adaptation appears to proceed in steps which correspond to the large changes demanded by the controller. Since the nature of the adaptor is to be more sensitive when more information is available (large changes in fuel feed setting), these large changes take place when swings in the rotary valve mA setting occur. Although the adaptor behaved as designed, the fuel action took the unit to undesirable operating conditions (over/under-feeding), and adaptation at these times (spikes in values) is undesirable.

For the most part, the variables appear to be trending in a predictable manner. As one proceeds to the end of the test, the change in the coefficients becomes less pronounced. This may be an indication that the coefficients are close to the correct "tuned" values, although the smaller changes may simply result from the adapter having less sensitivity over time.



**Figure 6.6: Adaptation of B Polynomial - Run #12 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**

The controller behaved exceedingly well, given its ad hoc implementation. The adaptor was shown to provide reasonable values for the coefficients and did not indicate any "blow-up" or unacceptable loss of alertness during this test. The ability of the controller to maintain control with a poor initial guess of transfer function and a fixed A polynomial demonstrates one of the greatest advantages of GPC - its robustness to modeling errors and under/overestimated model orders.

A final point of interest in this test is the effect of heat removal. At 77 min, the calorimetric cooling was increased by approximately 1 to 2 kW and at 88 min all cooling was shut off, leading to a reduction in heat removal of approximately 10 to 12 kW. The effects of these changes in heat removal are very difficult to see in the temperature response (Figure 6.3). There was no change in temperature due to the increase in cooling while there was a very slight increase in temperature when all cooling was removed. 10 kW is only about 7% of the total heat input to the system so it is not surprising that significant, short-term effects were not noted.

Referring to Figure 6.4, the effects of these changes become more apparent. At 77 min there was a slight increase in fuel feedrate to maintain temperature for the increased heat removal, while at 88 min there was a notable drop in fuel feedrate corresponding to the removal of cooling. Perhaps most apparent is the response of the adaptor during the above periods (Figure 6.6). While the observable behaviour of the CFB appears insignificant (slight adjustments of controller to maintain temperature), there are large corrections shown by the adapter at these times. This indicates that, while these changes may appear unobservable, they are providing extensive dynamic information to the controller for adaptation.

#### **6.3.1.2 Run #13**

Run #13 was performed in June of 1992, during which four tests were conducted. These tests were unavoidably shorter than desirable due to a building power shutdown.

By the time of this run, it had been determined that the multivariable control mentioned above was unrealizable. Therefore, the model function was slightly modified from the previous run. Noise was again not considered, and the B polynomial was still taken as an adaptable third order expression. However, the adaptor was also allowed to operate on the A polynomial, although of first order:

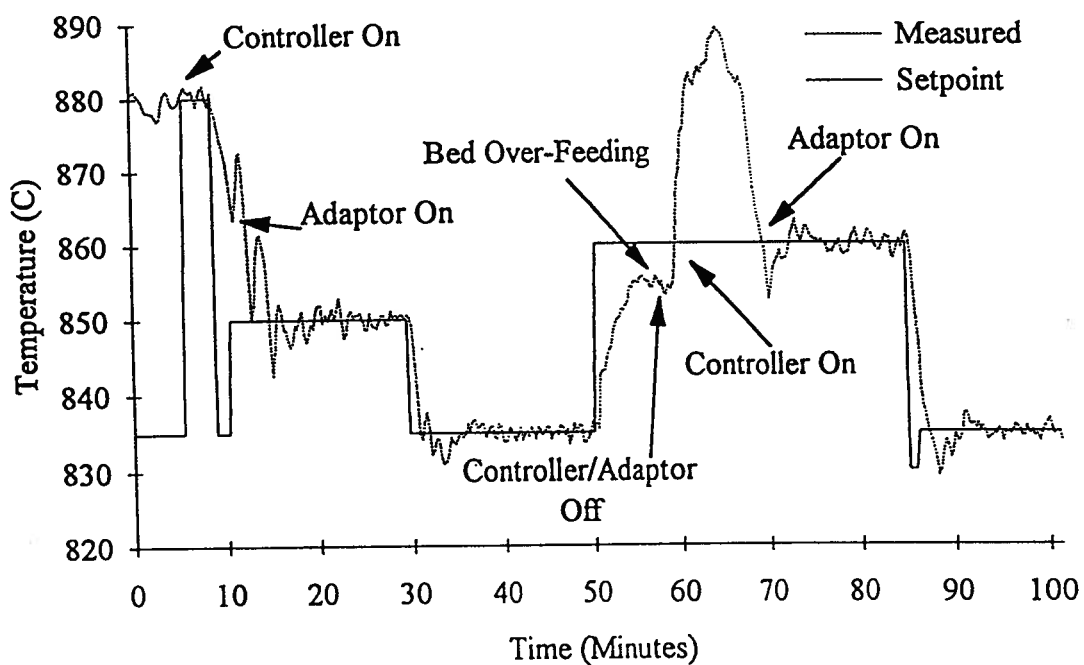
$$A(q^{-1}) = 1 + a_1 q^{-1} \quad (6.4)$$

The first test was a repeat of Run #12 to attempt to adaptively tune the controller with the variable A polynomial. Figure 6.7 shows the temperature during this test and indicates that, after the initial tuning period, temperature control appears quite good (except at the 60 min mark).

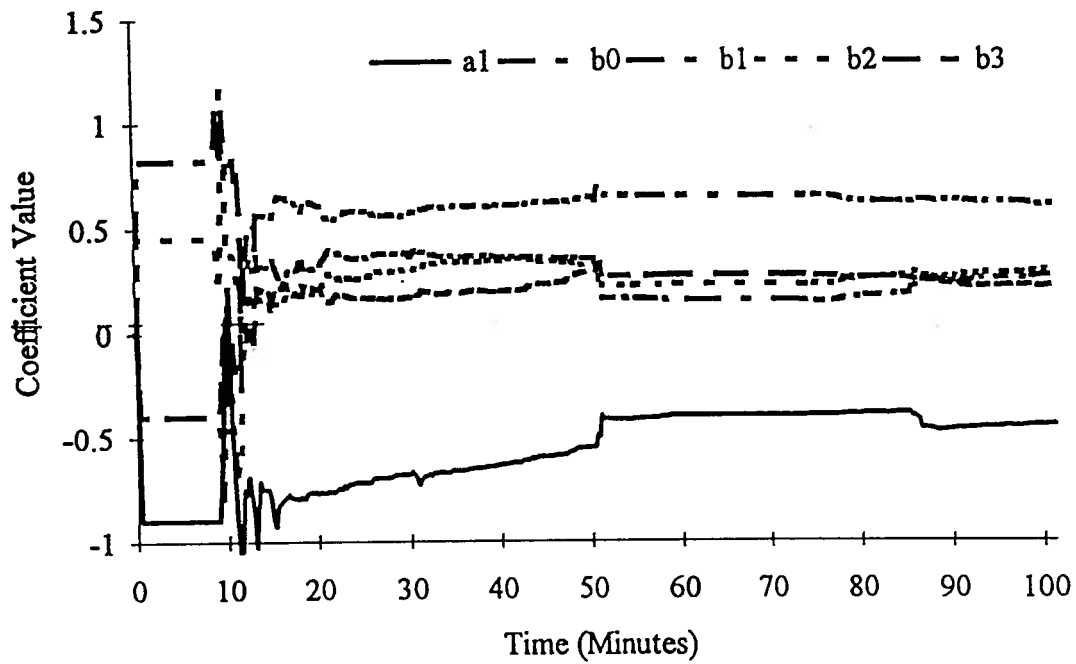
Figure 6.8 presents the polynomial coefficients throughout this test. The initial values were taken to be those at the end of Run #12. As can be seen, the fixed value of -0.9 used for the coefficient,  $a_1$ , in the previous run is quite different from the value (-0.45) determined by the adapter. Also the coefficients of the B polynomial are very different from the previous run. One reason for this is the fact that the A polynomial is now adapting and, therefore, is probably more correctly accounting for some of the dynamics. In addition, as discussed above, it is impossible to maintain fully consistent conditions throughout a run, let alone between different runs. The most obvious difference in this case is that the blockage in the secondary return which had occurred in Run #12 was not a factor for Run #13.

Most notable in this test is the dramatic demonstration of overfeeding following the step change in the temperature setpoint from 835 to 860 °C at 50 min. The rotary valve control signal is shown in Figure 6.9. At the setpoint change, the valve opened to its allowable maximum (16 mA) and stayed there in its attempt to raise the temperature. The temperature plot (Figure 6.7) shows the temperature responding to this massive overfeeding.

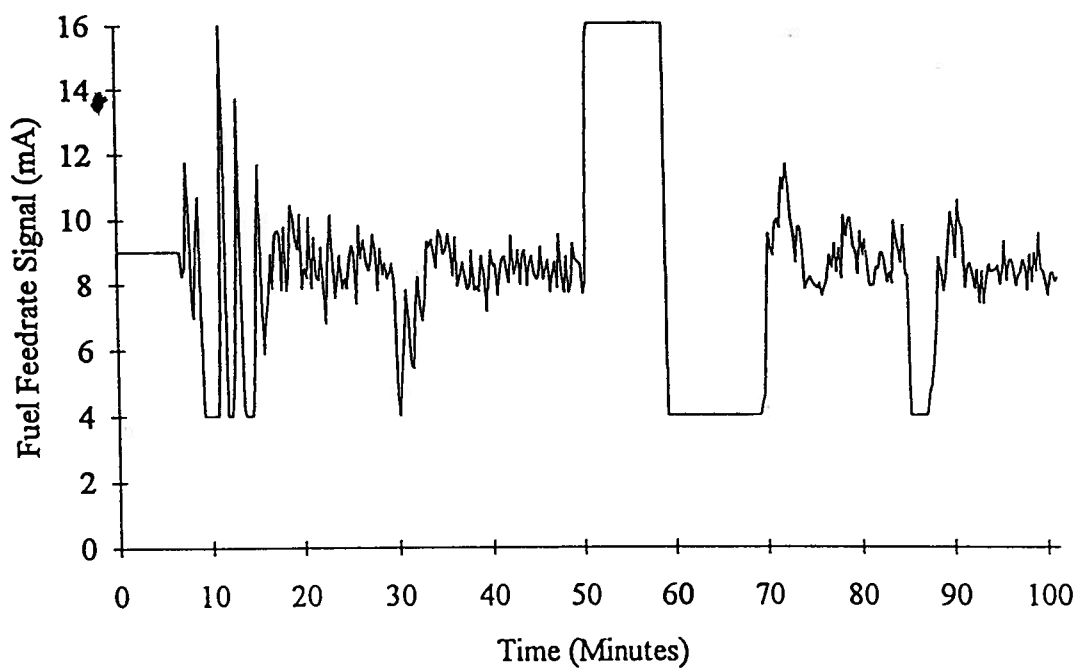




**Figure 6.7: Temperature Response - Run #13, Test #1 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**



**Figure 6.8: Adapted Model Coefficients - Run #13, Test #1 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S = 2, average solids loading = 120 kg/m<sup>3</sup>)**



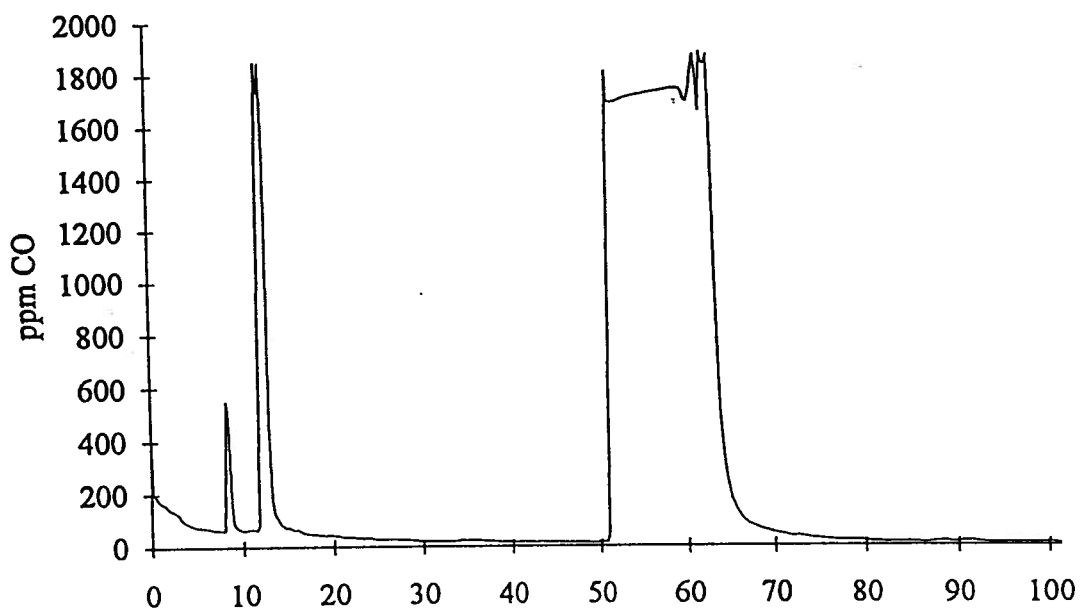
**Figure 6.9: Rotary Valve Control Signal - Run #13, Test #1 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**

After about 5 minutes, the temperature reached a maximum of approximately 855 °C and then began to drop even though the feeder was wide open. The bed had become so overfed with carbon that a highly reducing condition had been created throughout the system with additional fuel serving to absorb sensible heat. Insufficient heat was being generated to maintain the temperature.

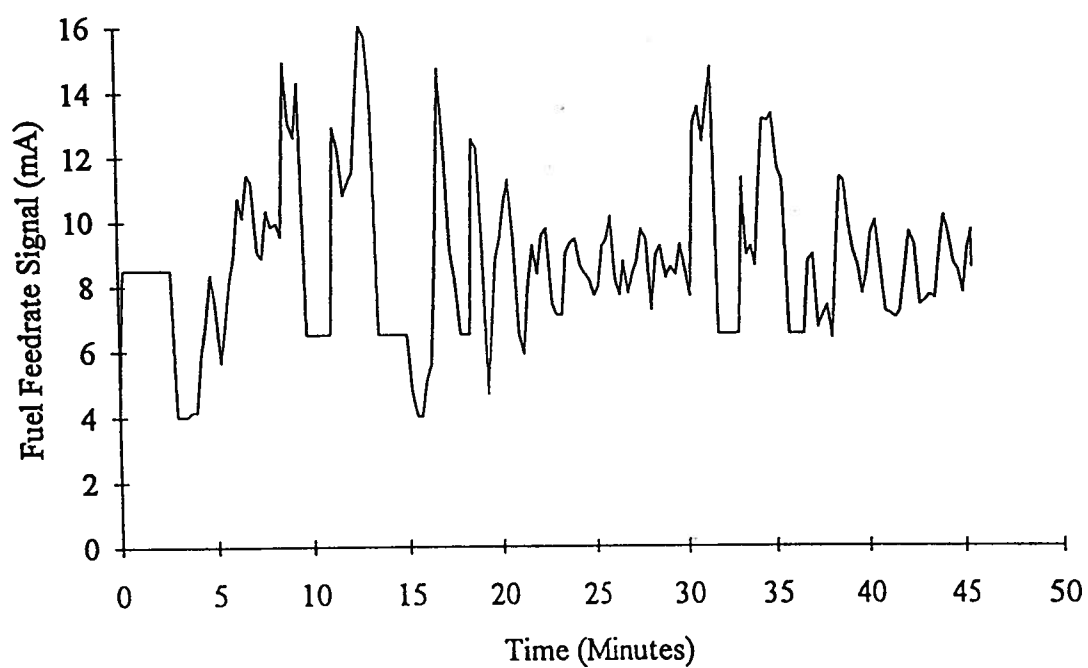
At approximately 59 min, the control was switched to manual and the fuel feed reduced to 4 mA (ie. shut-off). Almost immediately there was a sharp rise in temperature as the carbon inventory of the bed was reduced to levels conducive to combustion. Once the temperature had risen above the setpoint, the control was returned to automatic and the controller maintained a 4 mA rotary valve signal until the carbon had burned out and the temperature had returned to the desired value of 860 °C. It is notable how well the controller regained control over the temperature after the extreme overfeeding of the bed and subsequent manual intervention. Again, the only physical indication of overfeeding was a sharp increase in the CO concentration of the flue gas as shown in Figure 6.10.

In order to avoid overfeeding the bed, Tests #2 and #3 were performed with the addition of a "safety" switch within the algorithm which, upon detection of a CO spike, would reduce the rotary valve control signal to 6.5 mA (test 2) or 7.5 mA (test 3). This simple solution resulted in a significant deterioration in control performance. In Test #1, the control could maintain the setpoint within  $\pm 2$  °C (noise) while in Tests #2 and #3 fluctuations were of the order of  $\pm 5$  °C. Also, these fluctuations were much more regular in nature. This periodic cycling is directly related to the fuel feedrate determined by the controller as shown in Figures 6.11 and 6.12. With the "safety" switch, instead of a single peak, or plateau, occurring in the CO concentration with a step increase, there are a number of peaks associated with each step change, as illustrated in Figures 6.13 and 6.14.

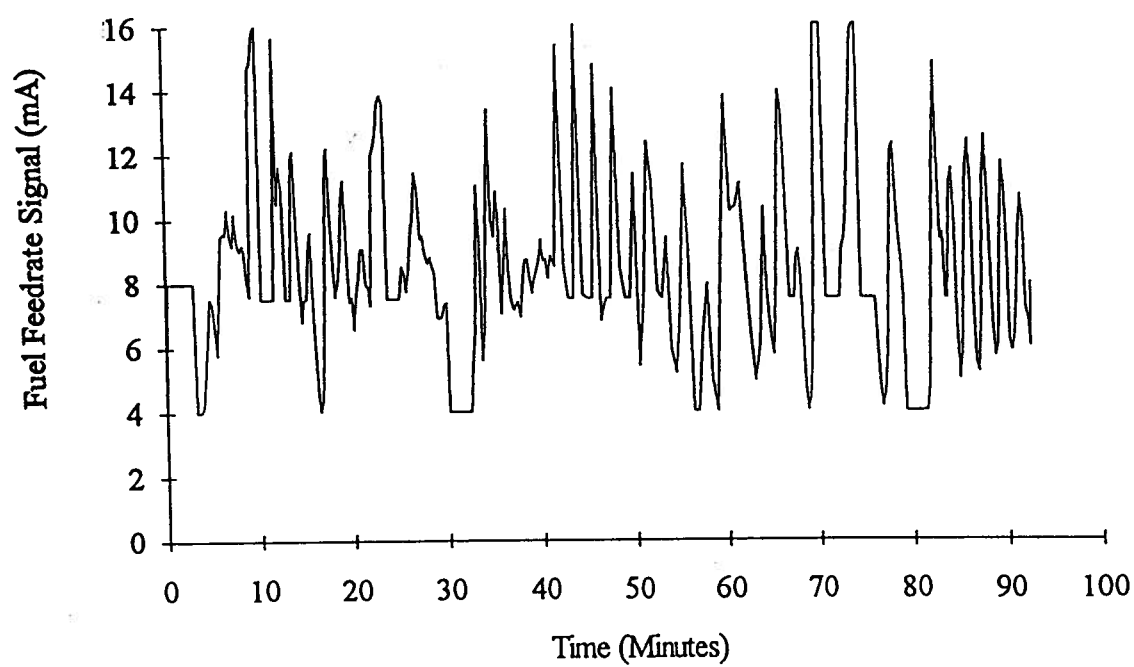
A typical increase in temperature setpoint caused the controller to open the fuel feed to some very high value (or to its maximum allowable setting of 16 mA). After a few minutes of overfeeding, the CO analyser sees the CO spike in the flue gas. The switch then closes the



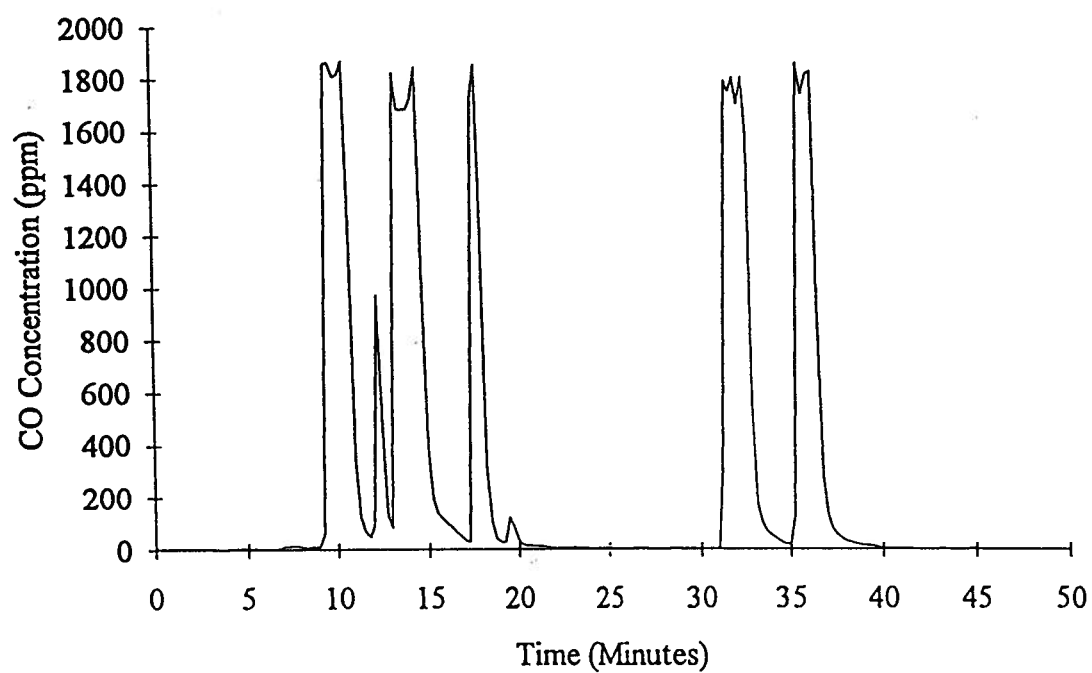
**Figure 6.10: CO Response - Run #13, Test #1 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**



**Figure 6.11: Rotary Valve Speed - Run #13, Test#2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**

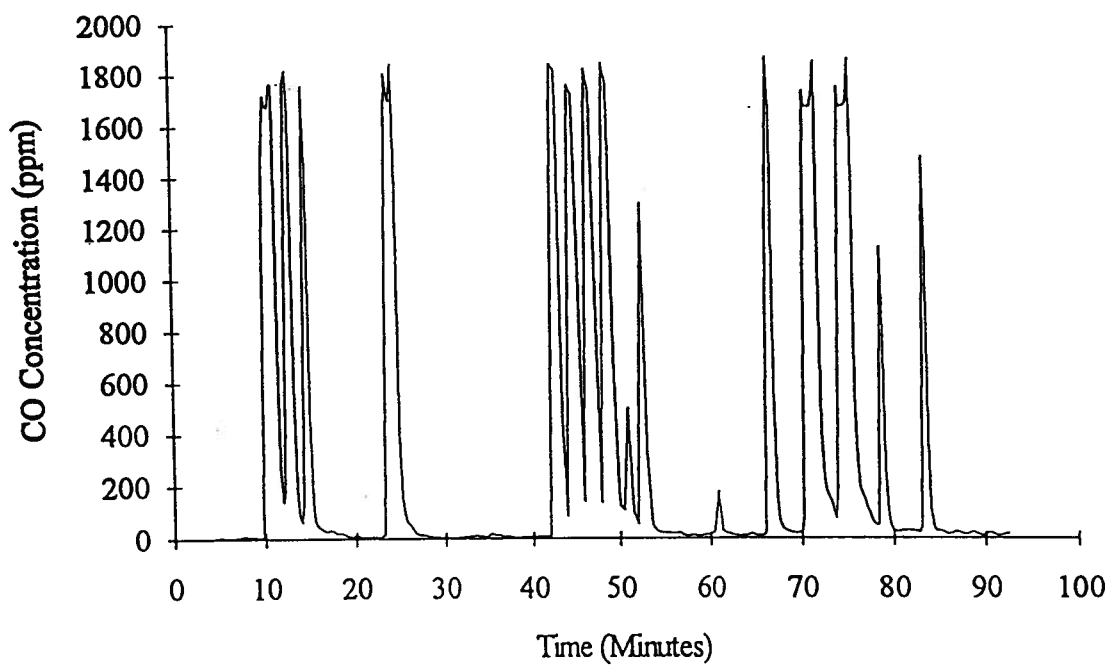


**Figure 6.12: Rotary Valve Speed - Run #13, Test #3 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**



**Figure 6.13: CO Response - Run #13, Test #2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**





**Figure 6.14: CO Response - Run #13, Test#3 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**

feedrate to its minimum setting until such time as the CO concentration drops back to "normal" levels. At this point, the temperature setpoint has usually not been reached, so the controller again opens up and over-feeds the bed once more. Depending on the size of this setpoint change, this sequence repeats itself as many as six times during the transition to the new steady state. This cycling, initiated by a setpoint change, continues long after the new temperature has been reached. The continuation demonstrates the instability caused by the CO monitoring switch. Inspection of the temperature responses in Figures 6.15 and 6.16 suggests that these cycles would eventually subside over a longer period of time.

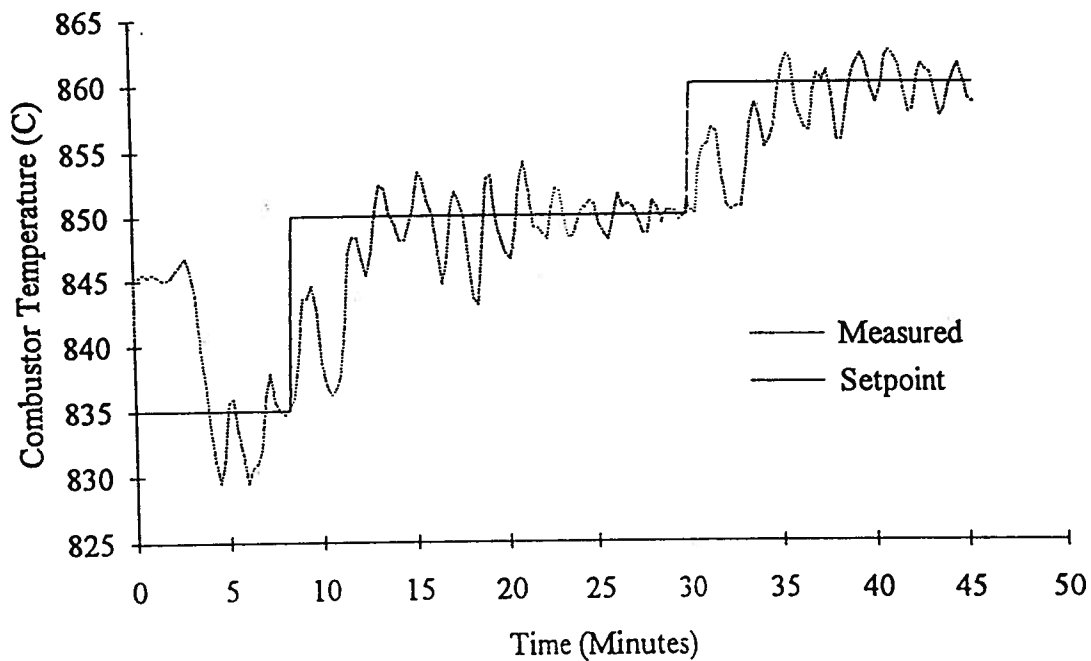
#### **6.3.1.3 Run #14**

Two final tests (Run #14) were performed in June of 1992 with the single variable GPC controller. The first implemented the previously used polynomials (Equation 6.3 and Equation 6.4). A series of step changes in temperature setpoint was induced, and the controller horizon, NU, was varied to study its effect on the control of temperature. NU was changed from 1 to 4 during this test. The times at which these changes were made are indicated on the temperature response shown in Figure 6.17. This figure shows very little effect of control horizon on the temperature response, although the responses for  $NU = 1$  (first and last step increases) do appear to be somewhat more underdamped second order responses than the others.

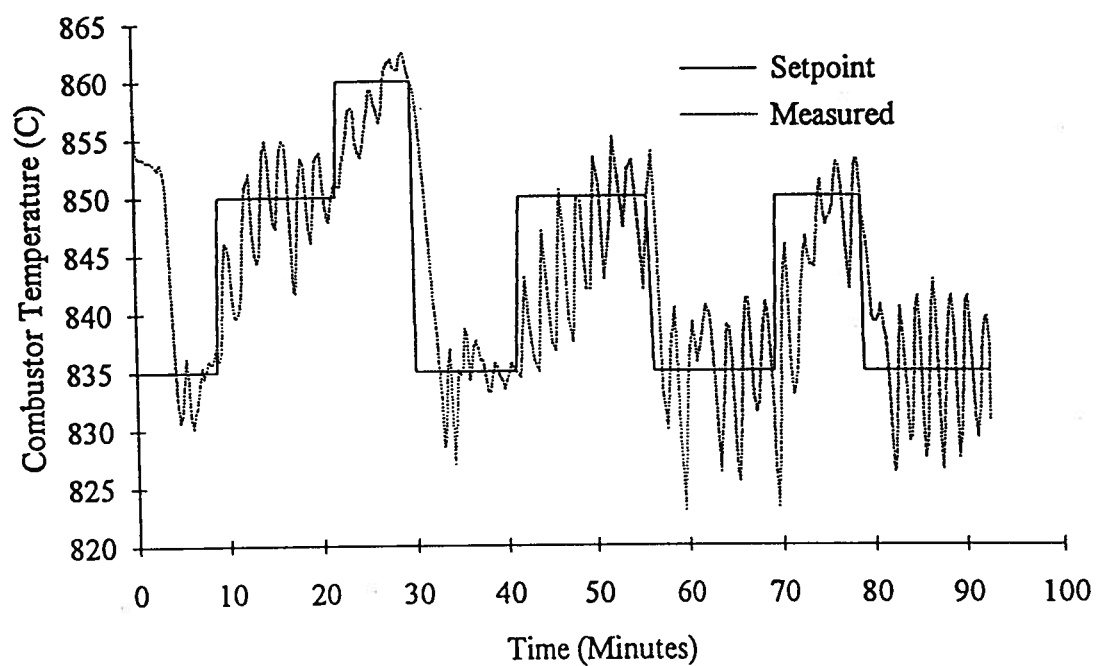
The rotary valve control signal for this test plotted in Figure 6.18 shows the expected stronger control actions being taken for  $NU > 1$ . This is typical of higher control horizons. The subsequent increased control action is undesirable in this application and, as shown in Figure 6.17, has very little influence on the control performance.

A final note on the test is that, as shown by the CO response in Figure 6.19, bed overfeeding remains a problem under this control scheme.

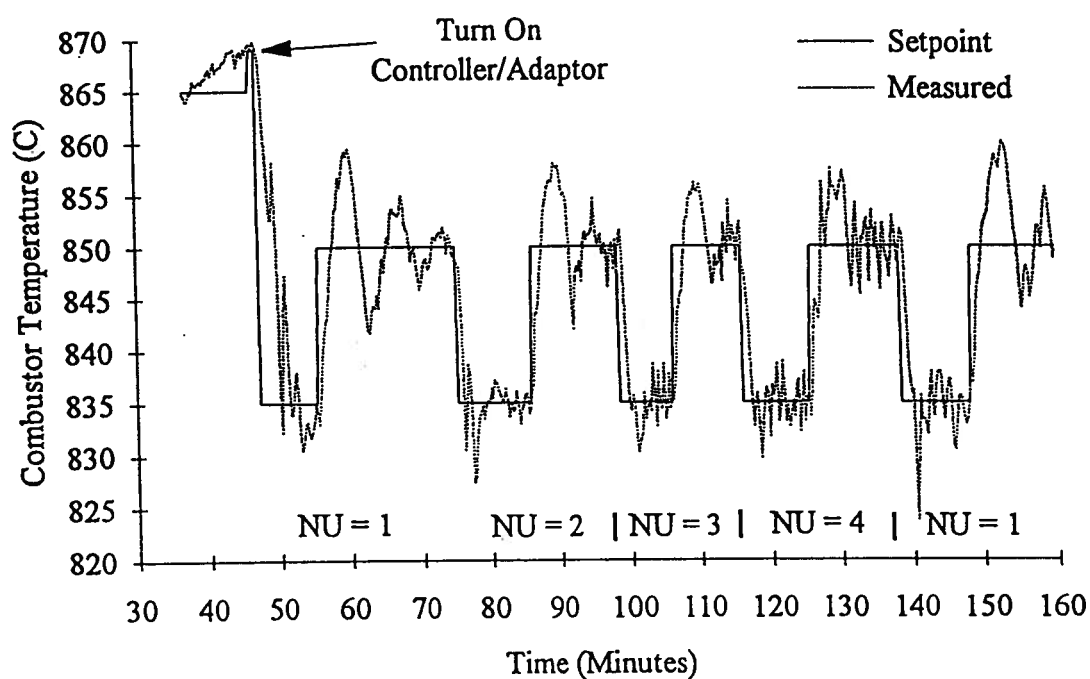
In the second test, the order of the adaptable A polynomial was increased to second order:



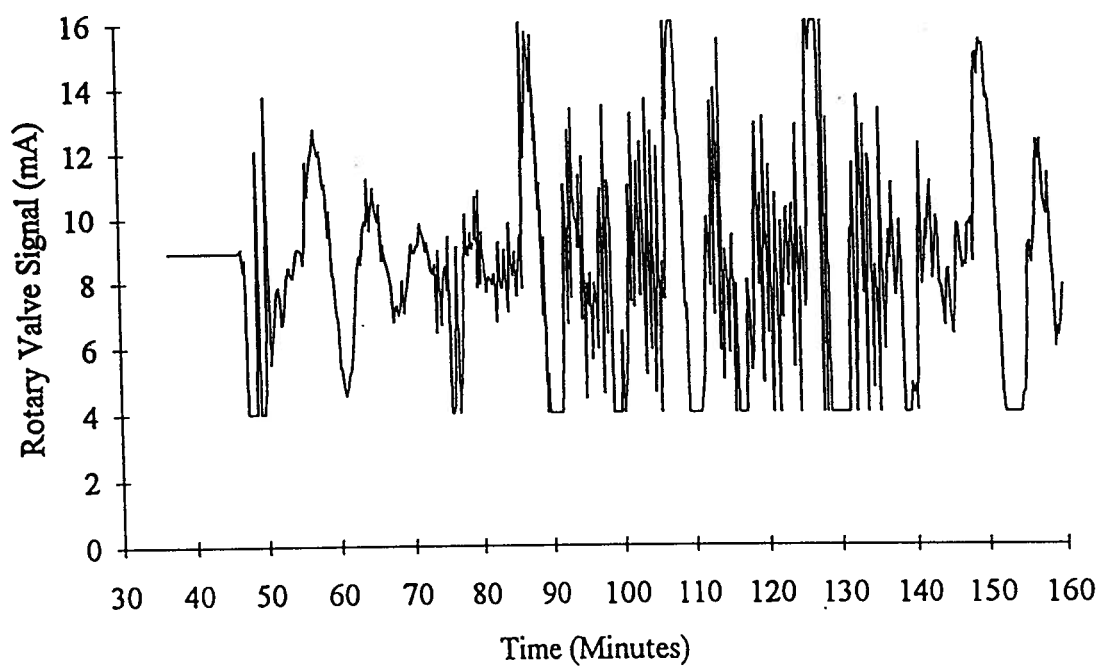
**Figure 6.15: Temperature Response - Run #13, Test#2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**



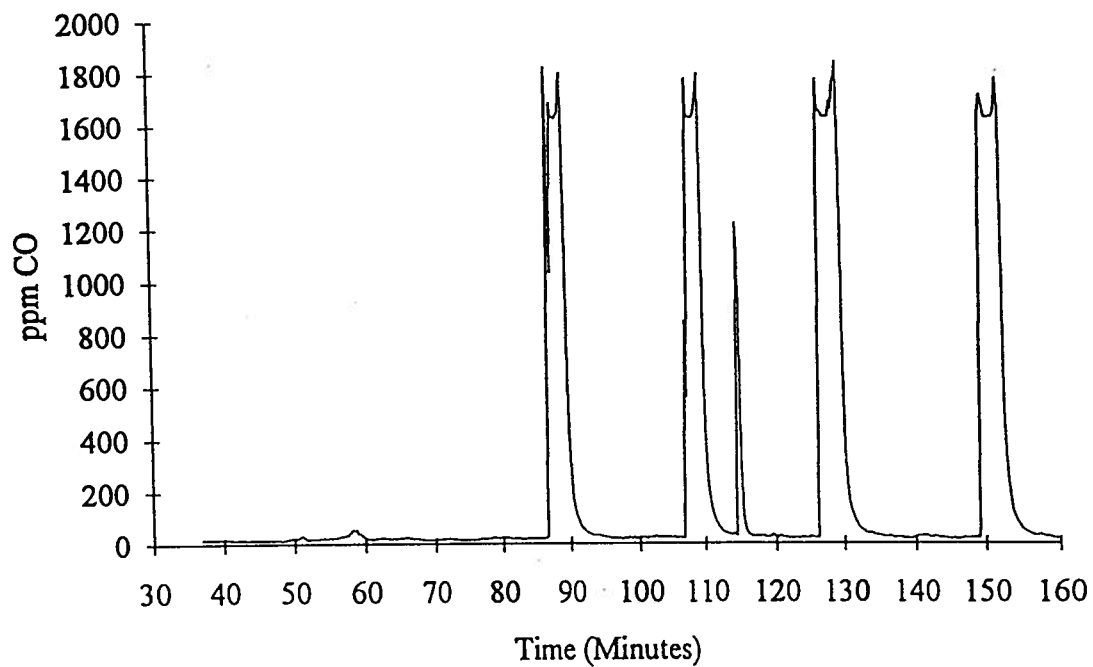
**Figure 6.16: Temperature Response - Run # 13, Test #3 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**



**Figure 6.17: Temperature Response - Run #14, Test #1 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S = 2, average solids loading = 120 kg/m<sup>3</sup>)**



**Figure 6.18: Rotary Valve Signal - Run #14, Test #1 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**



**Figure 6.19: CO Concentration - Run #14, Test #1 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**

$$A(q^{-1}) = 1 - a_1 q^{-1} - a_2 q^{-2} \quad (6.5)$$

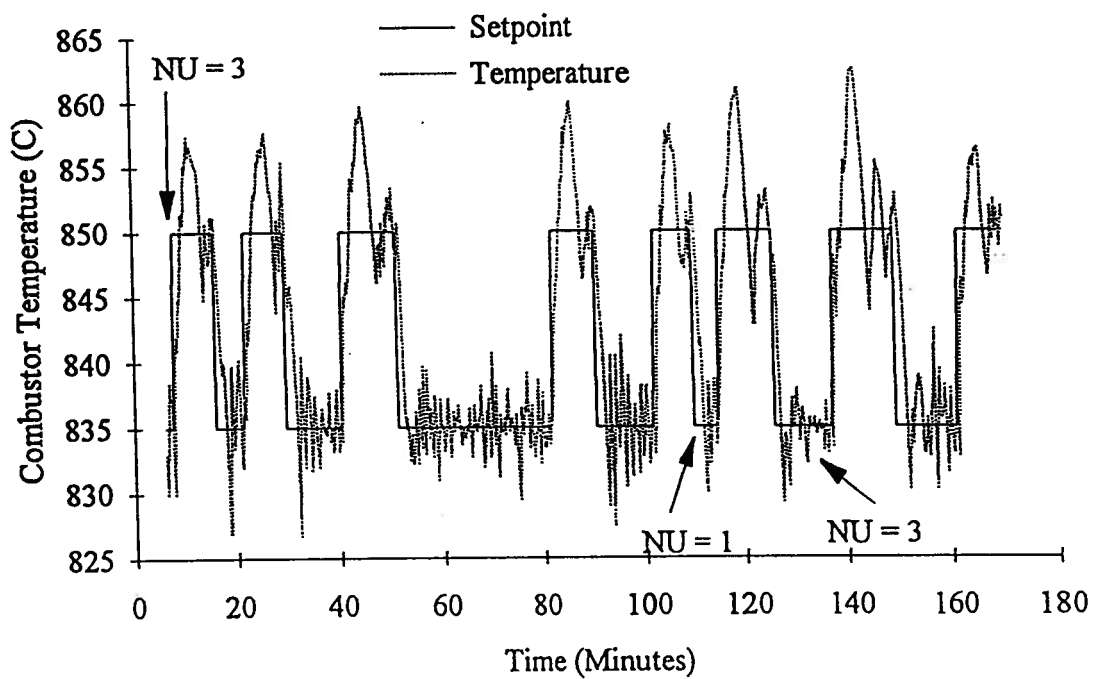
Figure 6.20 shows the combustor temperature during the first part of this final test. The large fluctuations about the setpoint are largely attributed to the deterioration of the bed material, which has been addressed in section 6.2.2.

The overall riser solids loading is presented in Figure 6.21. The inert solids used in this run consisted of old bed material from a previous experiment. The result of using this deteriorated material is poor control over riser solids loading. Typically, with a new bed, the loading is maintained within  $\pm 5 \text{ kg/m}^3$  of the setpoint. In this test, the variation ranged from  $\pm 15 \text{ kg/m}^3$  to  $\pm 30 \text{ kg/m}^3$ . The riser solids loading has a marked effect on temperature since it is directly related to the rate of heat transfer to both the cooling surface and the surrounding refractory. These swings in loading account for a large portion of the fluctuations in temperature, resulting in the large fuel controller action shown in Figure 6.22. The inclusion of a noise term in the controller model would have helped reduce control action, but was not implemented at this time.

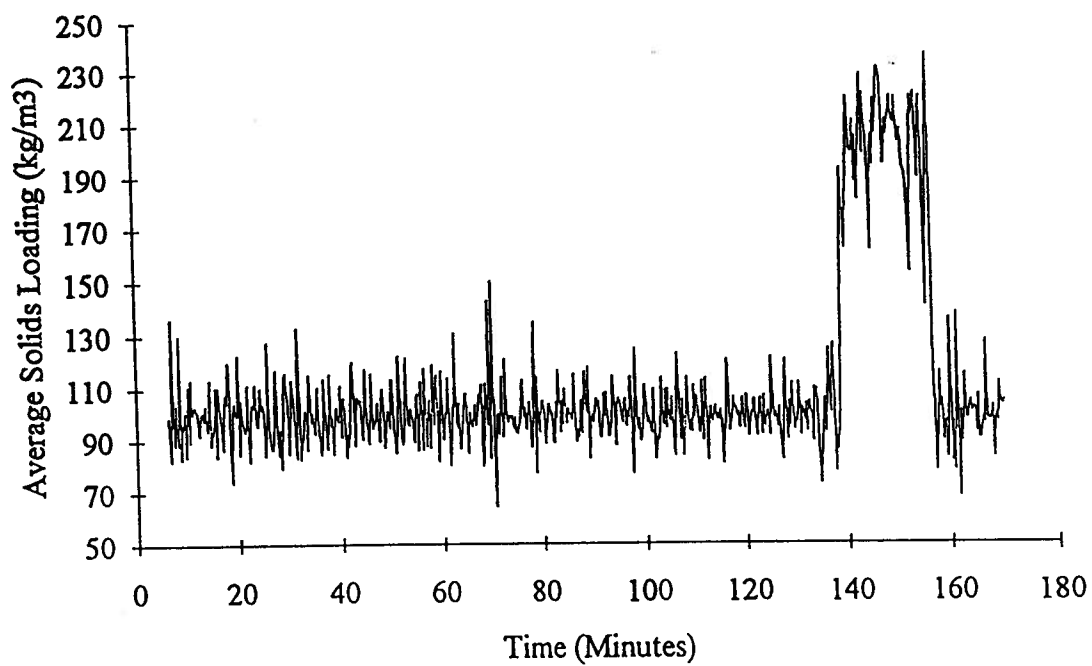
At time 170 min, the sample/control interval was changed from 15 to 30 s with no other adjustments at that time. A value for NU of 3 was in use and remained in place until 210 min at which point it was reduced to 1.

The adapted variables for this portion of the test are shown in Figure 6.23. The initial values in this plot were those to which the adapter had converged under the 15 s interval. These values were consistent with previous results. Upon changing the interval, there was an immediate response in the model parameters. At first, the response was erratic since the adaptor was still being influenced by information obtained using the shorter interval of 15 s. After about 20 minutes, the adapter began to settle down and converge on new values.

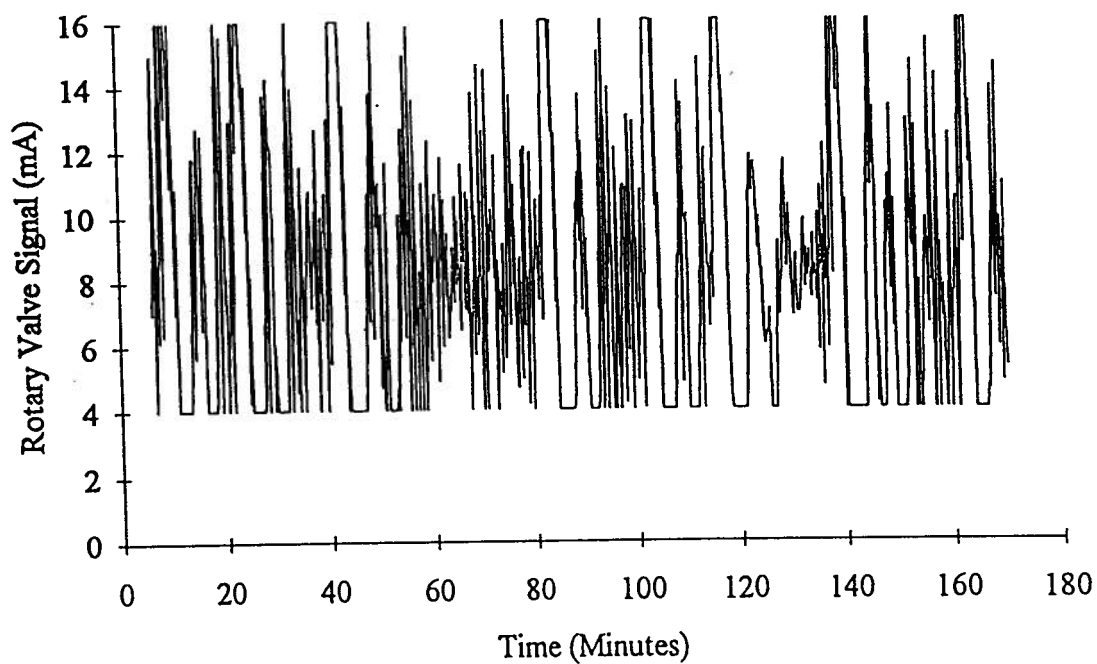




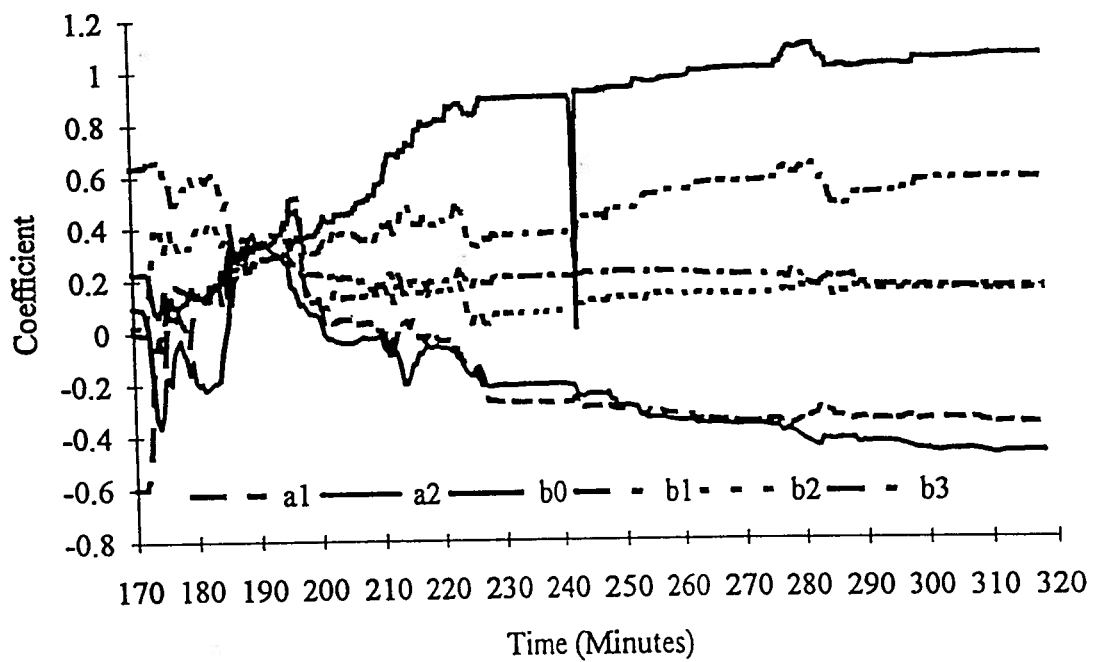
**Figure 6.20: Temperature Response - Run #14, Test#2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**



**Figure 6.21: Solids Loading - Run #14, Test #2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 100 kg/m³)**



**Figure 6.22: Rotary Valve Speed - Run #14, Test #2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S = 2, average solids loading = 120 kg/m<sup>3</sup>)**



**Figure 6.23: Time Variation of Model Coefficients - Run #14, Test #2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S = 2, average solids loading = 120 kg/m<sup>3</sup>)**

open loop poles @ 0.8683, -0.5183

Control does not appear to have improved during this initial period because of the erratic behaviour of the adapter. At 210 min, when the control horizon, NU, was returned to 1, a number of significant improvements began to take place. A combination of the reduced control horizon and tuning of model parameters resulted in significantly reduced control action (Figure 6.24), and notably better control over temperature (Figure 6.25). As the model tuned up, bed overfeeding (indicated by CO spikes in Figure 6.26) was virtually eliminated. The final CO spike, at 282 min in Figure 6.26, resulted from returning the control horizon to 3. This effect is also seen in the increased control action in Figure 6.24 where  $NU = 3$  was used from 275 to 285 min at which time it was returned to 1.

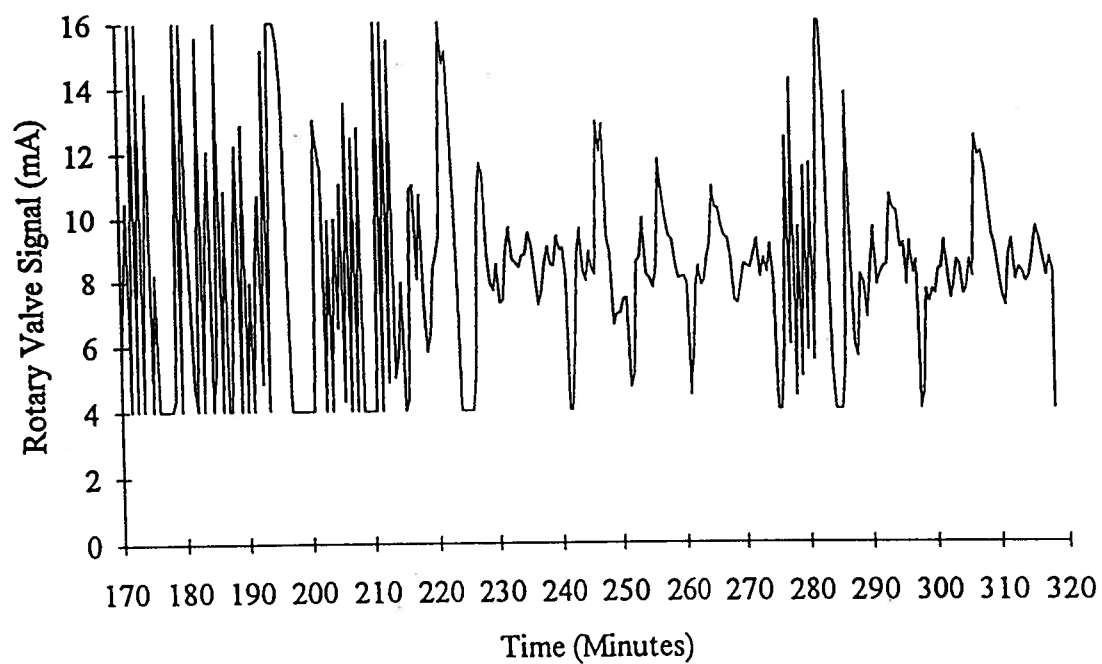
The final two step changes (Figure 6.25) from 850 to 825 °C and from 825 to 855 °C gave dramatically improved control. The simple act of increasing the sample/control interval to 30 s afforded much stricter control (even in the face of the solids loading fluctuations) and virtually removed the more serious difficulties of bed overfeeding and associated problems (poor emissions, loss of temperature, etc.).

#### **6.4 The Multi-loop Control Structure**

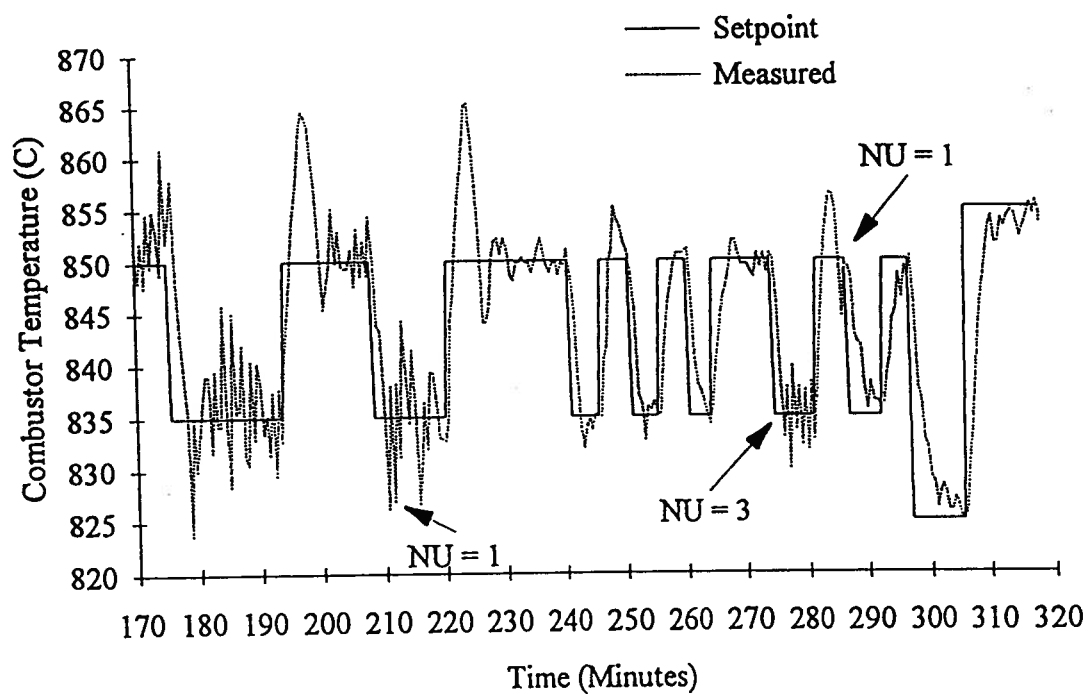
The final overall control approach adopted in this work accepts the fact that combustion control is a multivariable problem, but takes advantage of (1) differing time constants between the controlled variables, and (2) ranking of each variable being controlled. Practical application to industrial boilers was always kept in mind.

In the overall process of steam production, steam quality is of utmost importance. The design of downstream equipment is based on the required steam quality (pressure and temperature). The boiler must produce this steam at design conditions. When the CFBC unit is part of a power plant, there are continual changes in demand (from the power grid etc.) and the boiler must meet these varying steam flow requirements for electricity production.

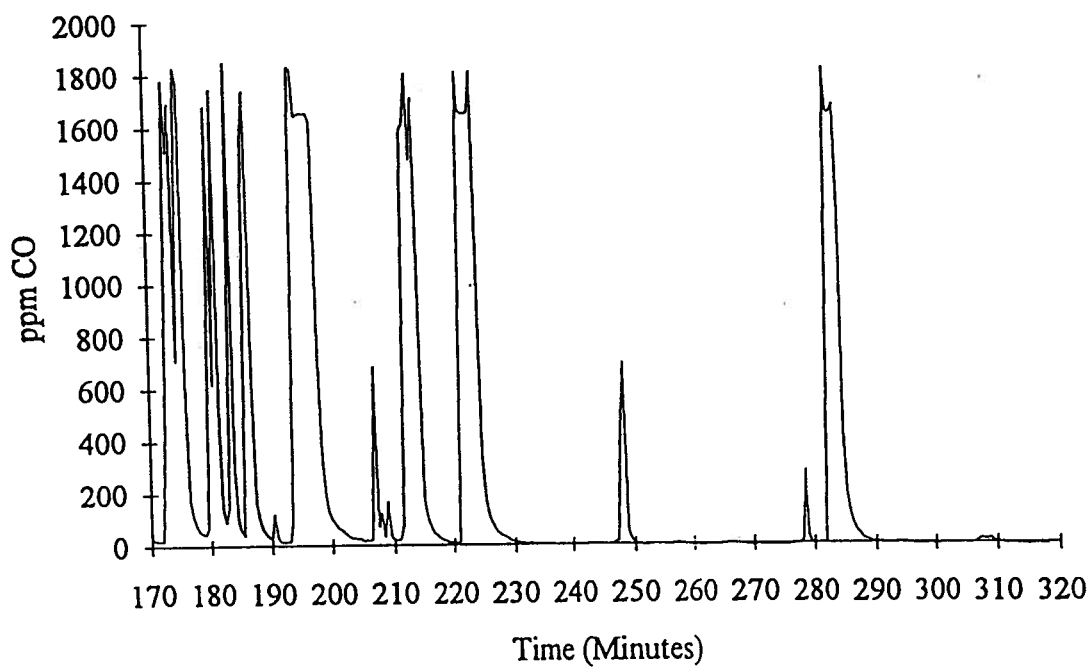
While it was not feasible to include steam generating facilities on the UBC pilot CFB combustion plant, an approach to the problem of steam quality was taken by analogy. The in-



**Figure 6.24: Rotary Valve Signal - Run #14, Test #2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**



**Figure 6.25: Temperature Response - Run #14, Test #2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S = 2, average solids loading = 120 kg/m<sup>3</sup>)**



**Figure 6.26: CO Concentration - Run #14, Test #2 (Highvale coal, GPC temperature controller test, gas velocity = 7.5 m/s, P/S =2, average solids loading = 120 kg/m<sup>3</sup>)**



bed heat exchanger was considered the boiler surface and the water outlet temperature from this heat exchanger was chosen as the variable of control.

Complete CFBC control can be reduced to a  $3 \times 3$  multivariable problem. In particular, specifications for excess air (or flue gas oxygen concentration), combustor operating temperature, and rate of heat removal can be achieved by adjusting air flow, fuel feed, and solids recycle rate. Temperature is important in maintaining conditions of safety and viability for combustion and to meet emissions restrictions. Flue gas oxygen concentration is indicative of source air utilization and combustion conditions while also being of critical importance in avoiding excessive generation of pollutants. By maintaining a given oxygen setpoint, one ensures that sufficient air is being supplied for combustion, while limiting the amount of excess air being used. Beyond the scope of this work, optimization can be applied to determine the best combinations of temperature and air flows for flue gas emissions optimization. In this work, however, it was desired to define specific values for these three variables to be achieved and maintained.

In order to provide control over these variables, three combustion side inputs were identified - riser solids loading, fuel feedrate, and total air flow. The interaction of each of these inputs affects, in one way or another, each of the outputs. This is illustrated in Figure 6.27. The riser solids loading directly influences the rate of heat transfer to the heat exchanger (Equation 4.26). It also determines how much of the bed carbon inventory is in the combustion zone at any given time, thereby affecting both temperature and oxygen utilization. The fuel feedrate is required to maintain a certain inventory of carbon in the bed. This carbon is one factor influencing the rate of combustion, which also affects hot water temperature through radiation (Equation 4.26), combustor temperature (via the amount of heat generated), and oxygen concentration (because of reaction). The air flow determines the excess oxygen in the system and hence the flue gas oxygen concentration. Oxygen availability impacts on the combustor temperature through the rate of combustion (Equation 4.15). As well, the total air flow (as well as the primary-to-secondary air ratio) determines the gas velocity within the riser

**PID Controller Settings**

**Gain**

**Integral Time**

**Derivative Time**

**Oxygen**

**8 kg·hr<sup>-1</sup>/‰O<sub>2</sub>**

**150 s**

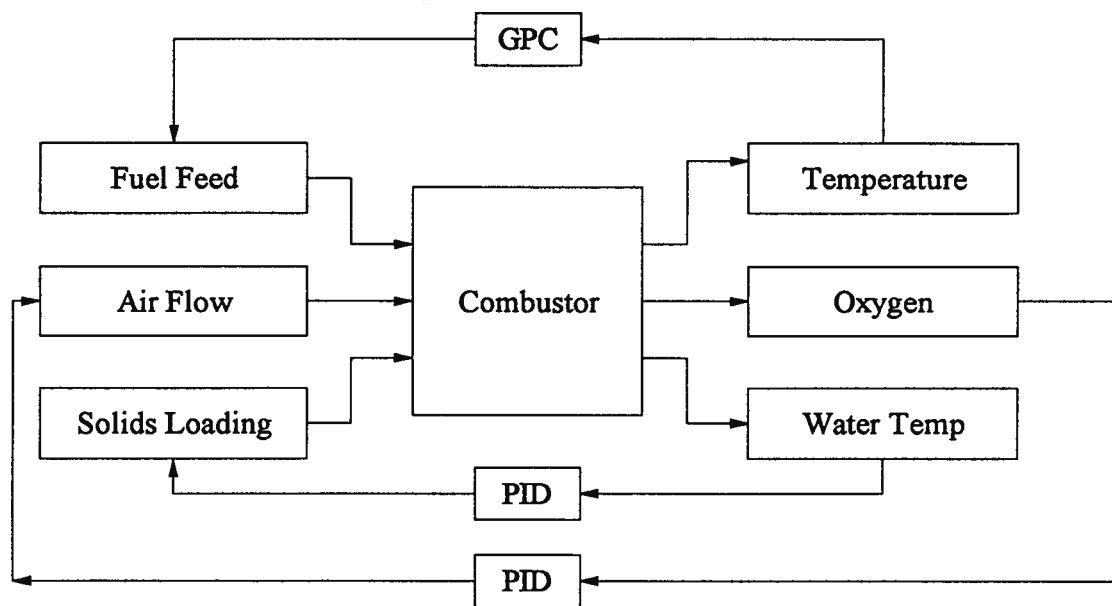
**0.25 s**

**Water Temperature**

**0.26 mA/°C**

**30 s**

**0.06 s**



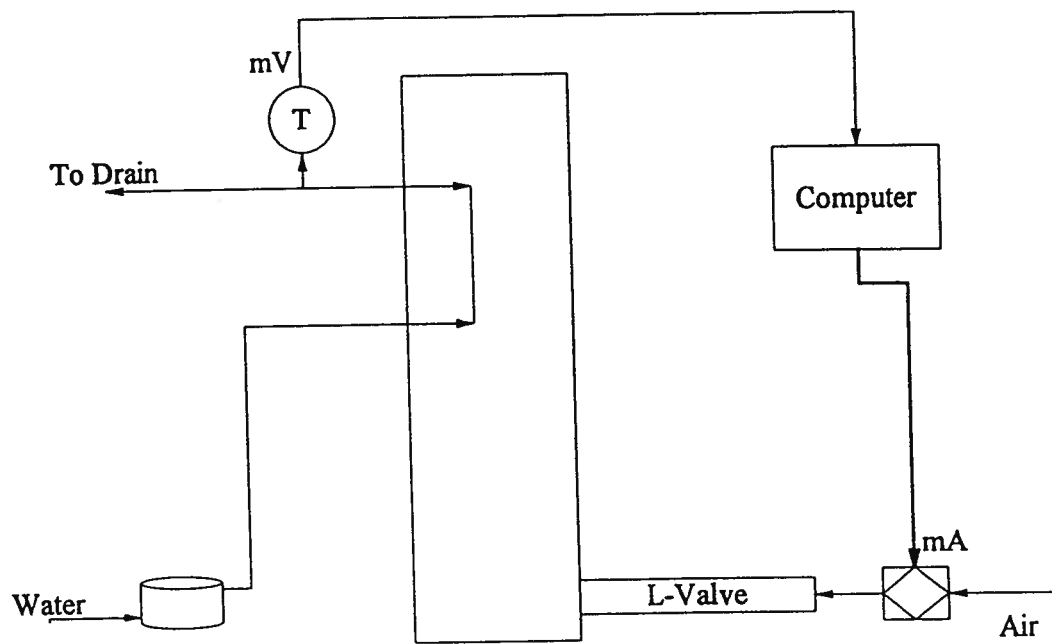
**Figure 6.27: Investigated Control Structure**

for a given temperature, which has a direct effect on the axial solids distribution within the riser. The interaction of these inputs is significant. The approach taken here, based on experience, is to minimize the effect of these interactions by choosing appropriate individual loops.

Based on the steam production analogy, the primary goal is to maintain a desired heat exchanger water outlet temperature. This control loop is shown in Figure 6.28. It is not only important that this controller be able to meet a setpoint requirement, but it must also respond quickly to demand changes (i.e. changes in water throughput). Riser solids loading (Figure 6.2) is the input which most quickly and most significantly affects this variable. The solids loading controller is very fast in meeting setpoint demands (response times of 10 to 15 s) and its extension to water temperature control is logical.

The philosophy here is that, with a controlled solids recycle device, one can control the solids loading in the riser and thereby directly change the rate of heat transfer to the boiler surface permitting fast, short term, control of steam production. In a conventional boiler, any change in steam demand is satisfied by a change in the amount of fuel and air fed to the system. It may take a relatively long period of time to respond to these demands due to fuel buildup and to heat sinks within the process. Having the ability to control the solids recycle rate allows for quick adjustments to new heat demands. The solids loading at the heat exchange surface accounts for typically 30 to 50% of the heat transfer coefficient to that surface. The loading can be rapidly adjusted to meet and maintain most changes in steam demand, while longer term transients in response to fuel feed changes can be used for long term stabilization of the plant. Superimposed on this controller is a limit checker which ensures that the riser solids loading remains between 50 and 180 kg/m<sup>3</sup> to maintain the combustor in operable regions.

Total air flow is the most obvious choice to maintain the flue gas oxygen concentration at its desired level. This loop is shown in Figure 6.29. As optimization is not an objective, a detuned PID controller works acceptably well. While concentration deviations



**Figure 6.28: Schematic of Heat Exchanger Outlet Water Temperature Controller**

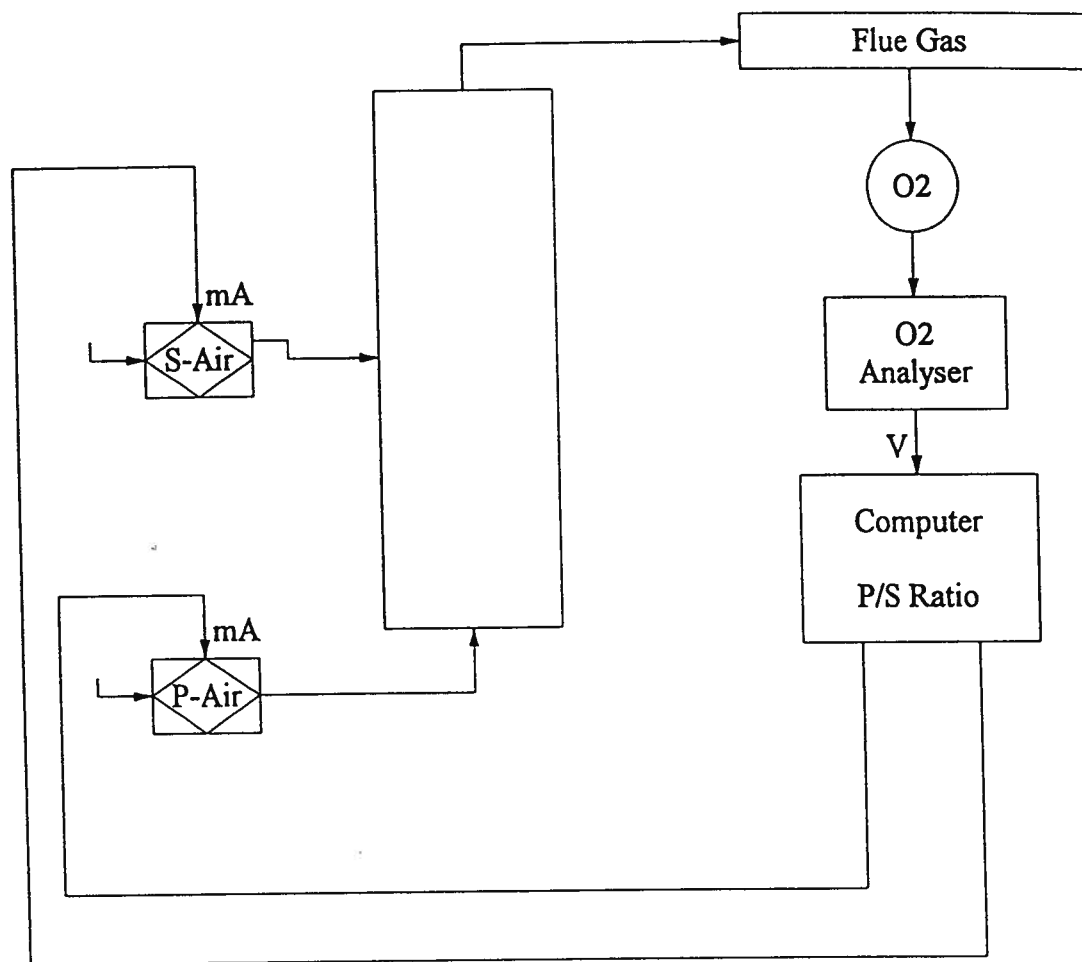


Figure 6.29: Schematic of Flue Gas Oxygen Concentration Controller

greater than 1% may occur at times, this controller ensures that there is a sufficient supply of oxygen available while limiting addition of excess air. By detuning this controller, large corrections in air flow are avoided, thereby serving to limit over/under-supply and eliminating any noticeable effects that faster changes would have on the heat exchanger outlet water temperature due to changes in the solids loading profile. This loop shows response times of 2 to 3 min.

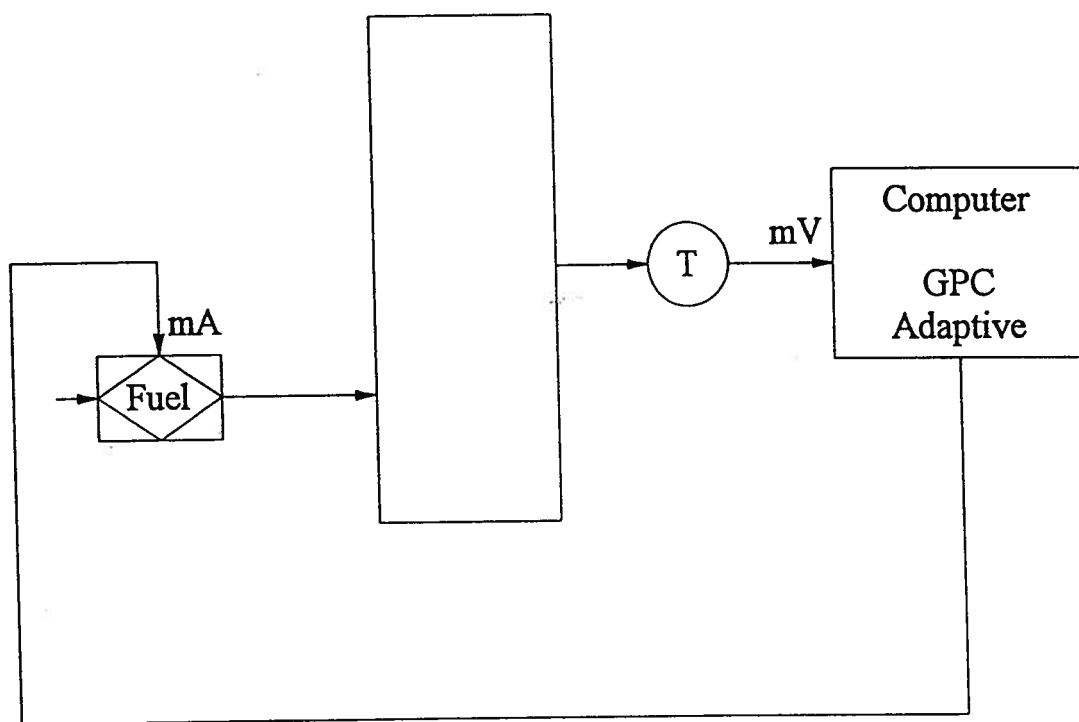
The combustor temperature is controlled by a GPC controller with fuel-feedrate as shown in Figure 6.30. The use of the GPC controller allows the system to be sampled at significantly longer time periods (2 minutes) allowing the controller to ignore higher frequency noise. This controller reacts to long term effects of fuel feedrate on temperature and, having a response time of 15 min, is somewhat removed from the shorter term actions of the other two controllers.

#### **6.4.1 Tuning the PID Loops**

The two PID controllers were tuned on-line by trial and error. Noise and time restrictions made conventional tuning impossible. Although probably not optimal, the tuned parameters provided more than satisfactory control as illustrated below. The oxygen controller employed a gain of  $8 \text{ kg}\cdot\text{hr}^{-1}/\% \text{O}_2$ , an integral time of 150 s, and a derivative time of 0.25 s, and the water temperature controller had a gain of  $0.26 \text{ mA}/^\circ\text{C}$ , an integral time of 30 s, and a derivative time of 0.06 s.

#### **6.4.2 Model Identification for GPC Control**

Prior to the implementation of a model based controller, proper identification of the process order and dead-time and approximate values for the model coefficients are essential. As demonstrated in the initial work with GPC in this research, this may not be such a critical step in the overall design scheme.



**Figure 6.30: Schematic of Combustor Temperature Control Loop**

One of the significant advantages of GPC, as discussed in detail in Chapter 5, is its ability to maintain robustness in the presence of plant/model mismatch (error), variable or fractional dead-time, and plant order over/under-estimation. The robustness of the controller used in the present work in the face of poor identification suggests that the identification step may not be as important as in earlier cases. Optimality is not attainable without proper identification, but, when faced with a process that does not readily give up its dynamic properties, it is encouraging that the GPC algorithm attained some level of control.

This section presents the work involved in identification. The approach used here was open loop PRBS testing. These results are then compared to the on-line adaptation performed by the adaptor within the GPC controller.

Although properly designed PRBS testing should provide a plenitude of dynamic information for the identification of a plant model, reliance was placed upon on-line adaptation to de-emphasize the identification stage. This was because of difficulties in obtaining acceptable dynamic information from the PRBS tests. This problem is not associated with the theory of identification, employing a PRBS signal, or time series analysis of the results, but rather is due to a combination of gas/solid combustion and pilot plant limitations.

CFB combustion is a somewhat non-linear process. Conventional model-based control (GPC included) uses a linear model of the plant. One advantage of employing a PRBS signal at the identification stage is that identification may be performed about a fixed operating point. This allows a linear model to be fitted at this point. Unfortunately, this application is both transient and time-variant in nature. One is concerned with moving the process from this operating point in response to changing demand. In the present work, the GPC controller was assigned to maintain combustion temperature by means of fuel feed. As already noted, this single variable coupling is a simplification of the solution since temperature is a function of air flowrate and riser solids loading as well as fuel feedrate. Although, conceptually, it should be possible to perform a PRBS test about a tight operating point, the linear model obtained would be too specific for these particular conditions to be generally applicable for the process.



Alternatively, a more dynamic PRBS test, which moves the process throughout its operable range limits, could be performed. Such a test would provide a linear model which is an average of the expected dynamics of the process. The advantage of the open loop PRBS is that it provides this average fit, as opposed to the adaptive algorithm which, given its recursive nature, more heavily weighs newer data. An important distinction between parameter fitting through time series analysis of a PRBS test and through adaptation is that time series analysis fits all available data to a given model, while the true purpose of adaptation is to change the model parameters subject to time variants in the process dynamics. Conceptually, a long term PRBS could traverse the entire operating range of the system in a manner not only to provide dynamic information from all operating points, but also to provide weighting of these points. A PRBS test, as outlined above, must move the process extensively throughout the operating region in order to provide enough dynamic information to determine a "mean" linear model.

As discussed above, it proved to be difficult to ensure continuous, trouble-free operation of the pilot CFBC unit for more than 10 hours. Unfortunately, this time period is insufficient to permit the collection of completely adequate information for identification. It is estimated that 40 to 50 hours of continuous testing would be required to properly identify this process. Because of the limited information available, these tests tend to indicate relatively high order models. This, in itself, is not a serious drawback as this controller was intended to perform during switches in fuel type and, as such, higher order models may be needed to account for dead-time changes.

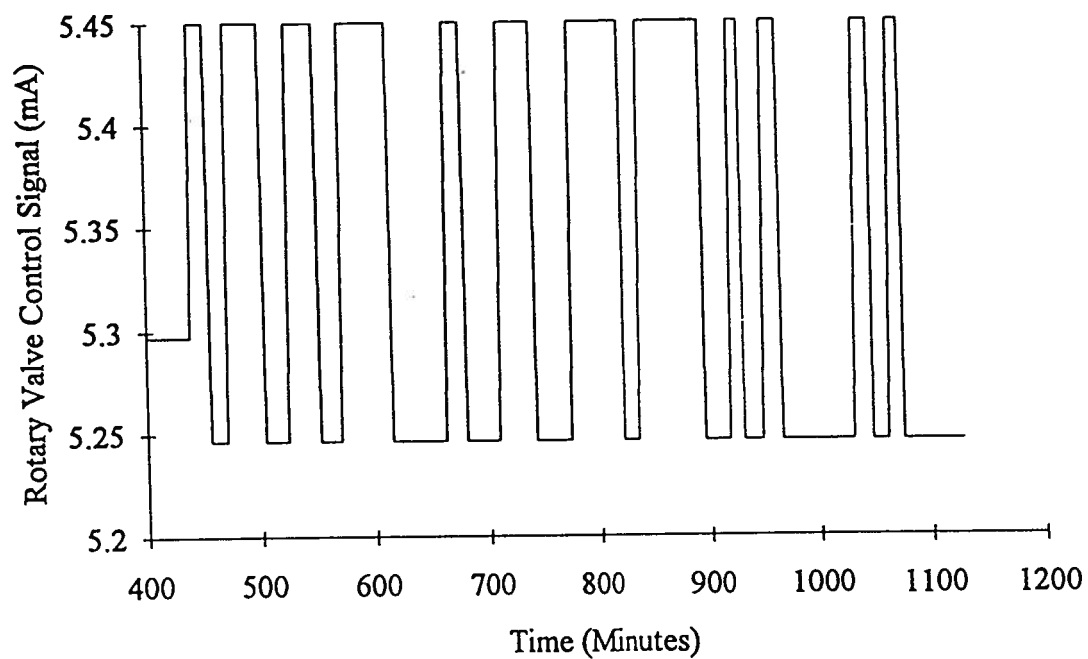
The identification methods employed are outlined in Chapter 5. These consist of PRBS stimulation of the fuel feed rotary valve speed and measurement of the corresponding response of the combustor temperature. As discussed earlier, this is a multivariable problem in which each of the controlled inputs - fuel feedrate, air flow rate, and riser solids loading - affects each of the outputs - combustor temperature, water outlet temperature, and flue gas oxygen concentration. Since changes in fuel feedrate affect all of the output variables and, in

turn, the other controllers indirectly cause changes in temperature, during these identification tests the water temperature and oxygen controllers were employed in order to include their effect on temperature in response to changes in the fuel feedrate. Although, intuitively, this should result in a noisier and less distinct process model involving the effect of fuel feedrate on combustor temperature, it does result in a closer approximation of the true process during control.

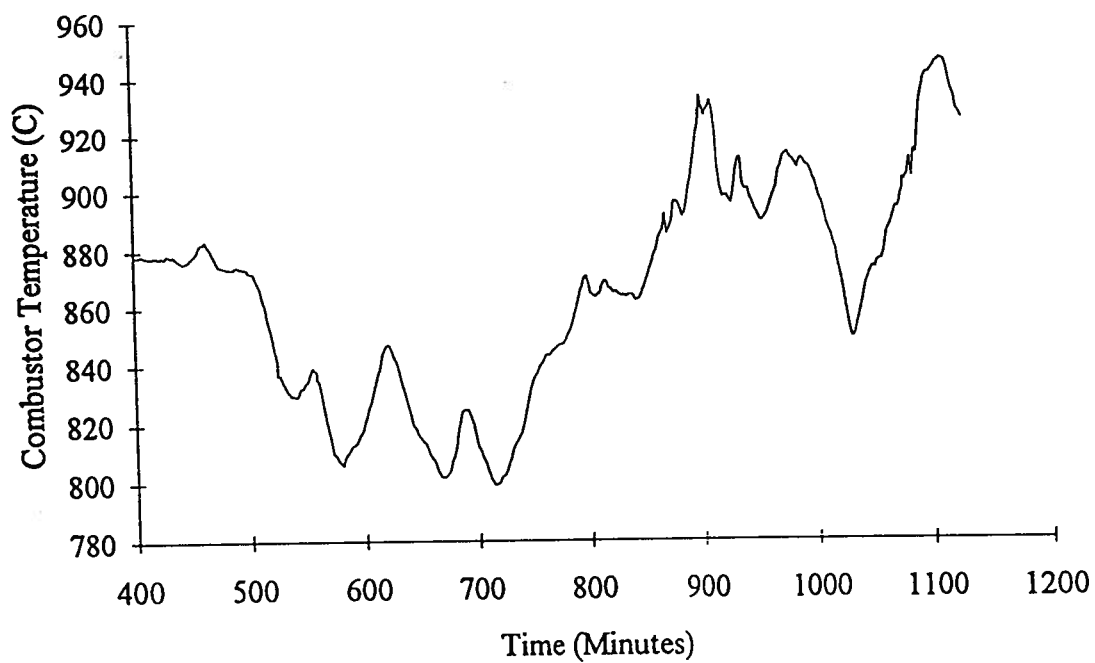
Run #48 illustrates results obtained from PRBS tests performed on the UBC pilot CFBC. The fuel used during this run was Syncrude Coke. The PRBS stimulation of the rotary valve speed is shown in Figure 6.31. The system was sampled every 2 minutes, while every 16 minutes the random choice to switch the rotary valve setting between 5.45 mA or 5.25 mA was made.

The measured combustor temperature during this test is shown in Figure 6.32. The temperature ranged from a minimum of 800 °C to a maximum of 950 °C. While this range was greater than desirable, and far larger than conventional operating limits, it did remain bounded throughout the test. The wide temperature range was caused by feed flow inconsistencies, not over-stimulation of the system by too great an amplitude of the PRBS. As can be seen in Figure 6.31, the rotary valve speed is very low. At these levels, motion of the valve was almost imperceptible. Syncrude Coke is a very fine fuel whose packing characteristics can vary greatly over time leading to extensive variations of fuel feed. As well, the rotary valve does not provide a perfect seal between the fuel hopper and the combustor. For a fuel such as Syncrude Coke, there is always leakage around the rotary valve to some extent. The combination of these two effects can create significant fluctuations in the fuel feedrate.

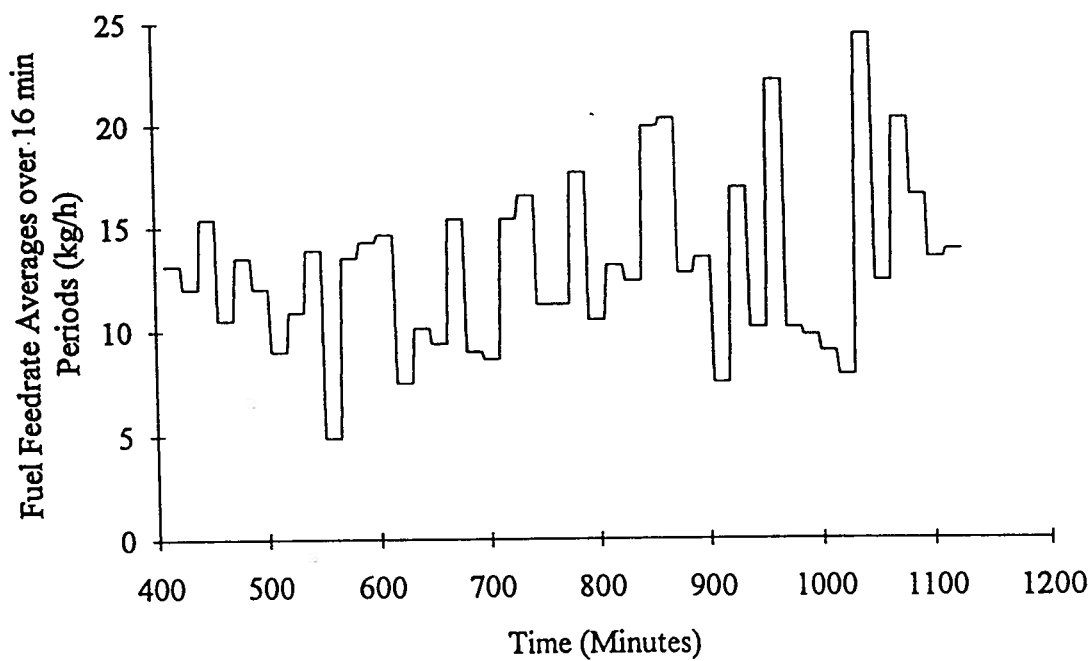
Figure 6.33 presents the fuel feedrate during this test computed by taking the loss-in-weight from the fuel feed hopper over the same 16 minute intervals as the switch intervals for the PRBS test. There is clearly a trend toward higher feedrates at the end of the test. This trend corresponds to that seen in Figure 6.32 where there are higher temperatures at the end



**Figure 6.31: Rotary Valve Signal during PRBS Test Run #48 (Syncrude coke)**



**Figure 6.32: Temperature during PRBS Test Run #48 (Syncrude coke)**



**Figure 6.33: Fuel Feedrate during PRBS Test of Run #48 - Averaged over 16 Minute Intervals (Syncrude coke)**

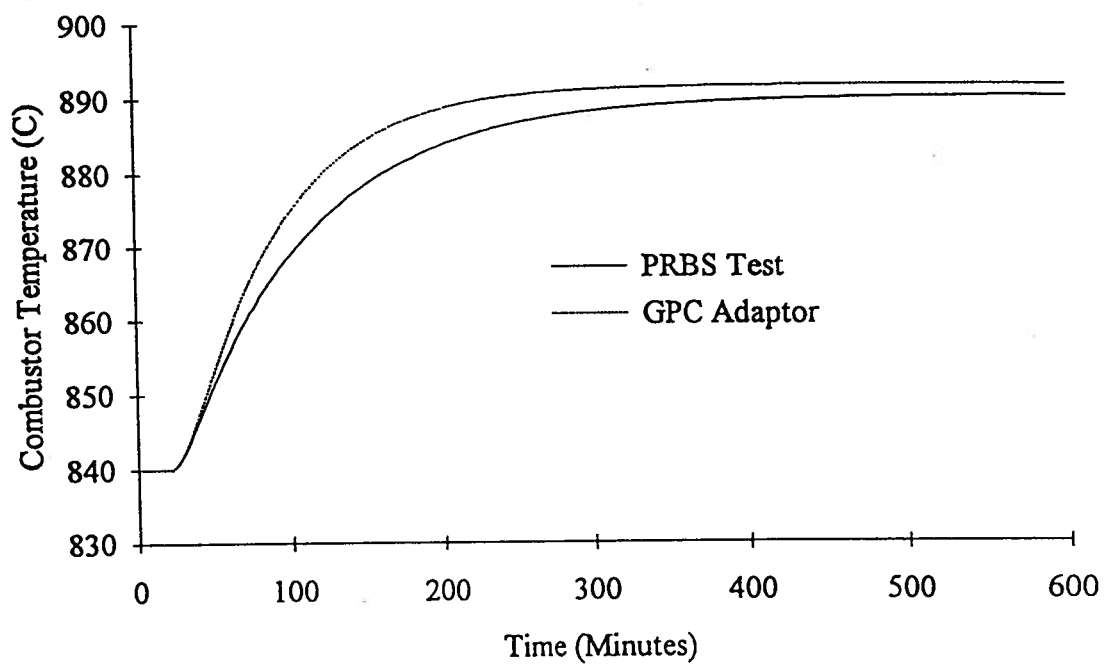
of this test. These fuel feed flow inconsistencies are an uncontrollable disturbance within the process and, in a sufficiently long test, would contribute to the noise term of a properly identified noise model.

Time series analysis was applied to the rotary valve control signal and to the temperature response to identify a transfer function model for the process. Because these tests are shorter than ideal, statistical significance suffers. Consequently, there is a tendency for identified models to be of relatively higher order than would be expected from tests of longer duration.

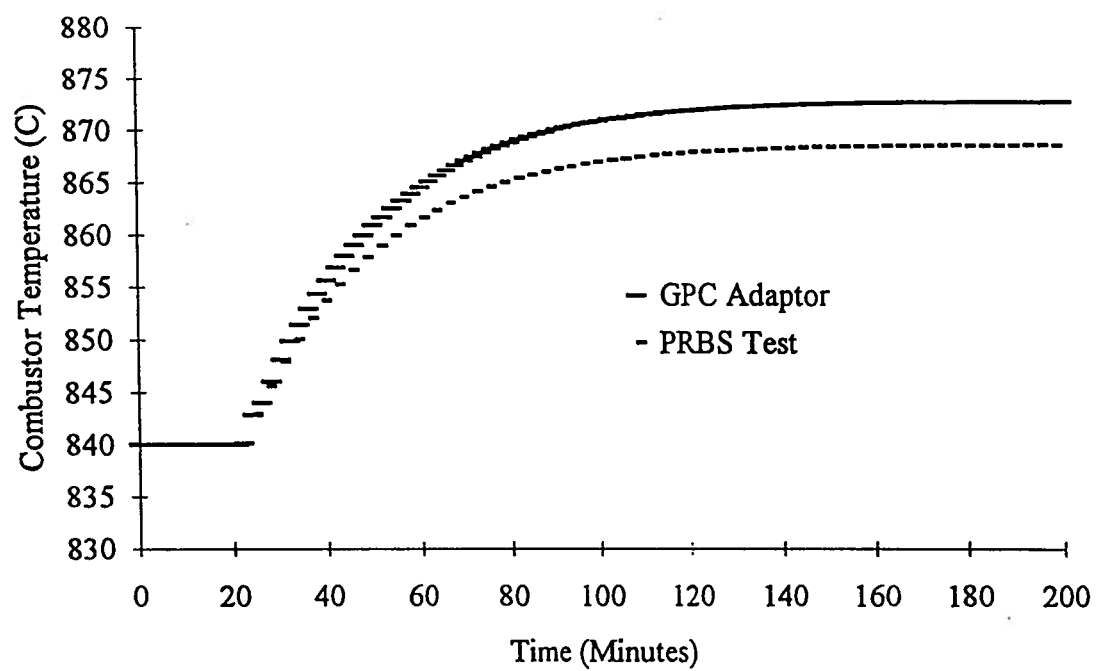
It is probably more important to view the results of these tests in terms of the behaviour of the adaptor in the GPC controller. The PRBS identification program is designed to generate dynamically informative data and the most efficient analysis of this information. If the design is correct, then the results of the PRBS testing can be used to judge the performance of the adaptor used in the controller.

During the forcing of the CFBC process with the PRBS sequence in Run #48, the adaptive portion of the controller was implemented so that the adaptor could also fit a model to the process. Figure 6.34 compares the step response of the adapted CARIMA model to the transfer function model from the time series analysis. The predicted gains of the two models are within 2 °C of each other, while the CARIMA model predicts a slightly faster response. Possible explanations for the differences have been addressed above. In brief, the differences are related to the fact that the time-series analysis provides equal weighting of information over the test period, while the adaptor more heavily weighs data from the later portion of the test. Conceptually, the step response of the PRBS is representative of an average model for the period while the CARIMA model concentrates on the behaviour of the process at the end of the test.

Perhaps more important is a comparison of the adaptor under normal situations with the PRBS test and analysis (ie. a comparison of the adapted model for a less dynamically stimulated controls test with that obtained through time series analysis). Figure 6.35 provides



**Figure 6.34: Comparison of Step Response of Models Obtained from Time Series Analysis and the GPC Adaptor - Run #48 (Syncrude coke)**



**Figure 6.35: Comparison of Transfer Function Step Responses from PRBS Test (Run #22) and Adapted Variables (Run #40)**



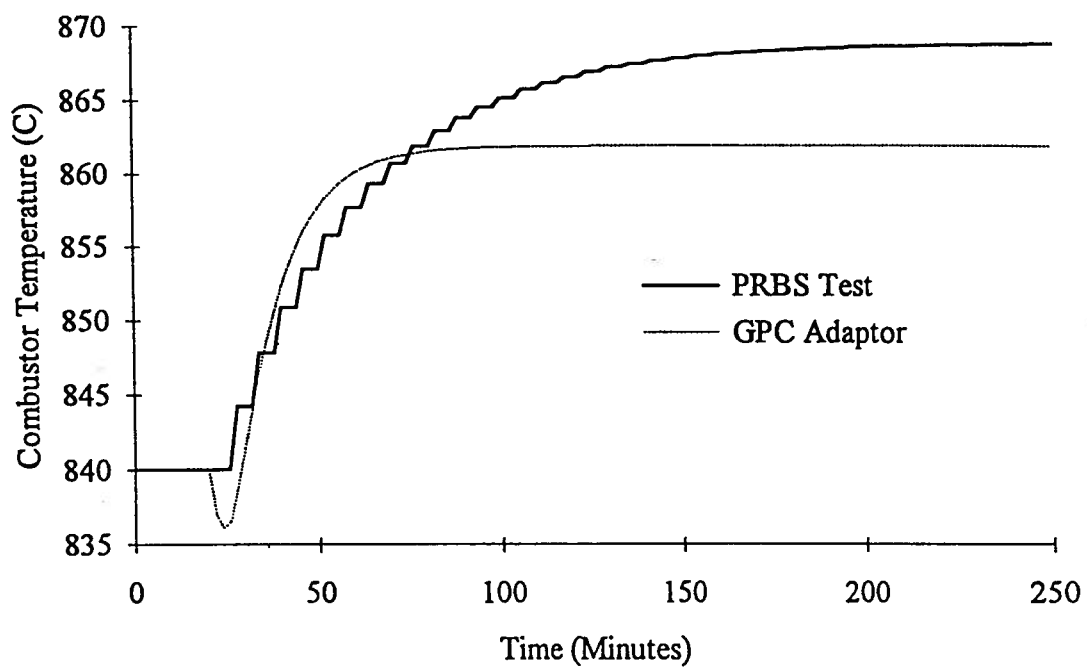
such a comparison. These tests employed Highvale coal as fuel. Run #22 was a PRBS identification test, while Run #40 was designed to test the GPC controller. The two models differ primarily in their predictions of the gain. Otherwise, the responses are almost identical. The favourable agreement in response is encouraging as these models were developed from two different runs in which experimental conditions differed substantially. The differences in gain could be due to run changes, data averaging by the time series analysis, or model error. Whatever the source, the difference is less than 5 °C, insignificant for the process in question.

The final comparison involved Conoco coke as fuel. Figure 6.36 provides the step responses for the transfer function model from time series analysis of a PRBS and the CARIMA model fitted by the adaptor during a controls test. Significantly differing responses are predicted. There is an approximately 8 °C difference in the gains, and the adapted model expects an inverse response from the process. Again these conditions could be associated with differing characteristics of different runs.

Figures 6.34, 6.35, and 6.36 all demonstrate that the PRBS techniques and the GPC adaptor provide similar model responses for the process. The time series analysis provides a time-averaged model, while the adapted CARIMA model is again more indicative of current, or final, conditions. As CFBC is a time-variant process, it is perhaps more appropriate to rely on the adaptor, than on a time-averaged model, to provide up-to-date information on the current behaviour of the process.

The primary role of the PRBS test is the determination of model order. Its effectiveness here is reduced by insufficient data. If longer tests had been possible, lower order models would have become statistically significant. Run #22 and Run #48 both provided third order transfer function models, while Run #44 resulted in a first order transfer function. Inspection of the responses for these three models indicates that they are possibly all first order. Lack of statistical significance militated against this reduction in model order.

The results of this section indicate the relative importance of both the identification procedure and adaptive control. Ideally, identification should provide a low order transfer



**Figure 6.36: Comparison of Step Response of PRBS Test (Run #44) and Adapted Model (Run #45)**

function model representative of the process. This low order model would be used in the adaptor which would adjust the coefficients of the model to account for time variations of the process. Higher order CARIMA models were used in this work because: (1) short PRBS testing would not provide lower order models; (2) the controller is expected to function during fuel type changes in which the behaviour of the process is unknown; and (3) as shown by Figure 6.36, it is possible that model order may change between runs, even for the same fuel.

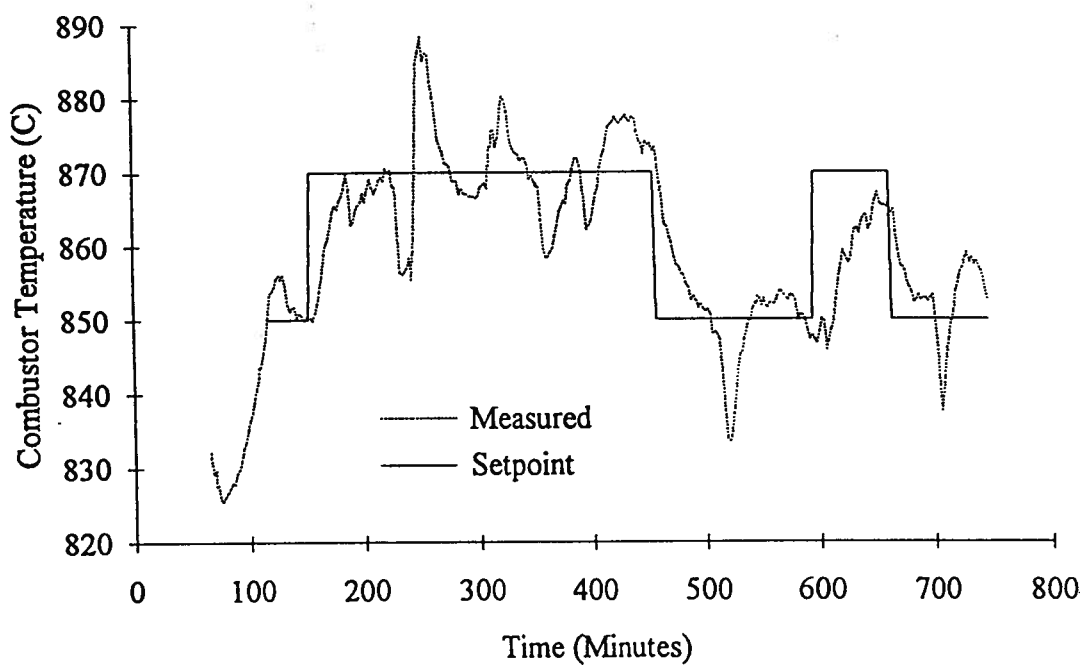
#### **6.4.3 Controller Testing**

Run #49, performed in March, 1994, was designed to demonstrate the control structure with Syncrude coke as fuel. It was to provide implementation of the GPC controller with step changes in combustor temperature, water outlet temperature, and flue gas oxygen concentration.

The controller testing took place until 748 minutes into the run. Except for the last 150 minutes of this test, the adaptive portion of this controller was deactivated. As indicated earlier, numerous difficulties often arise during a run related to equipment failures rather than controller performance. Unfortunately, if a failure occurs while the adaptor is monitoring the process, drastic, unreasonable changes are forced on the control model, inadvertently crippling the GPC controller. Run #49 was carried out with the model identified from Run #48 (described in Section 6.4.2) without adaptation. After demonstration of this controller, the adaptor was employed in the final 150 minutes to indicate the adequacy of this model for this particular run. The results of this adaptor test are presented below.

##### **Step change combustor temperature from 850 to 870 °C (150 to 215 min):**

The first dynamic change was a step increase in temperature from 850 to 870 °C at 150 min. The temperature response shown in Figure 6.37 appears acceptable with approximately 20 minutes required to reach the new setpoint. The Syncrude coke is a very



**Figure 6.37: Temperature during Run #49 (Syncrude coke)**

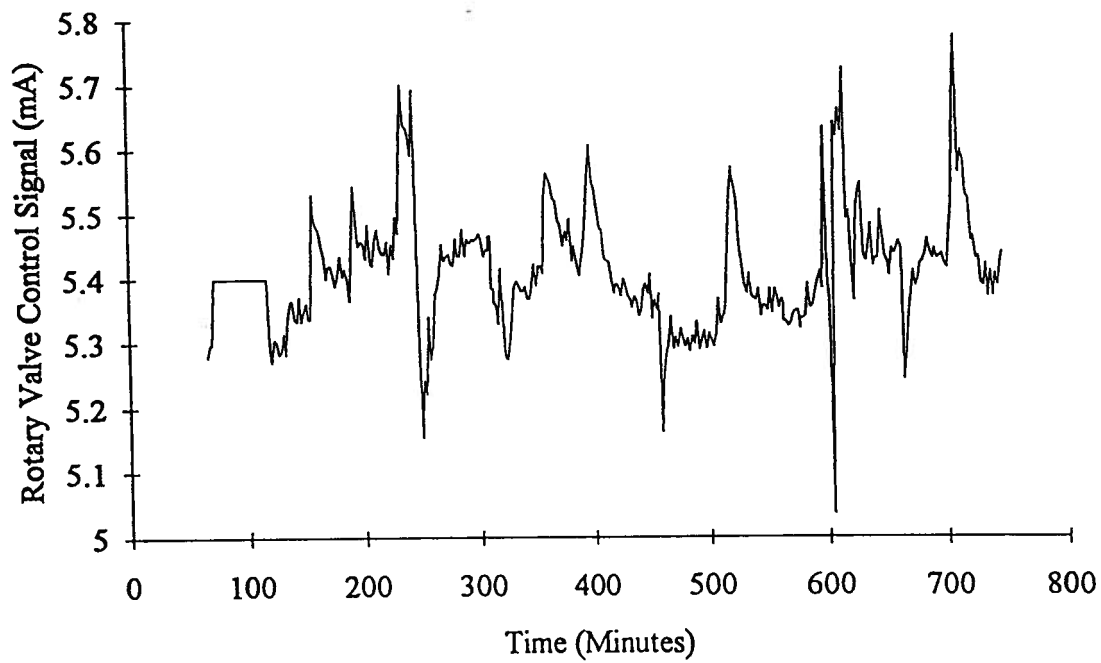
fine, high heating value fuel. As such, with the current feed system, the speed of the rotary valve was well below 1 RPM, with much of the fuel forced through the valve due to the positive pressure in the fuel hopper. This probably caused an added noise component due to uncontrollable fuel flow fluctuations.

While the initial temperature response to the step change is acceptable, at 185 min there was a drop of approximately 5 °C in the combustor temperature. This sudden loss is attributed to some form of noise whose source is unknown. The controller recovered from this error quite acceptably.

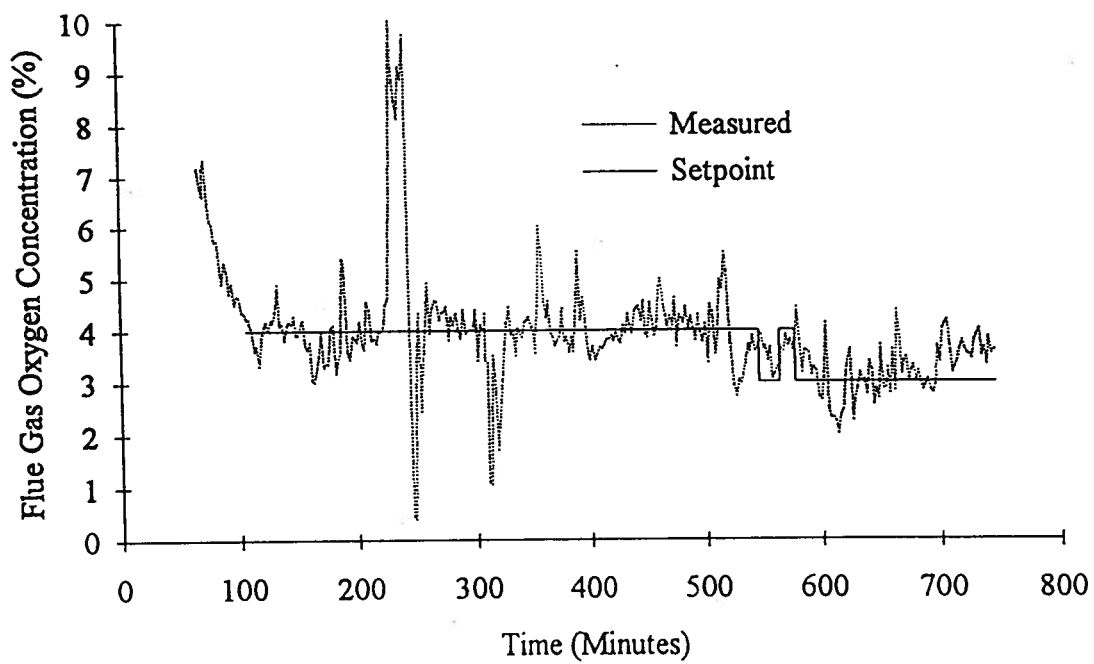
At 210 min, there was a 15 °C loss of temperature from the setpoint caused by a blockage in the fuel feed system. Because of this temperature drop, the controller increased the rotary valve speed in an attempt to feed more fuel to the system. The rotary valve control signal is shown in Figure 6.38. Because the controller had increased the speed of the rotary valve, overfeeding of the system occurred as soon as the blockage was cleared. This resulted in the overshoot of temperature from which the controller was able re-establish the setpoint in about 20 minutes.

During the fuel blockage, the flue gas oxygen dramatically increased to a high value due to reduced combustion as shown in Figure 6.39. In response to this increase, the oxygen controller reduced the riser gas velocity to a minimum as shown in Figure 6.40. Once the blockage was cleared, the oxygen controller quickly re-established the oxygen setpoint.

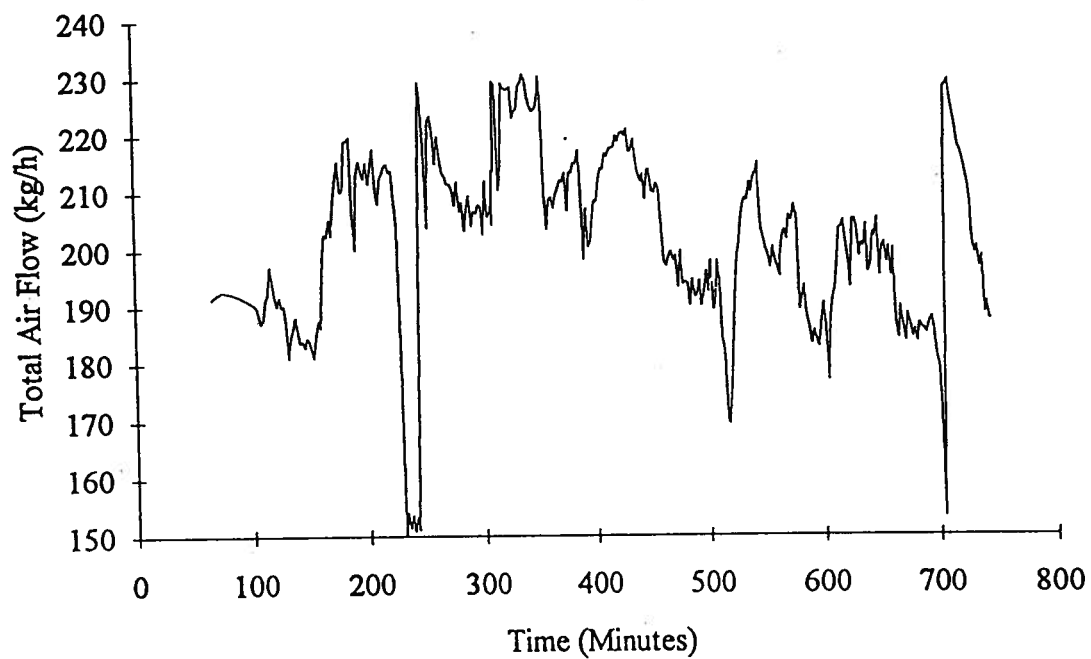
The effect of the fuel blockage on the water temperature can be seen in Figure 6.41. Although a definite disturbance can be seen at the time of the blockage, the controller was able to maintain the setpoint of water temperature within a few degrees Celsius. Figure 6.42 shows the overall riser solids loading as the variable responsible for maintaining water temperature. There was a sharp increase in the solids loading at the time of the blockage. This response was mostly due to the loss in gas velocity rather than a result of the drop in combustion temperature. As the gas velocity was reduced, there was a tendency for the solids profile to shift. Given the constant overall riser solids loading, as the gas velocity was



**Figure 6.38: Rotary Valve Signal during Run #49 (Syncrude coke)**

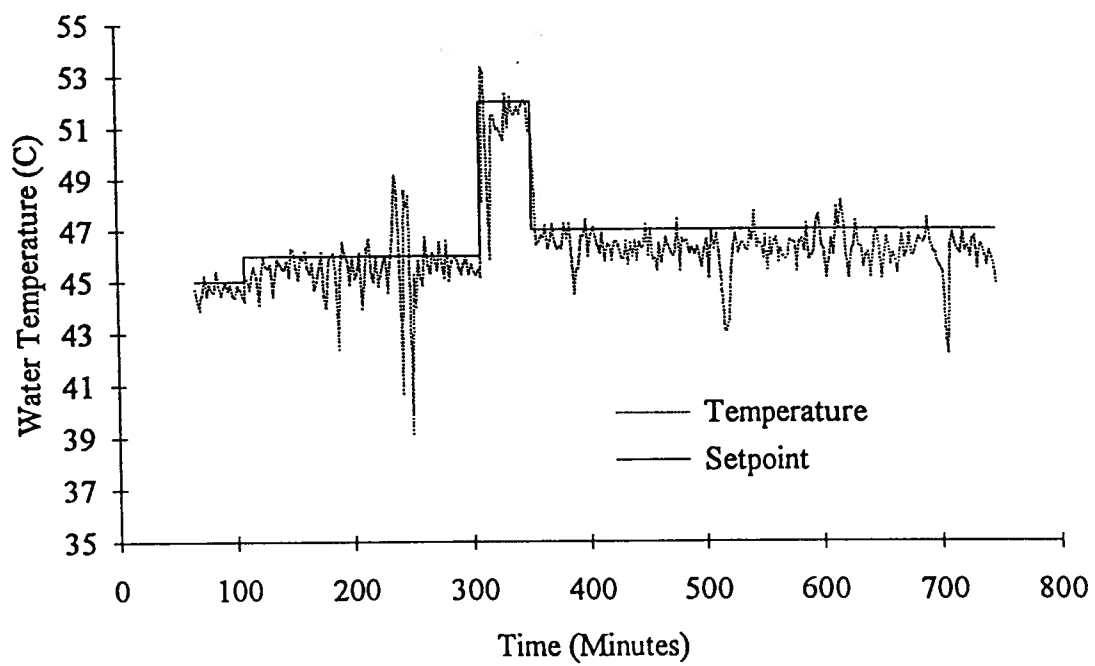


**Figure 6.39: Oxygen during Run #49 (Syncrude coke)**

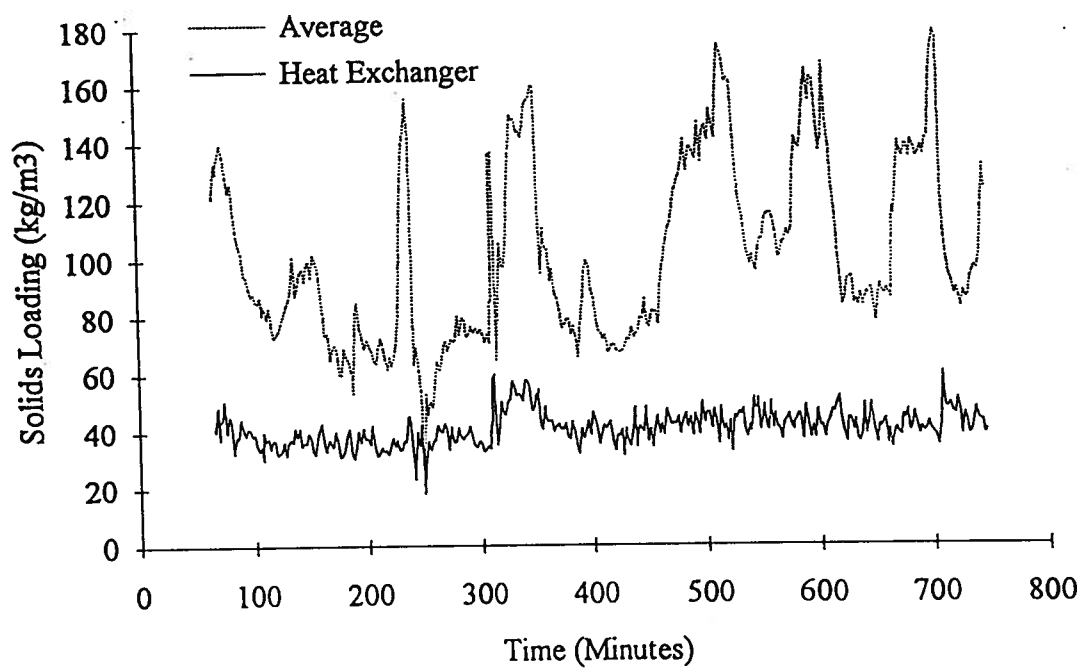


**Figure 6.40: Air Flow during Run #49 (Syncrude coke)**





**Figure 6.41: Water Outlet Temperature during Run #49 (Syncrude coke)**



**Figure 6.42: Solids Loading during Run #49 (Syncrude coke)**

decreased, the solids loading at the base of the riser increased while that at the top decreased. Because of the strong dependency of the suspension-to-tube heat transfer coefficient on the solids loading around the tubes, when the gas velocity is decreased, the overall riser solids loading must be increased to maintain a constant loading over the heat exchange surface (Figure 6.42).

**Step change water outlet temperature from 46 to 52 °C (308 to 353 min):**

This increase in water setpoint required an increase in heat removal by the exchanger from 22 kW to 25.1 kW (an increase of 15%). The performance of the controller can be seen in Figure 6.41. Figure 6.42 shows the expected increase in solids loading over the heat exchanger in order to achieve this increase in heat removal.

The increased solids loading was provided by increasing the solids circulation rate. This higher circulation rate resulted in carbon being returned to the riser at a greater rate which increased the combustor temperature (Figure 6.37) and decreased the flue gas oxygen concentration (Figure 6.39). The oxygen controller increased the riser gas velocity and quickly reattained the oxygen setpoint. The GPC controller initially lowered the fuel feedrate (Figure 6.38) and, as the temperature setpoint was re-established, increased the fuel feedrate to maintain the appropriate carbon bed inventory.

**Step decrease in water outlet temperature from 52 to 47 °C (353 to 458 min):**

This decrease in heat removal provided a near mirror image of the previous step change. The request for a decrease in heat removal resulted in a lower solids circulation rate in order to lower the riser solids loading. This resulted in a decrease of the carbon flow rate to the bed, causing an increase in flue gas oxygen and a decrease in combustor temperature. The gas velocity dropped in order to lower oxygen, while the GPC controller caused the fuel feed valve to open more quickly in order to supply more carbon to increase the combustion temperature.

At approximately 400 min, some disturbance affected the system. From the appearance of the response at this time (a loss in temperature and an increase in flue gas oxygen), it appears likely that this disturbance was another interruption in the fuel feed. Recovery of conditions consumed the remainder of this time interval.

**Step change combustor temperature from 870 to 850 °C (458 to 546 min):**

The first 30 minutes of this step appears to have shown acceptable behaviour. The temperature dropped in a time interval comparable to that after the initial step increase at 150 min and approached the new setpoint in the expected manner. The gas velocity decreased as less combustion took place in order to maintain the oxygen setpoint. The overall riser solids loading increased in order to maintain the loading in the vicinity of the heat exchanger at the lower gas velocity.

At approximately 500 min, another equipment failure occurred. As a result of the lower gas velocity the system pressure also decreased. The pressure dropped to a level where there was no longer adequate flow to the oxygen analyser. This resulted in air being drawn into the analyser through the sample vent. In response to the apparent increase in oxygen, the riser gas velocity was reduced resulting in the temporary loss of combustion temperature. When the sample flow was restored, the controllers recovered to the appropriate setpoints.

**Step change in oxygen setpoint from 4 to 3%, 3 to 4%, 4 to 3% (546 to 596 min):**

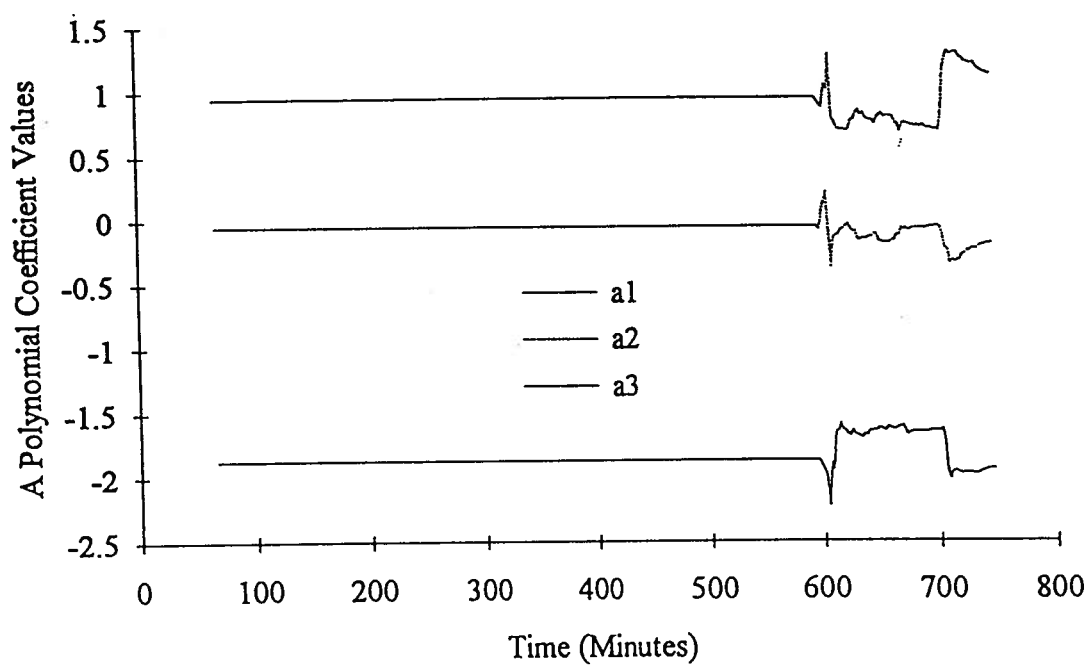
These steps in oxygen setpoint had no noticeable affect on combustor temperature or water outlet temperature. However, as shown in Figure 6.39, response times of the order of 5 to 10 minutes were achieved by this controller. Figure 6.40 indicates that there was a long term, upward drift of oxygen concentration following these step changes, but it can be estimated from this figure that changes in the air supply of about 15 kg/h were sufficient to bring about the required changes between 3 and 4% outlet oxygen concentration. There appeared to be no overshooting by this controller. This is a consequence of detuning and is

advantageous in this situation since it avoids the possibility of cycling throughout the system due to swings in the riser gas velocity. Because the control loops are treated individually, and they are interactive, an overshoot and decay in one loop might otherwise carry over into the others, perpetuating the cycling and leading to instability.

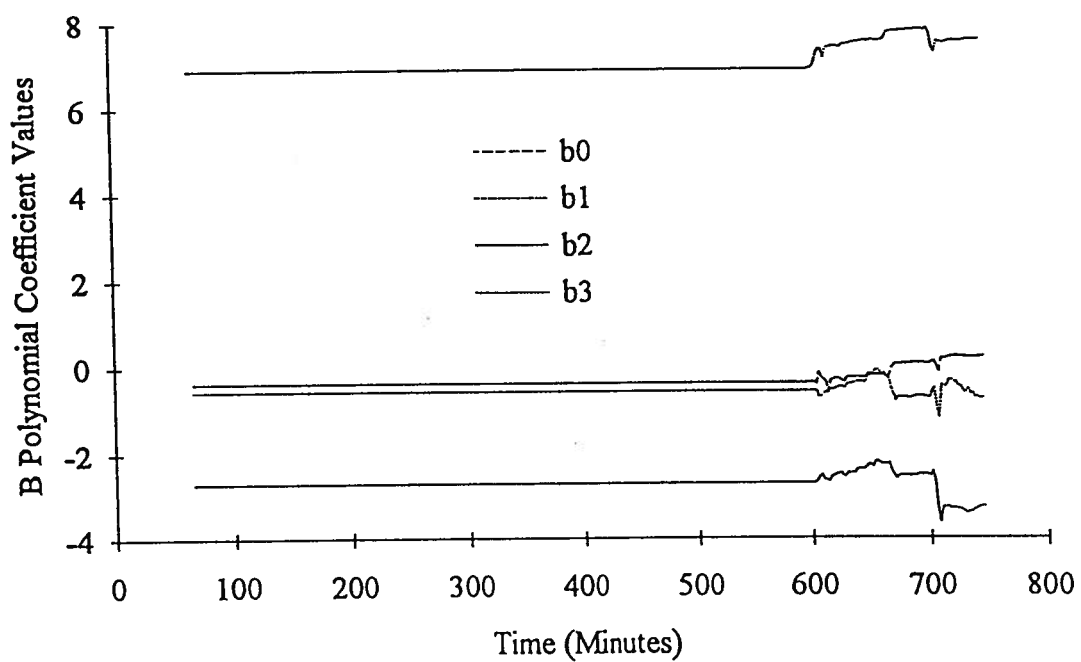
**Adaptation (596 min to end of test):**

The remainder of this test involved two step changes in temperature with the adaptive portion of the controller on-line. The purpose was to ascertain whether or not the model determined from the previous run was adequate. The adaptation of the variables is shown in Figure 6.43 and Figure 6.44. There is no significant change in the parameters from those employed by the GPC controller throughout the first portion of this test. The initial spikes at 596 minutes seen in the  $A(q^{-1})$  polynomial in Figure 6.43 result from initialization of the adaptor. After this initialization period, the coefficients were relatively unchanged. The  $B(q^{-1})$  polynomial shows a very slight upward trend, indicating a slightly lower process gain than that employed by the model.

As occurred at the 500 min mark while responding to a decrease in the temperature setpoint, there was again a loss in flue gas sample at 700 min resulting in air being drawn into the oxygen analyser. The effect of this failure on the adapter is clearly seen in Figures 6.43 and 6.44. The  $A(q^{-1})$  polynomial, which is a predictor of the inertia of the process, is most strongly affected. Fortunately in this case, the change in the model was not so great as to cause a failure of the control system. From the trend of the coefficients to the end of the test, it appears that the model, prior to the analyser failure, would have been re-established.



**Figure 6.43: A Polynomial Coefficients during Run #49 (Syncrude coke)**  
open loop poles @ 1, 0.6458, 0.2942



**Figure 6.44: B polynomial Coefficients during Run #49 (Syncrude coke)**

## **CHAPTER 7: RESULTS AND DISCUSSION - SIMULATOR**

### **7.1 Introduction**

This chapter presents results obtained for the portion of this research focussing on dynamic modeling of CFB combustion. These results are in reference to the model described in Chapter 4. The model is validated by applying PRBS identification methods to the simulator. The resulting model transfer functions are compared to those obtained experimentally on the UBC pilot CFBC. The results indicate that the simulator provides an adequate representation of the dynamic properties of the pilot CFBC. Further validation is achieved by comparison with experimental data taken from the UBC pilot CFBC. These data were obtained from dynamic stimulation by a series of step changes in the fuel feedrate to move the combustor through its operating regime. A brief investigation of various model parameters is presented. This focusses on the effects of char diameter, heat loss, and combustor size. Finally, a controls test simulation is carried out using the same structure employed on the pilot plant as in section 6.4.



## **7.2 Model Validation**

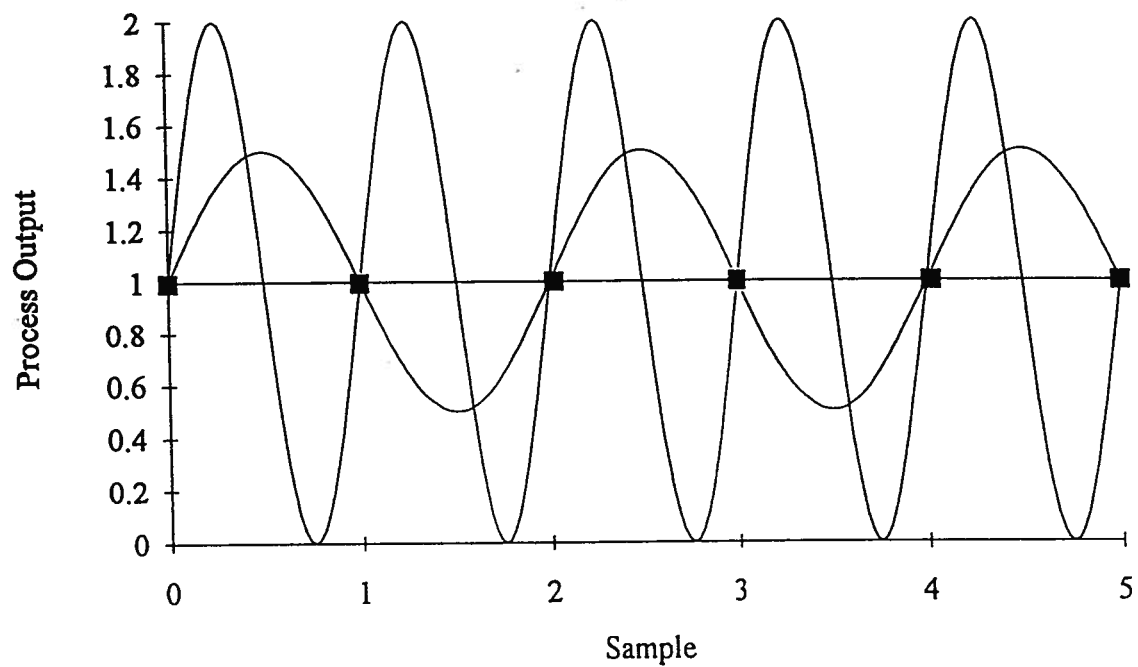
This section provides validation of the model described in Chapter 4. This validation is achieved by comparing transfer function models obtained through simulated PRBS testing on the simulator with those obtained experimentally as described in Section 6.4.2. Further verification is provided by comparing measured and simulated step responses. Two fuels - Highvale coal and Conoco coke - were investigated for this comparison.

### **7.2.1 Transfer Function Comparison**

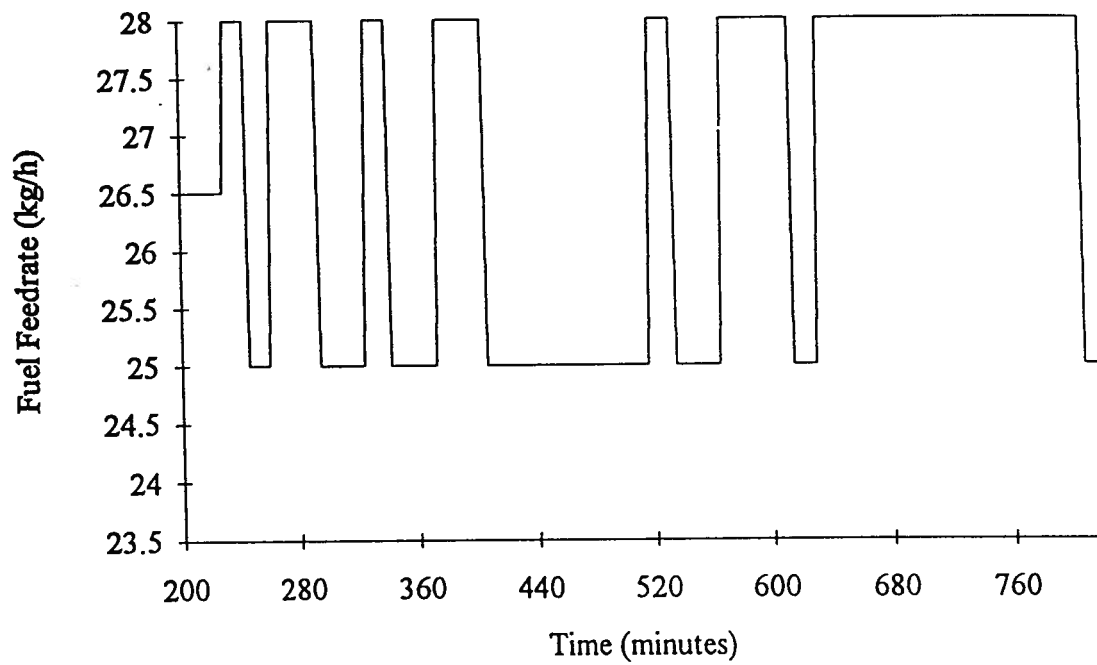
PRBS testing, similar to that performed for model identification in the experimental portion of this research, was carried out on the simulator. The purpose was to generate discrete time process models as a means of comparing the dynamics of the simulator to the UBC pilot CFBC.

Discrete time models only provide information on the process at the instants of sampling. However, as illustrated in Figure 7.1, they do not provide any knowledge of the process between samples. There are an infinite number of models that could fit the data shown in this figure. Further, from the data presented, it is impossible to determine which of the possible continuous time process outputs is the true plant output. By comparing discrete models, high frequency fluctuations are ignored and the comparison is performed on a time scale coinciding with the sample intervals.

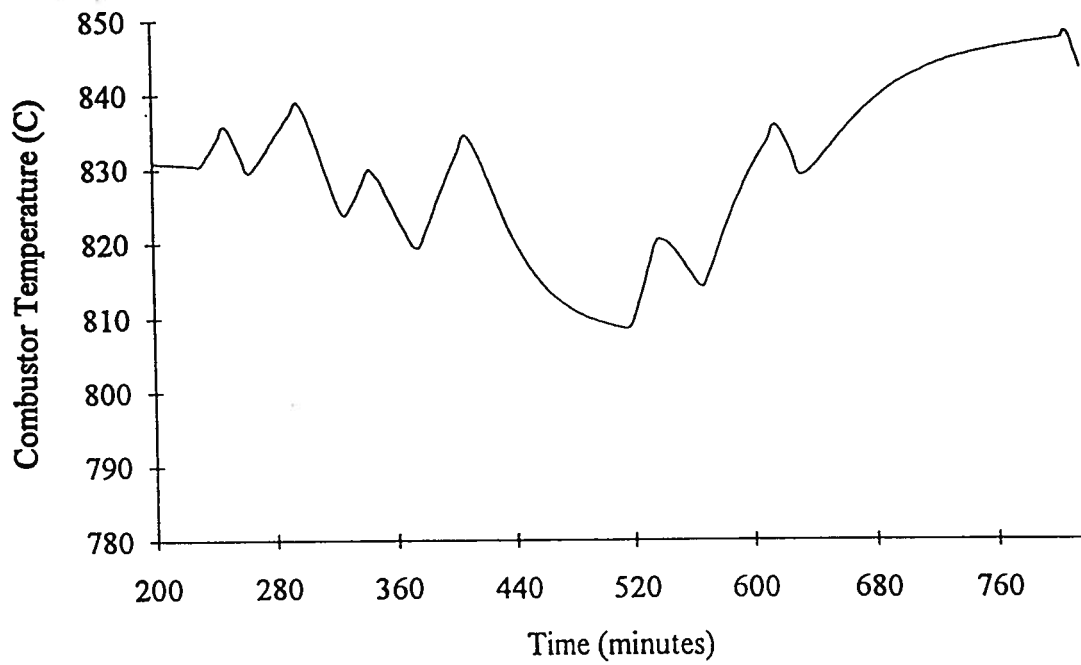
Figure 7.2 presents the PRBS input sequence to the simulator for the generation of data for discrete time modelling of the UBC pilot CFB combustor operating with Highvale coal. Figure 7.3 presents the predicted temperature response during this test. Analysis was performed using the time series methods discussed previously. A comparison of the step response from the time series model determined from the simulation with that determined experimentally from Run #40 is presented in Figure 7.4. Similarly, a comparison of the discrete time models obtained from experimentation and simulation for Conoco coke is presented in Figure 7.5.



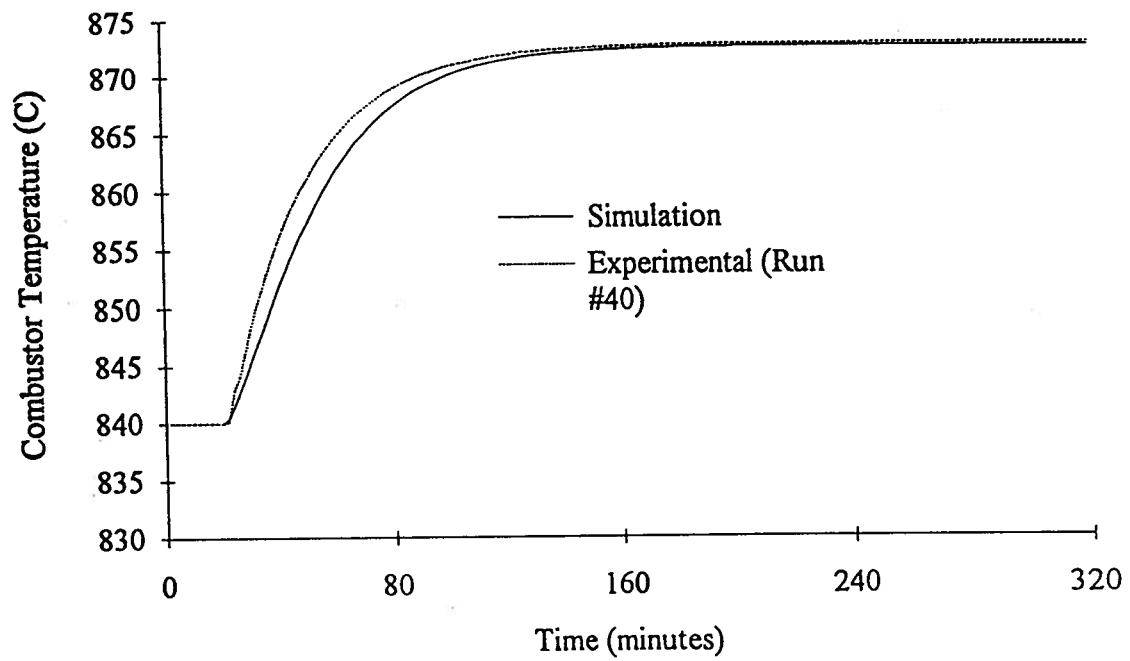
**Figure 7.1: Possible Models for Process Sampled Discretely**



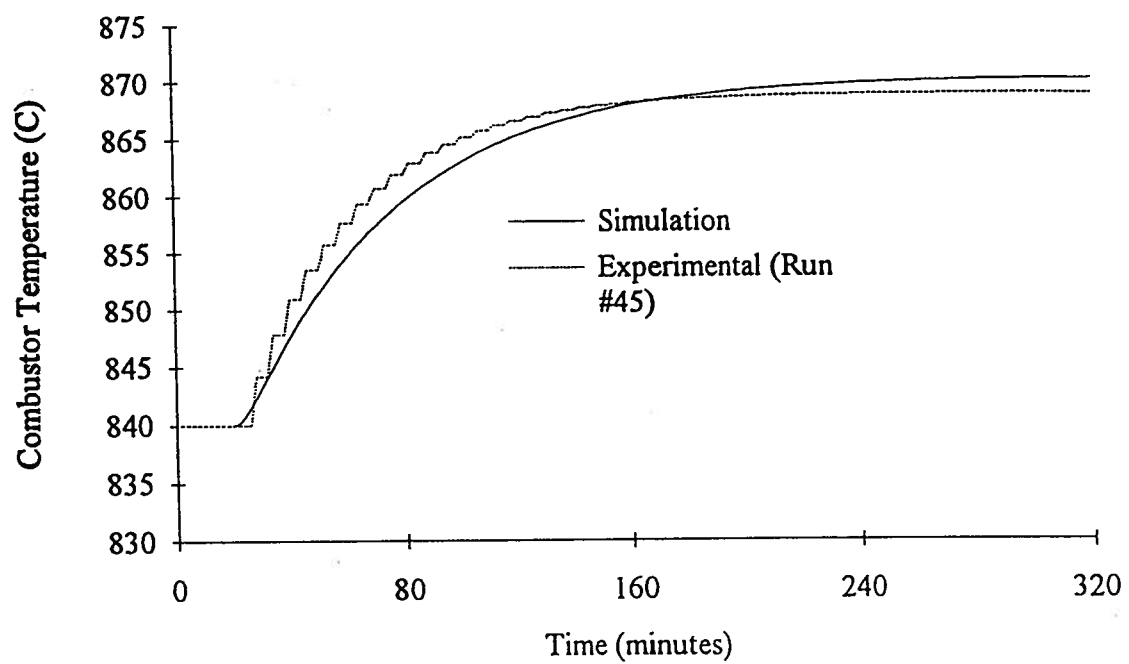
**Figure 7.2: Fuel Feedrate for Simulated PRBS Test with Highvale Coal**



**Figure 7.3: Temperature Response of PRBS Test Simulation with Highvale Coal**



**Figure 7.4: Comparison of Step Responses of Discrete Time Models for Highvale Coal**



**Figure 7.5: Comparison of Step Responses of Discrete Time Models for Conoco Coke**

In Figures 7.4 and 7.5, the predicted process gain of the simulator response has an error of  $\pm 5^{\circ}\text{C}$ . This is a result of the comparison of the transfer function models developed experimentally to those determined for the simulator. The input for the experimental transfer functions is the mA signal to the rotary valve controller while that for the simulator transfer functions is the fuel feedrate in kg/hr. In order to compare these models it is necessary to estimate the fuel feedrate to provide an identical step input to the simulator transfer function as that to the experimental transfer function. In these two figures, the fuel feedrate has been adjusted to match the gains of the transfer functions. The determined fuel feedrates are well within "expected" experimental error in the estimation of the true fuel feedrate based on a mA signal to the rotary valve controller.

Figures 7.4 and 7.5 both show favourable comparison of the simulator with experimental identification. These results indicate that the simulator provides an adequate representation of the dynamic properties of the pilot CFBC unit. Since these dynamics are suitably predicted, the simulator should provide an acceptable medium for the investigation of the behaviour of various schemes for the control of a CFBC.

### **7.2.2 Step Testing Comparison**

The previous section provided a very simple and direct validation of the model and is, itself, adequate to conclude that the model gives reasonable predictions of the dynamics of interest in this work. However, in order to obtain further validation of the proposed dynamic model, additional experimental testing was conducted on the UBC pilot CFB combustor. This testing involved forcing the combustor with a number of step changes in the fuel feedrate and comparing its response with similar forcing of the simulator.

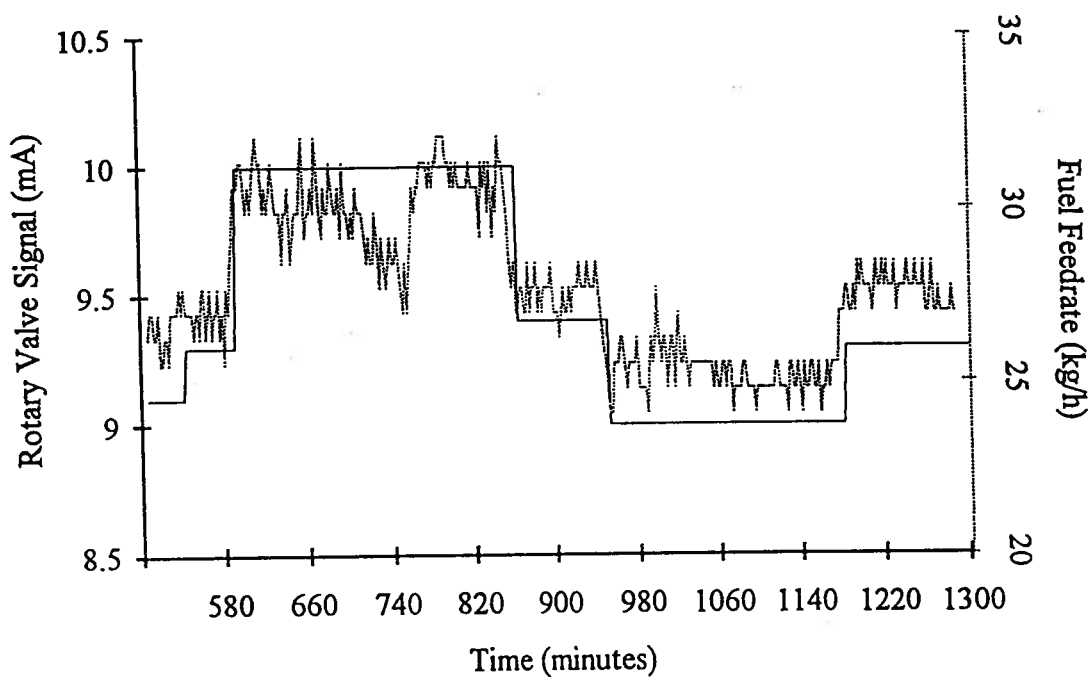
As pointed out earlier, there are a number of inconsistencies in the fuel feedrate due to leakage around the rotary valve feeder and variations in fuel packing within the feed hopper. Because of these inconsistencies, the measured responses from the pilot plant do not necessarily indicate step responses.

The tests on the pilot plant were performed by step changing the mA signal to the rotary valve. This does not always ensure an equivalent change in the fuel feedrate to the combustor. This is illustrated in Figure 7.6 in which the mA signal to the rotary valve is compared with the fuel feedrate measured by loss-in-weight from the fuel hopper, averaged over 10 minute periods. It is evident from this figure that there are discrepancies between these two measures. For example, there was a decrease in flow at 700 min while the mA signal was maintained constant, while there was a higher feedrate of fuel at the end of the test than at the beginning, even though the mA settings were the same.

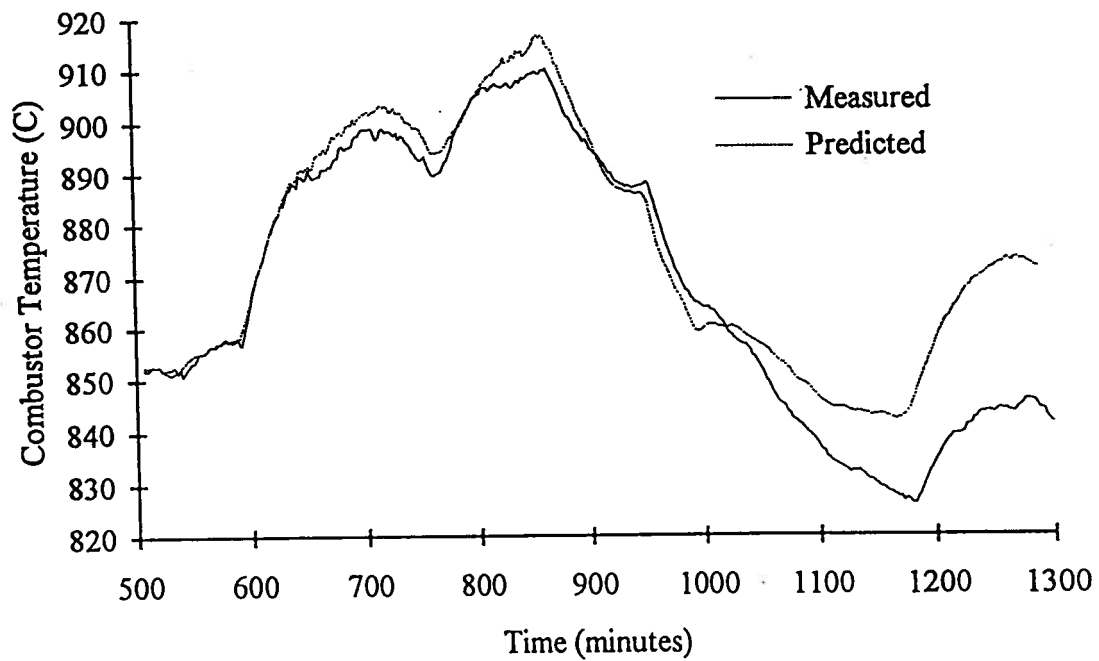
Figure 7.7 compares the measured response of the pilot CFB combustor to that of the simulator when subjected to the measured 10-minute-averaged fuel feedrate. There is very good agreement between the measured and predicted responses for the first 550 minutes. Only a slight over-prediction is apparent in the two temperature peaks at times of 700 and 850 min. The difference between the measured and predicted temperatures at these points is less than 5 °C, well within acceptable limits. The divergence during the final 250 minutes of this test could be due to several factors. Inconsistent operation of the experimental unit may lead to changes which could not be monitored. The difficulties of maintaining consistent conditions throughout an extended test period have already been discussed. The time variant characteristics of the combustor may lead to the discrepancies noted at the end of this run. The model itself may be another contributing factor. Since the current prediction is based upon all previous predictions, any error within the prediction is accumulative, becoming more apparent over time. In any event, excluding the final 300 minutes of this test, the mean and standard deviation of the prediction error of combustor temperature is -1.4 °C and 3.1 °C respectively, while those for the entire test period are -6.3 °C and 9.0 °C.

Figure 7.8 compares measured and predicted flue gas oxygen concentrations. The measured value goes off scale between 720 and 870 min when there was a blockage in the flue gas sampling line which resulted in incorrect oxygen measurements, which are therefore ignored for comparison purposes. In addition, measurements from a time of 600 min up to

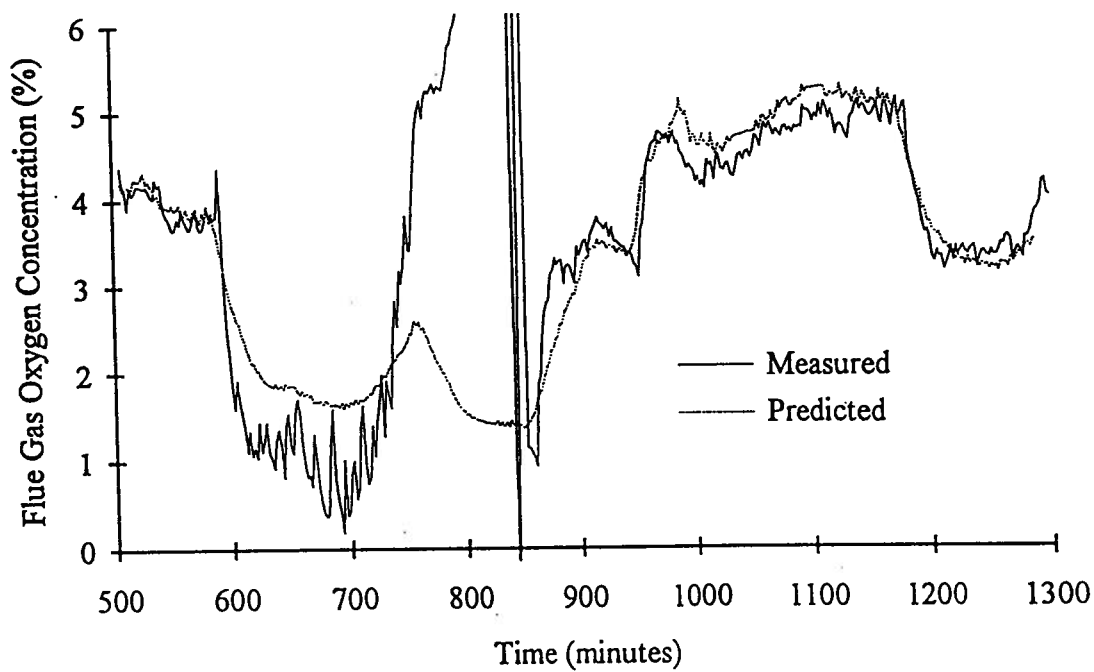




**Figure 7.6: Comparison of Rotary Valve Signal with 10-Minute-Averaged Fuel Feedrate (Highvale coal, Run #52)**



**Figure 7.7: Transient Comparison of Temperature for Highvale Coal (Run #52)**

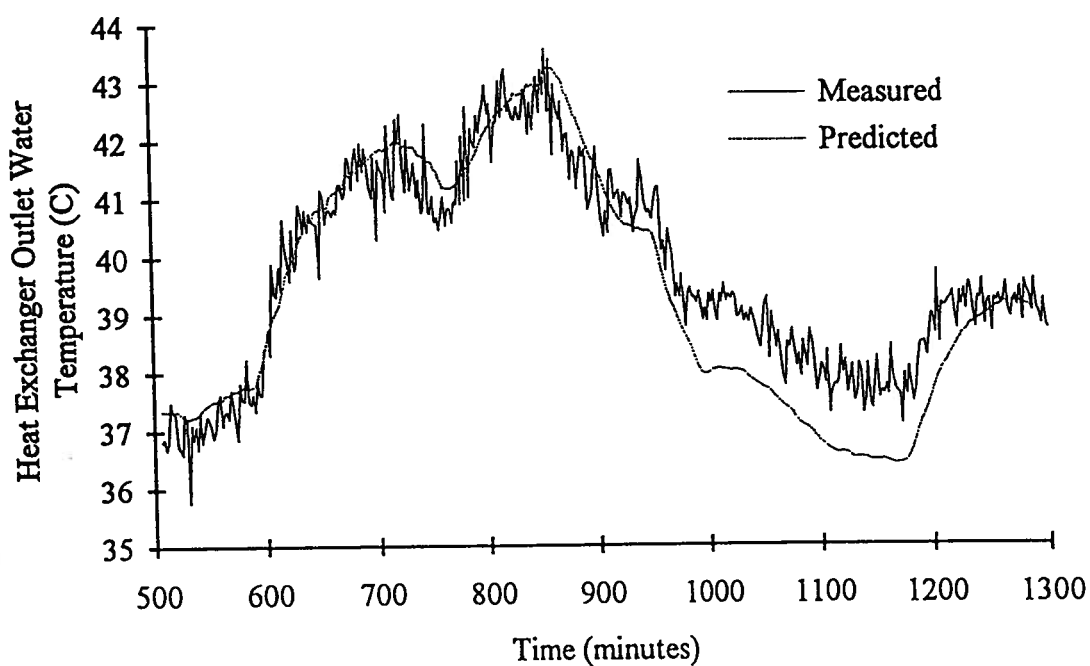


**Figure 7.8: Transient Comparison of Flue Gas Oxygen Concentration for Highvale Coal (Run #52)**

the blockage time are highly suspicious. The wide, rapid fluctuations observed during this time are not common and are more likely caused by development of the problems leading to the blockage; they are, therefore, also ignored. With these exclusions, there is excellent agreement between the measured and predicted oxygen concentration with a mean prediction error of  $-0.1\% \text{O}_2$  and a standard deviation of  $0.3\% \text{O}_2$ . During periods of large changes in concentration, the predicted concentration response was somewhat slower than the measured transient. This results from averaging the measured fuel feedrate over ten minute periods. When there were large changes in the fuel feedrate (at times of forcing step functions), the change forced on the simulator is filtered by averaging, causing the slower predicted response.

Figure 7.9 compares the measured and predicted in-bed heat exchanger water outlet temperature. The first part of this test shows favourable comparison with only slight over-prediction by the model in the early portion of this test. Again, excluding the final 300 minutes of this test gives rise to a mean prediction error of  $-0.1\text{ }^\circ\text{C}$  and a standard deviation of  $0.6\text{ }^\circ\text{C}$ . In contrast, the latter portion indicates approximately  $1\text{ }^\circ\text{C}$  under-prediction by the model for an overall prediction with a mean of  $0.3\text{ }^\circ\text{C}$  and a standard deviation of  $0.7\text{ }^\circ\text{C}$ . The errors discussed in reference to Figure 7.7 are also valid here. The change from a condition of slight over-prediction to  $1\text{ }^\circ\text{C}$  under-prediction occurred quickly and, therefore, is not indicative of an additive error elsewhere in the model (combustor temperature calculation). It is most likely attributable to changing operating conditions during the experiment. For example the nature of the bed material may have changed causing the solids loading over the heat exchange surface to vary, thereby creating a discrepancy between the calculated and predicted heat removal rates. Due to the fast response of these variables, any discrepancy is noted almost immediately.

Comparing Figure 7.7 and Figure 7.9, the divergence in temperature begins to take place at the same time as the large change in the error of heat removal prediction occurs. As well, the error in combustor temperature leads to a progressively higher predicted temperature consistent with the under-prediction of water temperature. The model under-predicted the

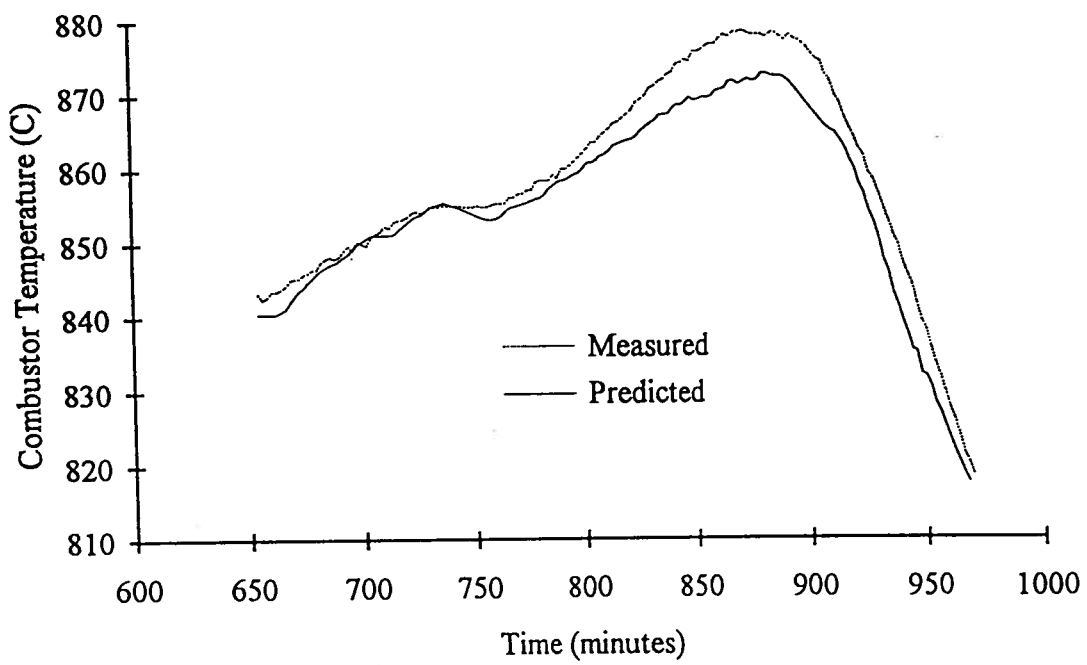


**Figure 7.9: Transient Comparison of Heat Exchanger Water Outlet Temperature for Highvale Coal (Run #52)**

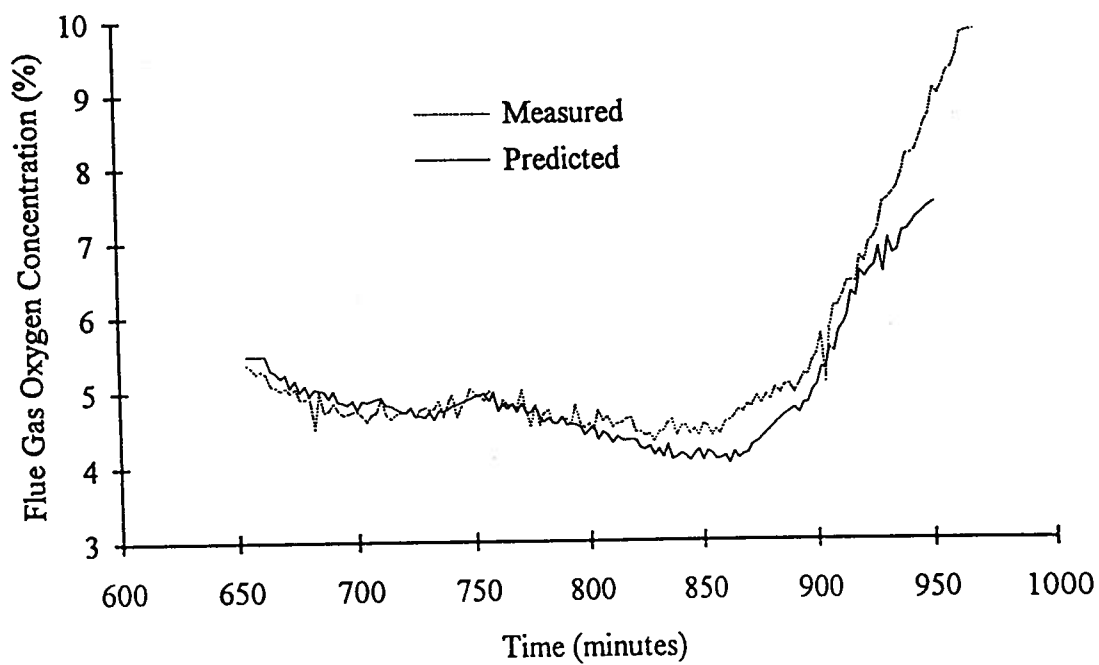
amount of heat being removed from the system during the final portion of the test. This heat builds up in the refractory and bed material (of the model) resulting in over-prediction of the combustor temperature. This suggests that, of the two possible sources of errors discussed, the discrepancy most likely results from changing bed material characteristics during the experimental test rather than any inherent additive error in the dynamic model.

As stated in Section 6.2.2, the changing bed characteristics are a result of the fuel being fed to the combustor. In this instance, the fuel was Highvale Coal. For this fuel, even after only short periods of testing, large stones ( $\approx 5$  mm in diameter compared to a sand size of  $259\text{ }\mu\text{m}$ ) accumulate in the bed. The source of these stones is shale in the fuel. After shut-down of the pilot unit, they are found mixed in with the circulating inventory as well as accumulated upon the primary air distributor at the base of the riser.

A similar comparison was made for Conoco coke, the results of which are presented in Figures 7.10, 7.11, and 7.12. In this case the water temperature prediction is excellent throughout (a mean error of  $0.0\text{ }^{\circ}\text{C}$  and a standard deviation of  $0.4\text{ }^{\circ}\text{C}$ ), indicating that changing bed material is not a source of error for this test. The temperature response shows a divergence as the maximum is approached at 860 min, while the predicted and measured values reconverge after this point is passed (a mean error of  $3.5\text{ }^{\circ}\text{C}$  and a standard deviation of  $2.5\text{ }^{\circ}\text{C}$ ). The oxygen response shows this same phenomenon (a mean error of  $0.7\text{ }\%\text{ O}_2$  and a standard deviation of  $2.1\text{ }\%\text{ O}_2$ ). There are two probable sources of this divergence/reconvergence error. It might be due to some anomaly or disturbance during the test, causing some unpredictable, temporary behaviour, giving rise to the initial divergence, with subsequent reconvergence. Alternatively, there may be some error in the kinetic rate coefficients determined in Appendix A, leading to an incorrect prediction of the temperature effects on the rate of combustion. This would account for the reconvergence as the temperatures which showed good comparison at the start of the test are again approached towards the end of the test. However, the temperature response indicates that, if this were the case, the prediction of the rate of combustion is too low at the higher temperatures, since the

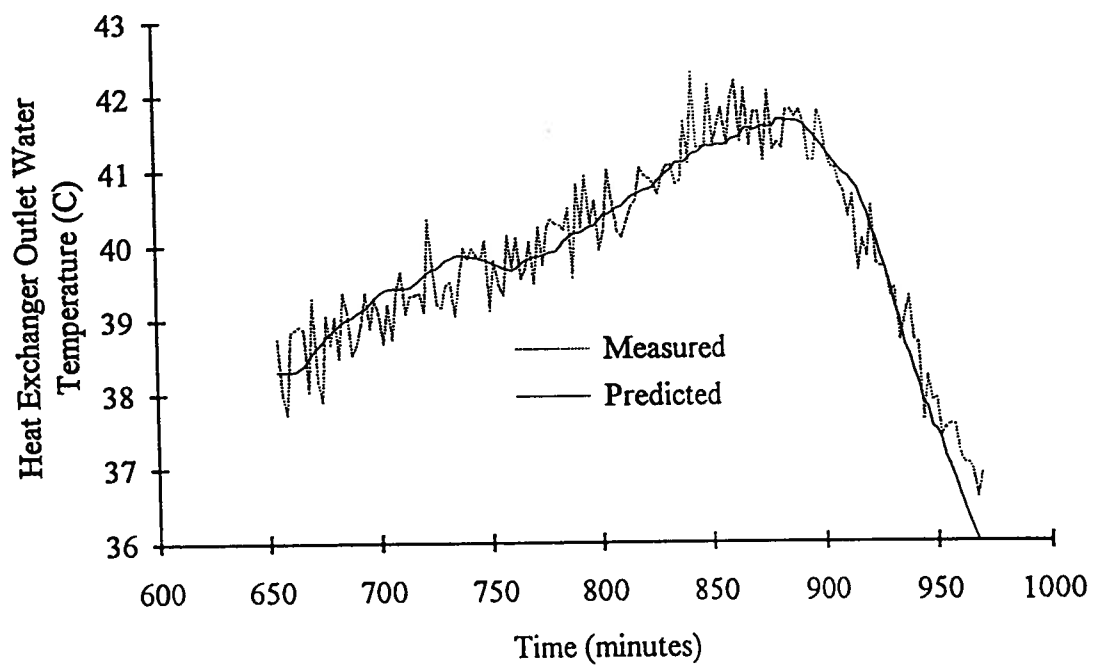


**Figure 7.10: Step Test Comparison of Temperature for Conoco Coke (Run #51)**



**Figure 7.11: Step Test Comparison of Flue Gas Oxygen Concentration for Conoco Coke (Run #51)**





**Figure 7.12: Step Test Comparison of Heat Exchanger Outlet Water Temperature for Conoco Coke (Run #51)**

predicted temperature is lower than measured. Given that the predicted rate of combustion is lower than the true value, one would expect that the predicted oxygen concentration would be higher than measured, since the predicted consumption is also lower. However, Figure 7.11 shows that this is not the case. During this period, the predicted oxygen concentration is also lower than measured, suggesting that an unmeasured disturbance is more likely the source of error in this test.

### **7.3 Model Characterization**

This section provides a brief evaluation of the model. It considers the effects of char particle size and heat transfer surface area, and simulates a control test as in Section 6.4. This is a discussion of the behaviour of the model predictions only, without direct comparison with experimental measurement.

#### **7.3.1 Effect of Char Size**

In fitting the data in the previous section, it was found that the particle size of the burning fuel had only a limited influence on the model predictions. Although the size chosen in the simulation gave a "best" fit, there is a range of diameters (600  $\mu\text{m}$  to 1 mm) over which the model generates acceptable predictions.

There are two factors which control the transient temperature response to a change in fuel feedrate. These are the thermal capacity, or inertia, of the system and the carbon inventory within the bed. The thermal capacity refers to the physical unit. Heat is stored by the materials of construction (refractory and other structural materials, included here with refractory) and the bed solids. At steady-state, the amount of heat entering these materials is equal to that leaving. Similarly, the heat lost by the solids passing through the standpipe is equal to that gained while in the combustion zone.

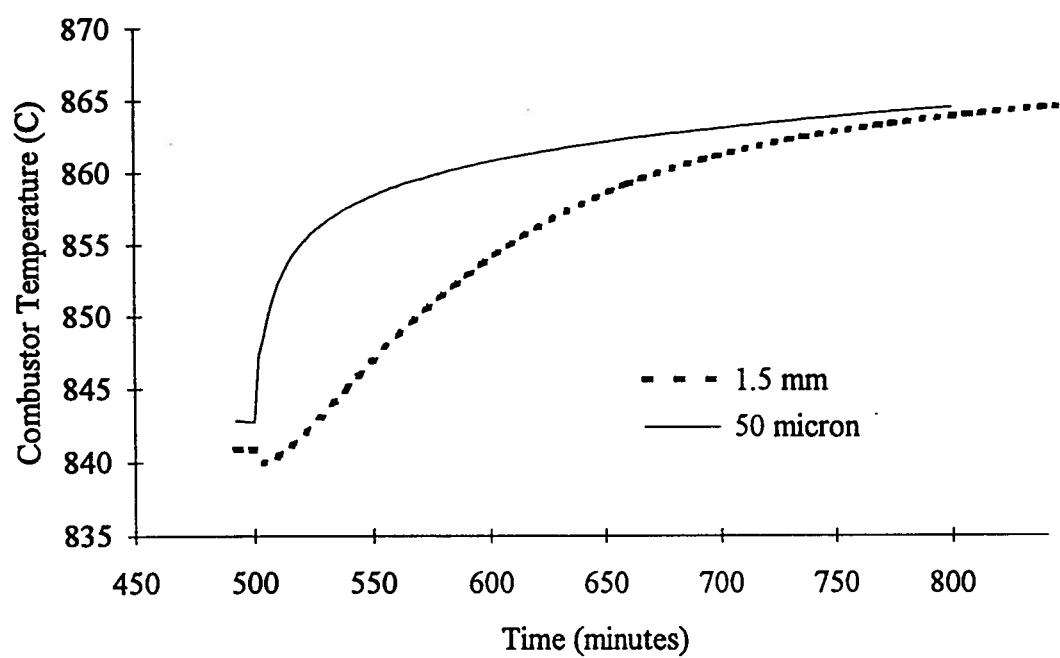
As an illustration, consider an attempt to raise the temperature of the combustor by increasing the fuel feedrate. As the amount of fuel in the combustion zone is increased, there

is a corresponding rise in the amount of heat generated due to combustion. The excess heat is absorbed by the system until a balance in heat loss and heat absorption is re-established. The time required to move from one steady-state to another is that required for the excess heat to be collected in the structure and circulating solids.

The carbon inventory also influences the transient temperature. At steady-state, there is a balance of carbon within the system, with carbon entering with the fuel feed and being returned via the L-valve being matched by that consumed by combustion. If there is an increase in fuel feedrate, there will be an increase in carbon consumed in the combustion zone due to greater carbon availability there. However, this increased rate of combustion is not as great as the increase in feedrate, and there is, therefore, an increase in carbon being returned to the standpipe from the primary cyclone. Since the total inventory of solids within the system remains essentially unchanged, there must also be an increase in carbon inventory within the riser. Given that the solids circulation rate remains constant, as the carbon inventory increases, the rate of carbon returned to the combustion zone from the L-valve must also rise. Thus, even though a step change in fuel feed results in a step change in carbon introduced with the fuel, a transient in total carbon introduction to the combustion zone also results as carbon builds up within the total inventory of solids.

As already stated, the thermal capacity of the system and the rate of carbon accumulation combine to dictate the combustor response to a change in fuel feedrate. Given the limited effect of particle size, a brief assessment of the effects of the above two controlling parameters was carried out.

In order to evaluate carbon accumulation, extremes in burning particle size were investigated. Figure 7.13 illustrates the temperature response of the model to a step change in fuel feedrate for fuel particles of two widely differing sizes - 50  $\mu\text{m}$  and 1.5 mm. As expected, the response for the smaller particle size is significantly faster than that for the larger fuel particles. This is a direct result of surface availability. The surface-area-to-mass ratio of the 50  $\mu\text{m}$  particle is approximately 67  $\text{m}^2/\text{kg}$ , while that of the 1.5 mm particles is only 2.2  $\text{m}^2/\text{kg}$ .

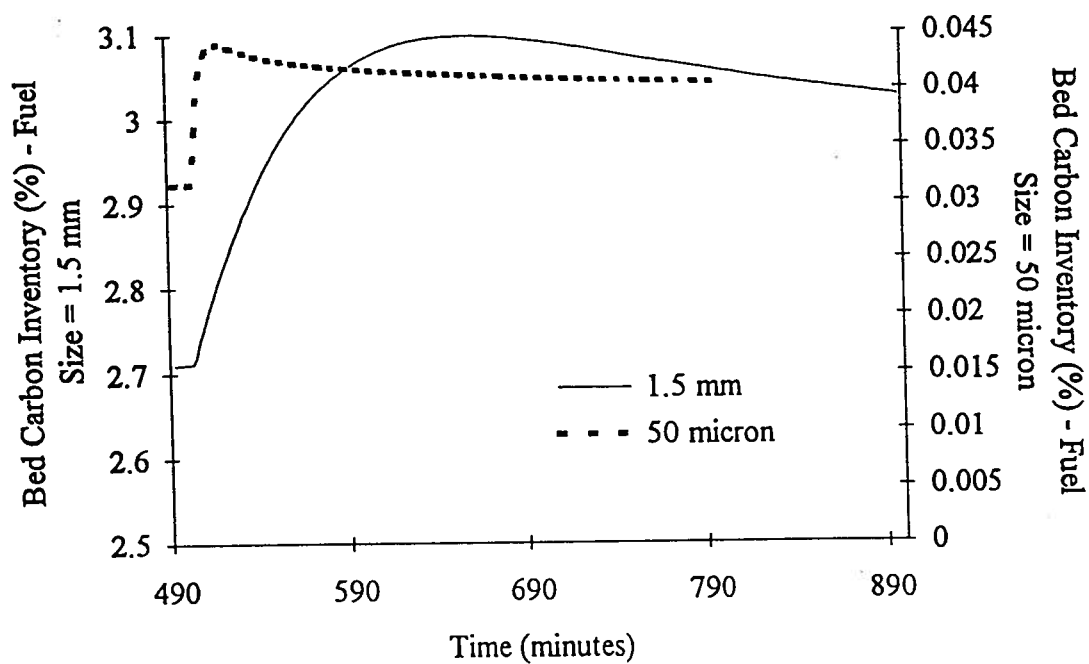


**Figure 7.13: Predicted Effect of Particle Size on Temperature Response (Conoco coke, solids loading = 120 kg/m<sup>3</sup>, total air feedrate = 205 kg/h, P/S ratio = 2)**

The much greater surface available for combustion of the smaller particles is reflected in a faster rise in temperature when excess solids are introduced in response to the higher rate of combustion.

Assuming that the 50  $\mu\text{m}$  particles are small enough to burn almost instantaneously upon introduction to the bed, their response (Figure 7.13) can be attributed to the thermal mass of the circulating solids and containing walls. Accepting this and simplifying the analysis, we can attribute the difference in the two responses to the rate of solids accumulation within the system. There should be no accumulation of the smaller particles (given the assumption of immediate consumption) so that the amount of heat released should change instantaneously when the fuel feedrate is changed.

This is a simplification of the situation. Figure 7.14 shows the total predicted carbon inventory during these changes. This inventory is based on the fraction of carbon in the solids being returned to the riser from the base of the standpipe. The assumption of instantaneous combustion for the smaller particles is not quite accurate. The initial steady state indicates a carbon inventory of 0.032 %. This is very small relative to the inventory of 2.7 % for the larger particles - higher by a factor of almost 100. After an increase in fuel feedrate, it requires about 5 minutes for the smaller particles to reach a new inventory of 0.045 %. This lag time indicates that burning is not instantaneous. Nevertheless, for the purposes of discussion, it is short enough to be considered negligible on the time scale considered here. As shown in Figure 7.14, after the new inventory is achieved, there is a downward trend. This is due to the dependency of rate of combustion on both surface availability and temperature. The increase in heat release on moving to a new steady state fuel feed is almost instantaneous for the small particles but, as temperature increases, there is an associated rise in the rate of combustion which tends to deplete the steady state bed inventory. In contrast, the 1.5 mm particles take approximately 150 minutes to reach peak carbon inventory. Although this time lag is a combination of surface area and temperature effects, it provides a rough estimate of the time required to build up the carbon inventory to a comparable surface area availability.



**Figure 7.14: Effect of Fuel Particle Size on Bed Carbon Inventory (Conoco coke, solids loading = 120 kg/m<sup>3</sup>, total air feedrate = 205 kg/h, P/S ratio = 2)**

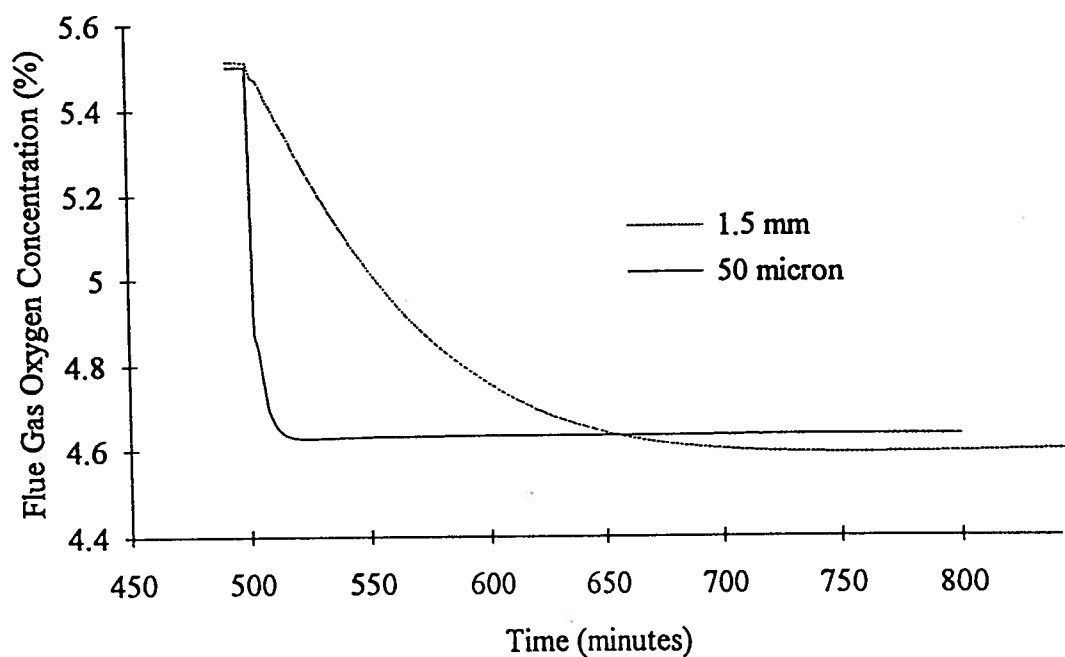
Again, the predicted reduction in inventory after this point results from increasing combustion due to increasing temperature.

This interpretation is further illustrated in Figure 7.15. The oxygen transients show less sensitivity to the temperature. The 50  $\mu\text{m}$  particles are predicted to reach a new steady state in the 5 minute period, while the 1.5 mm particles are predicted to have a response time of almost 200 minutes. The final difference in oxygen concentration is due to temperature effects. The oxygen concentration for the larger particles is drifting upwards and, given sufficient time, would eventually match that for the smaller particles.

The above discussion is a simplification since both temperature and particle size affect the rate of combustion. For any given particle size, when the fuel feedrate is increased, there is an increase in the rate of combustion in response to an increased surface area availability. This is accompanied by an increase in heat release, part of which is adsorbed by the system. The increase in fuel feedrate leads to a buildup of carbon in the bed. Solids being returned to the riser from the recirculation loop provide a secondary source of increased surface availability, intensifying the rate of combustion. Larger particles take longer to accumulate available surface area. After a period of time, a balance is re-established between the surface area availability and the rate of combustion (resulting in the peak in carbon inventory in Figure 7.14). From this point onward, the temperature response is primarily associated with absorption of heat by the thermal mass. As temperature continues to rise, there is a reduction in carbon inventory corresponding to an increasing rate of combustion, maintaining the balance of surface availability to rate of combustion.

### **7.3.2 Effect of Heat Transfer Area**

The response of the combustor employed in this work is dominated by the thermal mass of the refractory. It is of interest to consider how these responses might change in larger systems, for which a much lower percentage of heat generated is absorbed by the structure. A



**Figure 7.15: Effect of Particle Size on Oxygen Concentration - Conoco Coke**  
(Conoco coke, solids loading = 120 kg/m<sup>3</sup>, total air feedrate = 205 kg/h, P/S ratio = 2)



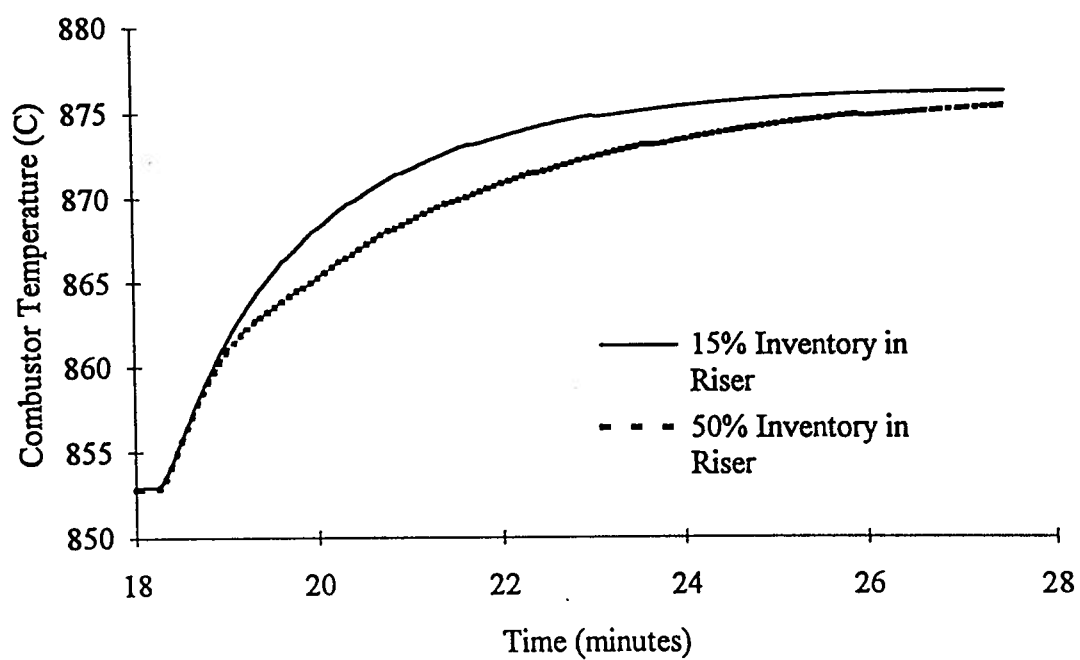
larger unit was simulated here for comparison. Only a crude approximation is used since the objective is to predict a major factor rather than to provide an accurate simulation.

An idealized combustor is envisaged in which there are no equipment heat losses, i.e. there are no heat losses to any riser structural material or through the standpipe. The only heat loss is due to heat absorbed (or released) by solids re-introduced to the combustion zone through the L-valve. This combustor is assumed to be 18 m tall with a cross-sectional area of 4 m<sup>2</sup>, producing about 20 MW<sub>t</sub>.

Figure 7.16 shows the predicted temperature responses of this unit to a step change in the fuel feedrate for two different inventories. These response times are significantly faster than in the UBC pilot CFB (about 4 minutes compared with 30 to 40 minutes). Since the effects of construction materials have been eliminated, this response is strongly related to the rate of carbon build-up in the bed, with only a minor contribution from heat absorbed by the circulating solids. This confirms the controlling effect the mass of the pilot plant refractory has on the temperature response.

The effect of the thermal mass of the circulating solids is also shown in Figure 7.16. For an average riser solids loading of 60 kg/m<sup>3</sup>, there will be a total mass of 4800 kg of solids in the riser at any given moment in time. Two cases are considered, one where the standpipe stores 15% of the total solids inventory for a total of 5650 kg of solids and the other where the standpipe contains 50% of the inventory, or 9600 kg total inventory. The increased inventory is predicted to reduce the temperature response only slightly. Although not indicative of the true response of a larger unit, these results demonstrate that heat absorption by the containing structure plays a major role in controlling the temperature response to changes in fuel feedrate.

It is suspected that there is another, unmodelled, factor which becomes controlling in larger units. This is the steam side of the system. For large industrial units, water and steam are required to remove the heat generated by the combustor. This water and steam contribute



**Figure 7.16: Effect of Bed Inventory on Temperature in Large Scale Unit (Conoco coke, solids loading = 120 kg/m<sup>3</sup>, total air feedrate = 205 kg/h, P/S ratio = 2, char diameter = 900  $\mu$ m)**

to the thermal mass of the process - the counterpart of the pilot plant refractory. The rate at which heat is absorbed by the circulating water/steam must affect the rate of temperature rise.

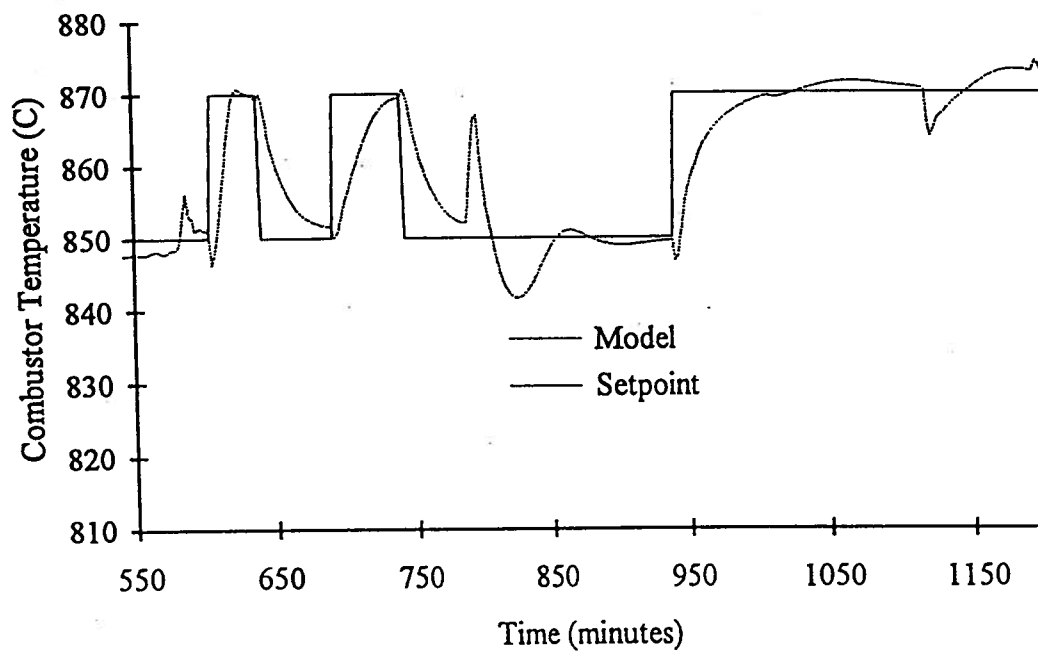
### **7.3.3 Controls Test Simulation**

This section presents the results of a control test carried out on the simulator using the same control structure as in the experimental portion of this work. Specifically, the structure employed in Section 6.4 was implemented in parallel to the model to provide a simulation of that structure in response to setpoint changes in combustor temperature, flue gas oxygen concentration, and in-bed heat exchanger water outlet temperature.

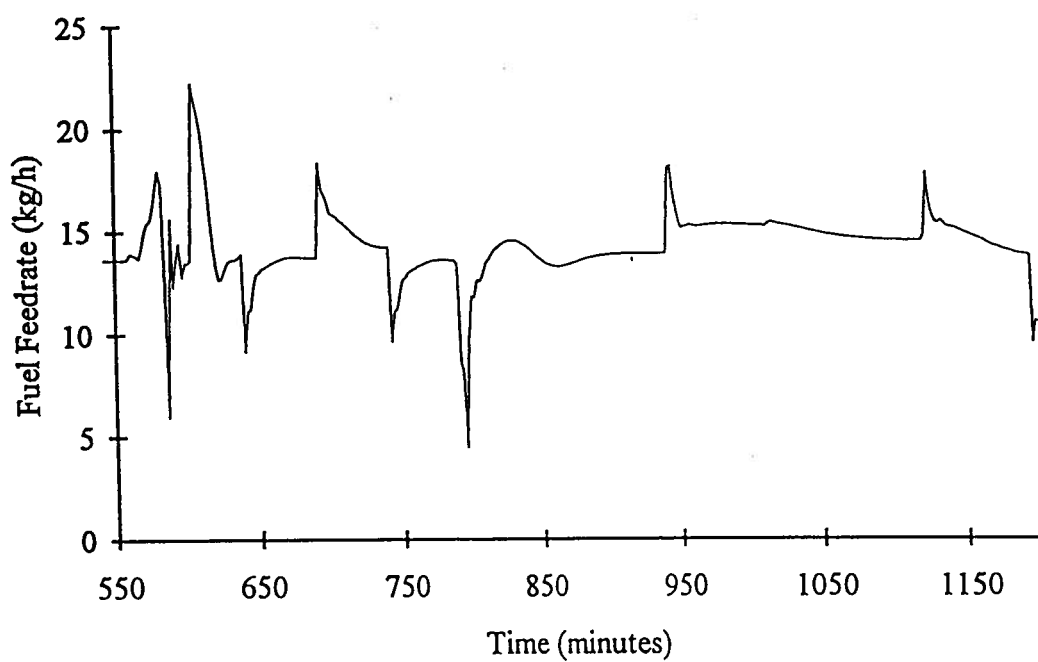
Figure 7.17 compares the combustor temperature with the setpoint for this run. Good GPC control over temperature is indicated. The controller is acting on a 2-minute sampling interval and, with the noise filter, achieves response times of the order of 30 minutes. The apparent loss of control at 790 minutes is due to a disturbance created by a change in the setpoint of the heat exchanger water outlet temperature controller as discussed below.

Figure 7.18 presents the fuel feedrate settings applied by the GPC algorithm to obtain control. After a somewhat erratic initial period of approximately 70 minutes, during which model tuning is being performed by the adaptor, the behaviour of this controller is very good. Each setpoint change leads to an initial large change in fuel feedrate, followed by a gradual approach to a final value. This controller behaviour is common for a first order, or underdamped second order process for which GPC employs a control horizon,  $NU$ , of 1. As in the experimental work for which model orders were somewhat unclear, the model polynomials,  $A(B)$  and  $B(B)$ , were over-parameterized, both being taken as third order. This over-parameterization does not appear to adversely affect the behaviour of the controller. It does allow for changes in fuel type to be employed without operator intervention.

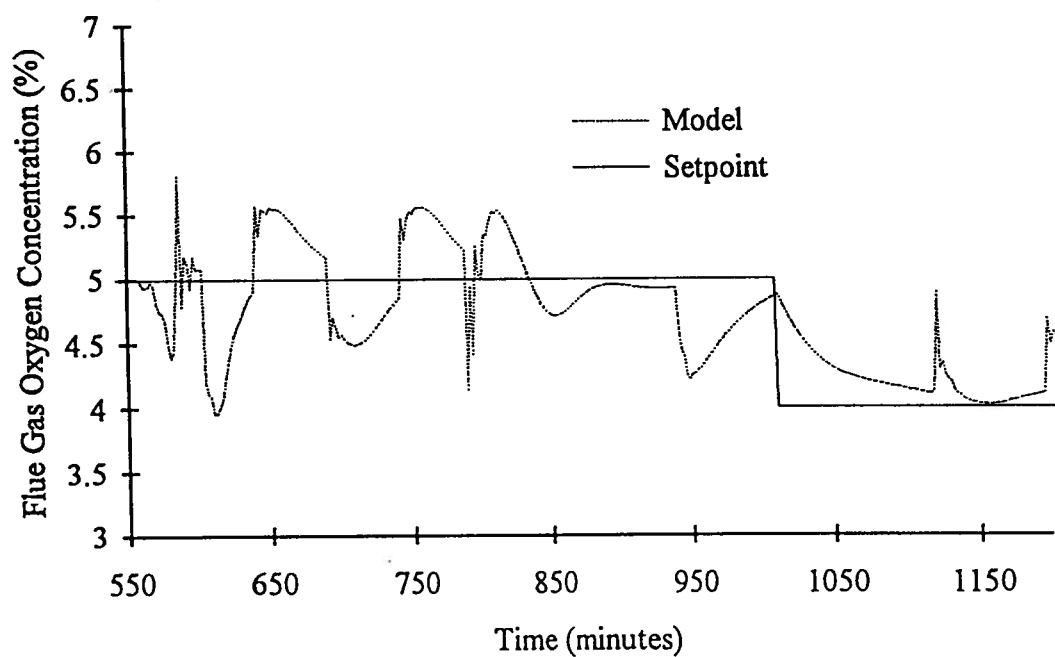
The oxygen control is shown in Figure 7.19. In order to approximate the delays and the mixing of the flue gas sampling encountered on the experimental apparatus, the flue gas oxygen prediction is passed through a first order filter before being supplied to the oxygen



**Figure 7.17: Temperature during Simulated Controls Test (Conoco coke, P/S ratio = 2, char diameter = 900  $\mu\text{m}$ )**



**Figure 7.18: Fuel Feedrate during Simulated Control Test (Conoco coke, P/S ratio = 2, char diameter = 900  $\mu\text{m}$ )**



**Figure 7.19: Oxygen during Simulated Control Test (Conoco coke, P/S ratio = 2, char diameter = 900  $\mu\text{m}$ )**

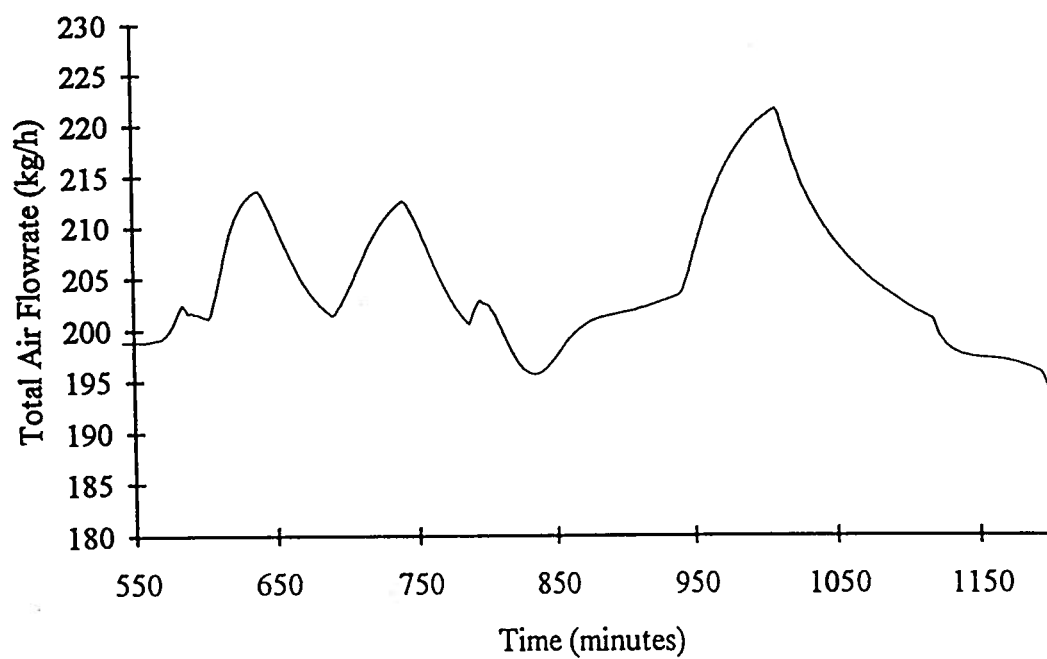
controller. This is only a crude approximation of the true process, but it adequately represents the associated signal distortion which is to be expected in reality. A simple proportional controller is employed to control the oxygen for this simulation. As before, the main purpose of this controller is to maintain the oxygen availability in the presence of the disturbances created by the other controllers. It can be seen from this figure, as shown in the experimental work, that this controller can maintain an oxygen concentration within  $\pm 1\%$  of the setpoint. For completeness, the total air flow to achieve this control is shown in Figure 7.20.

The heat exchanger water outlet temperature is shown in Figure 7.21. In the absence of the high frequency pressure fluctuations encountered in actuality, the controller maintains extremely tight control over this variable.

The loss in control after the setpoint change at 790 minutes is not a controller failure. Rather, it is due to a solids loading constraint which is employed to ensure that the overall riser solids loading is kept within reasonable bounds as discussed in Section 6.4.

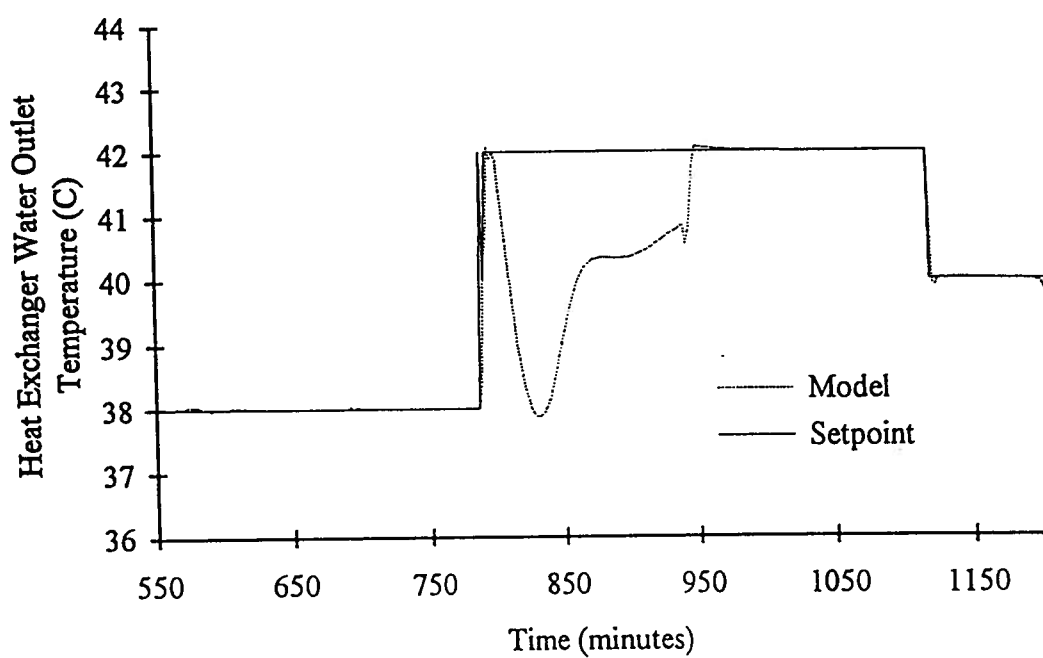
The riser solids loading is shown in Figure 7.22. At the time of this setpoint change, the loading rises to its allowable maximum of  $180 \text{ kg/m}^3$  and remains there for approximately 150 minutes. During this period, the water temperature controller is unable to achieve its setpoint. When reasonable operation is re-established at 950 minutes, the controller quickly regains control of the combustor temperature. The upward drift in loading from 1000 to 1100 minutes reflects the controller compensating for changes in combustion temperature. The water outlet temperature is a function of combustor temperature (radiation) and solids loading (particle convective heat transfer). Thus, as the combustion temperature changes, the solids loading adjusts the particle convective heat transfer coefficient to maintain the desired water temperature.

Figure 7.23 shows the polynomial coefficients for this test. There is very little change in these coefficients after the initial tuning period (until 650 minutes). The only significant change occurs at the 790 minute mark when there was a change in setpoint of water temperature in response to a significant change in the riser solids loading. This indicates that

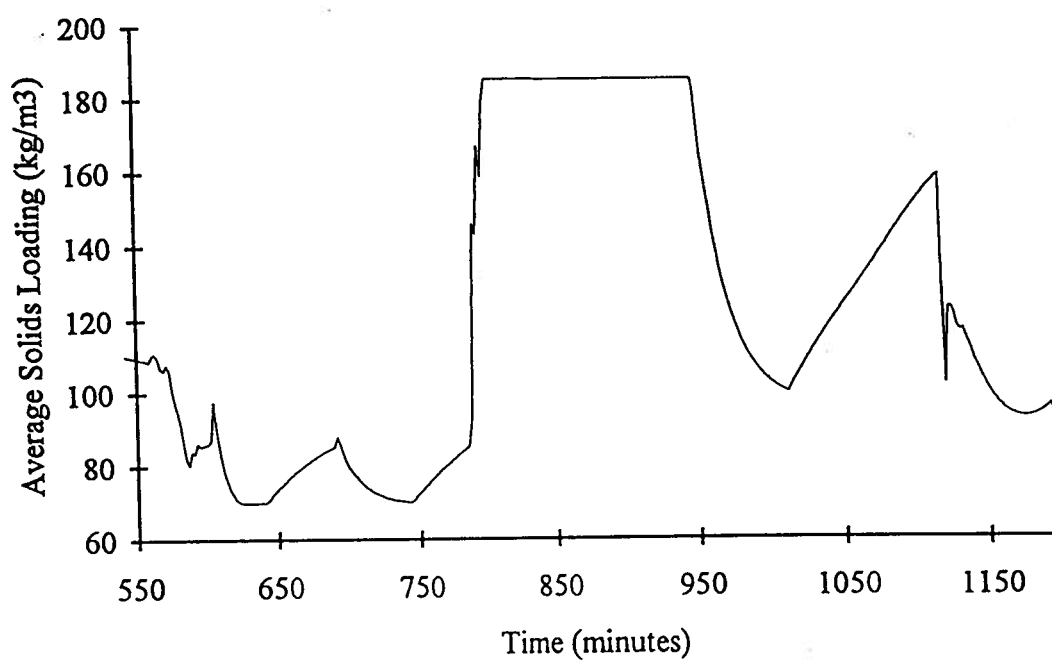


**Figure 7.20: Total Air Flow during Simulated Control Test (Conoco coke, P/S ratio = 2, char diameter = 900  $\mu\text{m}$ )**

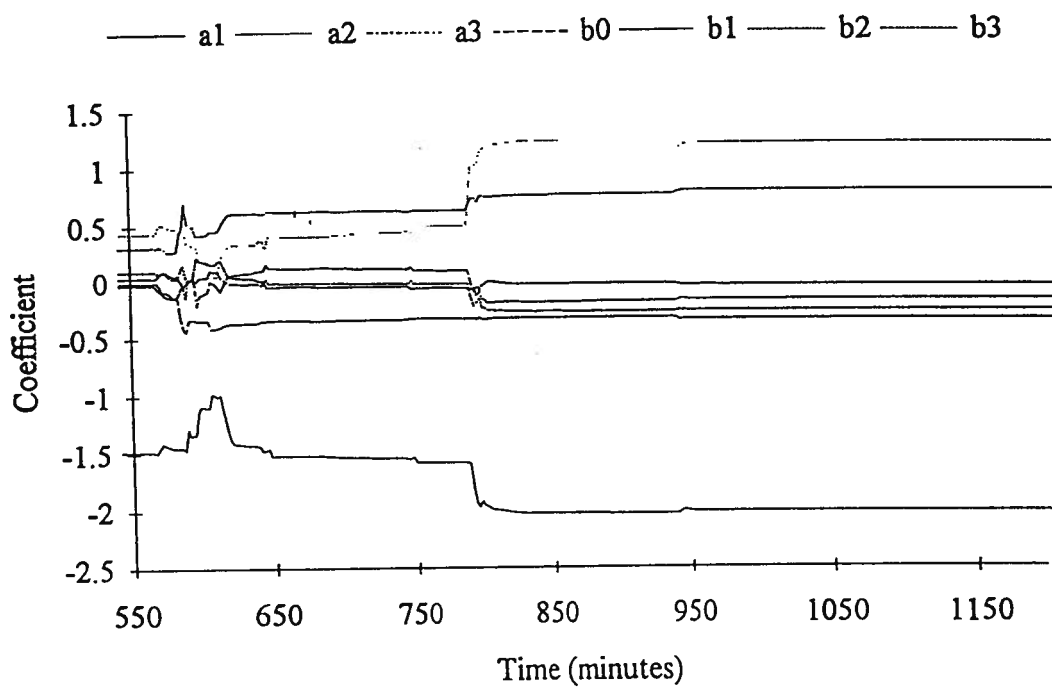




**Figure 7.21: Water Temperature during Control Test Simulation (Conoco coke, P/S ratio = 2, char diameter = 900  $\mu\text{m}$ )**



**Figure 7.22: Average Solids Loading during Simulated Controls Test (Conoco coke, P/S ratio = 2, char diameter = 900  $\mu\text{m}$ )**



**Figure 7.23: Coefficients during Simulated Controls Test (Conoco coke, P/S ratio = 2, char diameter = 900  $\mu\text{m}$ )**

the change in the discrete temperature model is due to the change in loading. Note that at the 950 minute mark, the adaptor was disengaged, and the model maintained constant for the remainder of the test.

The temperature changes and oxygen fluctuations do not appear to affect the model, while the change in riser solids loading is significant. This suggests that, in the regions investigated, the only non-linearity of the process in terms of temperature control is associated with the riser solids loading. Although this non-linearity exists, it likely has only a minor influence, since fixing the model for the last portion of the test does not noticeably affect the quality of the control achieved over combustion temperature.

#### **7.4 Summary**

This chapter has provided sufficient validation for the model described in Chapter 4 in the application of a controls simulator. As well, a brief investigation of various parameters was provided in addition to a demonstration of the application of the simulator in the implementation of a controls structure.

## **CHAPTER 8 - SUMMARY AND CONCLUSIONS**

There is a growing use of circulating fluidized beds as utility boilers. In this application, the combustor spends most of its life under transient conditions, following the demand of the steam-side. This, in addition to stricter emission requirements, has led to the increasing importance of understanding the transient nature of the circulating fluidized bed combustion (CFBC) process, specifically for control system design and development.

The work here responds to this need through investigation of the dynamics of the CFB combustion process. A dynamic model for the prediction of key variables for combustor control has been developed, and advanced control methods have been demonstrated using the UBC pilot CFB combustor.

### **8.1 Summary**

A dynamic model has been developed which predicts the transient behaviour of the combustion temperature, rate of heat removal by the in-bed heat exchanger, and the flue gas oxygen concentration for a circulating fluidized bed combustor. These three factors fully define the combustor at any time. This model was conceived to provide an environment in

which control systems could be investigated. It is therefore macroscopic and does not consider microscale phenomena.

The model was incorporated into a simulator to provide an environment for the reproduction of the combustion process on computer. The simulator predicts the behaviour of the combustor under either manual control, in which operator inputs are accepted from the keyboard, or automatic control, for which any given control scheme may be executed in algorithmic form to meet specified combustion conditions.

In order to accommodate the chemical kinetic behaviour of various fuels, separate tests were carried out in a bench scale fluid bed combustor to provide order-of-magnitude estimates of the kinetic parameters - activation energy, frequency factor, and apparent order of reaction - to approximate the reactivity of these fuels in the combustor. The estimates are based on an external surface area (or particle diameter), and, therefore, do not provide information on the true kinetic behaviour of individual particles undergoing combustion. However, they are adequate for the prediction of kinetic behaviour in the fluid bed environment and are assumed to apply to conditions within the riser of a circulating fluidized bed.

In order to obtain an estimate of the solids circulation rate, experiments were performed to measure this rate as a function of gas velocity, P/S ratio, and average riser solids loading. These were carried out using a novel time-of-descent method referred to as "line-and-sinker" in which a cage immersed in the moving packed bed descends attached to a chain which passes through a sealed port to the outside of the upper region of the standpipe. When the chain is released, the cage travels downward with the moving bed. The rate at which the chain is pulled through the port gives a measure of the velocity at which solids are descending through the standpipe.

The control model was validated under manual inspection by performing step response tests on the UBC pilot CFB combustor and comparing the results with simulated predictions. Because constant fuel feedrates could not be ensured during experimentation, direct step

responses could not be compared. Instead, 10-minute averages of the measured fuel feedrate from experimentation were input to the simulator, and the behaviour compared to that from the experiments. Very good agreement was generally obtained between experiment and simulation. Discrepancies have been shown to arise from unmodelled disturbances (e.g. changing bed behaviour characteristics) rather than from basic model concepts.

The primary factor affecting the temperature response of the combustor was found to be the thermal capacity of the materials of construction. The rate at which the structural materials gain or lose heat has a significant influence on the speed with which the combustor temperature can be changed. At the outset, it was expected that char size would be a tuning parameter. However, investigation of the effect of this parameter over a range (600  $\mu\text{m}$  to 1 mm) showed that it had limited effect on the transients and confirmed the importance of the refractory thermal capacity.

A brief investigation of the effect of the structural thermal capacity was carried out. This involved approximation of an "idealized" 20 MW<sub>t</sub> combustor in which all heat is absorbed by the heat exchange surfaces. A 10-fold reduction in the response time of temperature to changes in fuel feedrate was indicated. This idealization does not take into account the effects of the steam side, which might be expected to become a controlling factor for large scale units. Investigation of the effect of solids inventory in this idealized combustor shows that, although a greater inventory does slow the transient, the effect is very small (or negligible) - doubling the total inventory of solids increases the response time by less than 20%.

Application of the simulator in control scheme testing was demonstrated by utilizing a similar scheme to that employed in the pilot plant combustor. The action of the simulator was similar to that observed experimentally.

In parallel with the modeling work, advanced control methods were applied to the pilot CFB combustor. This included process identification through open-loop PRBS testing and time series analysis, application of single loop GPC control of combustion temperature,

and multivariable control of the combustor to meet heat removal, flue gas oxygen concentration, and combustion temperature specifications.

The identification tests did not provide sufficient information for complete identification because changing bed properties, as well as other factors, precluded operating the pilot plant for a sufficient length of time. This is not expected to be a problem on an industrial scale. Due to incompleteness, time series analysis of the data tends to indicate higher order models than would probably be necessary if complete identification were possible. Although model order is questionable, this is not as important as it might seem. Since the controller may well operate under conditions of changing fuel type, where knowledge of the model for the new fuel is lacking, it would be advantageous to employ an over-parameterized model in order to be able to account for changing dead-time and possible inverse responses.

Further validation of the dynamic model is provided by a comparison of discrete time models obtained from experimentation and simulation. These discrete time models are essential for model-based control, such as GPC, considered in this work. The results indicate that, on the time scale considered for control, the simulation provides discrete time model representations of the process similar to those determined from dynamic experiments on the pilot plant. It is concluded that the dynamic model can be used for controller behaviour simulation. The identification tests also confirm the validity of the performance of the adaptor within the GPC controller. Comparison of the step responses of the discrete time models identified from the PRBS testing and from the on-line adaptor show these to be essentially equivalent.

Finally, the applicability of GPC within a CFB combustor framework is demonstrated. The complete control structure takes advantage of the differing time constants of the various parameters within the system. The water outlet temperature control setpoint can be realized in the order of a few seconds; flue gas oxygen (since flue gas optimization is not considered), can be maintained within  $\pm 1\%$  with a detuned PID controller adjusting the total air flow



based on this measurement; a GPC controller adjusts the rotary valve control signal in order to maintain combustion temperature requirements. These three loops can be treated separately since (1) water temperature response to solids loading is extremely fast; (2) oxygen concentration need only remain within acceptable limits, allowing detuning; (3) the temperature response is strongly affected by the thermal capacity of the complete system, making it slow and ideal for long range predictive control where control actions are based on predictions of the response.

## **8.2 Conclusions**

A dynamic model has been developed for a CFB combustor to predict the responses of combustion temperature, flue gas oxygen concentration, and rate of heat removal to changes in fuel feedrate, air flowrates, and solids circulation rate. The validity of this model has been verified by a comparison of predicted values with those transients measured experimentally as well as through comparison of control models developed through simulation with those identified from the UBC pilot CFBC. In simulation form, this model provides a valuable tool for control structure design and testing. In addition, simulators are always useful as operator trainers.

For an estimation of the reactivity of the various fuels employed, a very simple experimental method has been described. This method uses batch combustion tests in a bench scale fluid bed reactor to provide estimates of frequency factor, activation energy, and apparent order of reaction for the particular fuel in question. These are simple, inexpensive tests which provide adequate estimates of these parameters.

A novel, time-of-descent measurement for the solids circulation rate was presented. Termed "line-and-sinker", it is a direct measurement of the particle descent velocity in the standpipe to determine the circulation rate of the solids. One drawback of this method is that it is intrusive, requiring plant shutdown for installation or removal. In addition, it does not provide continuous measurement of the solids circulation rate.

Generalized Predictive Control (GPC) has been shown to be a viable method for combustion control. Its ease of employment and reliability have been demonstrated in this application. This is due to its robustness in the presence of model errors and mismatch. Additionally, time series analysis has been presented and demonstrated as a method for identification in process control. Although the results obtained in this work were less than ideal, they did provide appropriate estimates of the transfer functions in question.

### **8.3 Recommendations for Further Work**

The dynamic model developed here has been shown to predict the behaviour of the UBC pilot CFB combustor. At this point, it is not applicable to units of industrial scale. Necessary assumptions and equipment-specific correlations, such as the solids loading profile and the particle convective component of the heat transfer coefficient to the heat exchange surface and solids circulation rate, render the simulator system-specific. Without such information, full-scale units cannot be successfully simulated. In addition, it is likely that the steam side may play a significant role in determining the temperature transients for industrial CFB boilers. The inclusion of a model for the behaviour of the steam plant would be necessary to complete the simulator. Because of the modular nature of programs such as this, with the necessary information available, appropriate additions or modifications can easily be incorporated to provide a powerful tool for the design of these processes.

The assumption of temperature uniformity makes the model somewhat fuel specific. This assumption is reasonable for most of the solid fuels investigated in the pilot plant but, incorporating a heat release model to allow for a non-uniform temperature distribution would extend the validity of the model, at the expense of increased computation. This addition would also enable simulation of gaseous and liquid fuels.

The particular control schemes employed in this work were presented for demonstration purposes. Although they performed well on the pilot plant, they are as yet unproven on an industrial level. Different control schemes could be considered. For example,

depending on the various response times, a multivariable GPC controller could control steam quality, combustion temperature and oxygen concentration together. Alternatively, steam quality and combustor temperature could be controlled with a multivariable GPC controller, with oxygen controlled conventionally by a feedforward controller from fuel feedrate and a feedback controller providing trim.

Process identification testing, as carried out here, can be easily performed on the industrial scale. The methods used are non-intrusive and, when properly designed, do not adversely affect product quality. In addition, analytical methods are available which enable identification, using dither signals, of the plant operating in closed loop.

## NOMENCLATURE

- $a$  - decay constant ( $m^{-1}$ ) (equation 2.8)
- $a_b$  - interfacial bubble area per unit bubble volume ( $m^2/m^3$ )
- $a_t$  - driving force in noise model (white noise)
- $ab$  - cross-sectional inlet area of cyclone ( $m^2$ )
- $A$  - frequency factor ( $((kmol/m^2 \cdot s)/(kmol/m^3)^n)$ )
- $A(q^{-1})$  - polynomial acting on  $y_t$  in CARIMA model (equation 5.8)
- $A_{ris}$  - cross-sectional area of riser ( $m^2$ )
- $A_{st}$  - cross-sectional area of standpipe ( $m^2$ )
- $A_{to}$  - total external heat transfer surface area ( $m^2$ )
- $b$  - process dead-time
- $B(q^{-1})$  - polynomial acting on  $u_t$  in CARIMA model (equation 5.8)
- $c_b$  - bulk oxygen concentration ( $kmol/m^3$ )
- $c_s$  - surface concentration of oxygen ( $kmol/m^3$ )
- $c_0, c_1, c_2$  - fitted constants in equation 4.2
- $C(q^{-1})$  - polynomial representing noise in CARIMA model (equation 5.8)
- $C_{Aout}/C_{Ain}$  - dimensionless outlet concentration
- $C_{pw}$  - heat capacity of water ( $J/kg \cdot K$ )
- $C_{wz}(k)$  - crosscovariance between series  $w_t$  and  $z_t$  at lag  $k$
- $C_{zz}(k)$  - autocovariance of series  $z_t$  at lag  $k$
- $d_b$  - bubble diameter ( $m$ )
- $d_c$  - char particle diameter ( $m$ )
- $d_{ci}$  - initial diameter of char particle ( $m$ )
- $d_{ct}$  - char particle diameter at time  $t$  ( $m$ )
- $d_p$  - particle diameter ( $m$ )

$D$	- oxygen diffusivity ( $\text{m}^2/\text{s}$ )
$D_c$	- riser diameter (m)
$e_p$	- particle emissivity
$e_{\text{surf}}$	- emissivity of heat transfer surface
$e_{\text{susp}}$	- suspension emissivity
$E$	- entrainment flux at riser exit ( $\text{kg}/\text{m}^2\cdot\text{s}$ )
$E_a$	- activation energy ( $\text{kPa}\cdot\text{m}^3/\text{kmol}$ )
$E_j(q^{-1})$	- polynomial defined by Diophantine identity (equation 5.15)
$E_o$	- entrainment flux at dense phase/dilute phase interface ( $\text{kg}/\text{m}^2\cdot\text{s}$ )
$E_\infty$	- entrainment flux above TDH ( $\text{kg}/\text{m}^2\cdot\text{s}$ )
$F_j(q^{-1})$	- polynomial defined by Diophantine identity (equation 5.15)
$f_v$	- volume fraction of particle clusters in the fast-fluidized bed
$Fr$	- Froude number based on superficial gas velocity (equation 4.6)
$Fr_t$	- Froude number based on particle terminal velocity (equation 4.7)
$g$	- acceleration of gravity, $9.81 \text{ m/s}^2$
$G$	- process transfer function
$G_j$	- step response of plant to time $t + j$
$G_s$	- net solids mass flux ( $\text{kg}/\text{m}^2\cdot\text{s}$ )
$G_{si}$	- hypothetical solids mass flux at solids return level ( $\text{kg}/\text{m}^2\cdot\text{s}$ )
$G_{sj}$	- hypothetical solids mass flux at riser exit ( $\text{kg}/\text{m}^2\cdot\text{s}$ )
$G^m$	- internal model transfer function
$G'_j(q^{-1})$	- polynomial defined by equation (5.20)
$G^*$	- controller transfer function
$h$	- height above bed interface (m)
$h_c$	- cyclone cylinder height (m)
$h_i$	- combustor inside heat transfer coefficient ( $\text{W}/\text{m}^2\cdot\text{K}$ )
$h_o$	- combustor outside heat transfer coefficient ( $\text{W}/\text{m}^2\cdot\text{K}$ )

$h_{ti}$	- inside heat transfer coefficient to exchanger tubes ( $\text{W/m}^2\cdot\text{K}$ )
$h_{to}$	- outside heat transfer coefficient to exchanger tubes ( $\text{W/m}^2\cdot\text{K}$ )
$h_{gc}$	- gas convective component of heat transfer ( $\text{W/m}^2\cdot\text{K}$ )
$h_{pc}$	- particle convective component of heat transfer ( $\text{W/m}^2\cdot\text{K}$ )
$h_{rad}$	- radiative component of heat transfer ( $\text{W/m}^2\cdot\text{K}$ )
$H$	- cyclone overall height (m)
$j$	- number of samples into the future from time $t$
$k$	- lag from time $t$ in correlation analysis
$k_c$	- kinetic rate constant based on external surface area ( $(\text{kmol/m}^2\cdot\text{s})/(\text{kmol/m}^3)^n$ )
$k_d$	- diffusion coefficient (m/s)
$k_{ss}$	- thermal conductivity of stainless steel ( $\text{W/m}\cdot\text{K}$ )
$k_w$	- thermal conductivity of water ( $\text{W/m}\cdot\text{K}$ )
$k_q$	- bubble phase to dense phase mass transfer coefficient (m/s)
$k_1'$	- dimensionless kinetic rate constant
$K(t)$	- gain matrix in RLS
$m$	- exponent used in equation (4.19)
$m_{\text{co}_2}$	- molar flow of $\text{CO}_2$ ( $\text{kmol/s}$ )
$\dot{m}$	- mass flowrate of cooling water ( $\text{kg/s}$ )
$\dot{m}_s$	- solids mass flowrate ( $\text{kg/s}$ )
$M_{ci}$	- mass charged to reactor (kg)
$M_{ct}$	- mass of char at time $t$ (kg)
$M_p$	- mass of single particle charged to reactor (kg)
$n$	- apparent order of reaction also exponent used in equation (4.19)
$n_c$	- number of char particles
$na$	- order of polynomial $A(q^{-1})$
$nb$	- order of polynomial $B(q^{-1})$
$nc$	- order of polynomial $C(q^{-1})$

$N$	- process noise
$N_t$	- discrete time process noise
$N_1$	- minimum horizon for prediction
$N_2$	- maximum horizon for prediction
$Nu$	- Nusselt number
$p$	- ratio of carbon monoxide to carbon dioxide formed
$P(t)$	- covariance matrix in RLS
$P/S$	- primary to secondary air ratio
$Pr$	- Prandtl number
$q_{hx}$	- rate of heat transfer to heat exchanger (W)
$q_s$	- volumetric flowrate of solids ( $m^3/s$ )
$q^{-1}$	- backward shift operator
$Q_w$	- flowrate of cooling water ( $m^3/s$ )
$r$	- order of polynomial $\delta(y)$ or radius (m)
$r_c$	- overall rate of combustion ( $kmol/m^2 \cdot s$ )
$r_i$	- inside radius of combustor (m)
$r_o$	- outside radius of combustor (m)
$r_{ti}$	- inside diameter of heat transfer tubes (m)
$r_{to}$	- outside diameter of heat transfer tubes (m)
$r_{wz}(k)$	- crosscorrelation function of series $w_t$ and series $z_t$
$r_{zz}(k)$	- autocorrelation function of series $z_t$
$R$	- gas constant ( $8.314 \text{ kPa} \cdot m^3/kmol \cdot K$ )
$Re$	- Reynolds number
$Re_p$	- particle Reynolds number
$Re_w$	- Reynolds number of water in exchanger tubes
$s$	- order of polynomial $\omega(u)$
$Sc$	- Schmidt number, $\mu/\rho_g \cdot D$

$Sh$	- Sherwood number, $k_d d_c / D$
$t$	- time (s)
$\bar{t}_g$	- mean gas residence time in cyclone (s)
$\bar{t}_s$	- mean solids residence time in cyclone (s)
$T$	- temperature (K)
$T(q^{-1})$	- fixed design polynomial replacing $C(q^{-1})$ in CARIMA model (equation 5.8)
$T_g$	- gas temperature (K)
$T_p$	- particle temperature (K)
$T_s$	- temperature of surroundings (K)
$T_{switch}$	- switching period in PRBS test
$T_{surf}$	- temperature of heat transfer surface (K)
$T_{susp}$	- suspension temperature (K)
$T_w$	- heat exchanger tube wall temperature (K)
$T_{waterin}$	- water inlet temperature to heat exchanger (K)
$T_{waterout}$	- water outlet temperature from heat exchanger (K)
$u_g$	- inlet gas velocity to cyclone (m/s)
$u_A$	- absolute rise velocity of bubbles (m/s)
$u_t$	- value of process input or control action at time $t$
$u^f$	- filtered input
$U$	- gas velocity (m/s)
$U_g$	- superficial gas velocity (m/s)
$U_c$	- velocity at incipient rapid fluidization (m/s) (Li et al., 1981))
$U_{mf}$	- superficial gas velocity at minimum fluidization (m/s)
$U_o$	- overall heat transfer coefficient based on combustion side ( $W/m^2 \cdot K$ )
$U_{tr}$	- transport velocity (m/s)
$U^*$	- dimensionless gas velocity (Grace, 1982)
$v_d$	- descent velocity of moving packed bed in standpipe (m/s)



- $v_t$  - particle terminal velocity (m/s)
- $V_c$  - cyclone volume excluding vortex finder (m<sup>3</sup>)
- $V_p$  - particle velocity (m/s)
- $V_{nl}$  - volume of cyclone at natural length (m<sup>3</sup>)
- $V_s$  - annular volume above exit duct to middle of entrance duct of cyclone (m<sup>3</sup>)
- $w(t)$  - process setpoint at time  $t$
- $X$  - number of mass transfer units
- $X_I$  - solids downflow ratio (Mori et al., 1991)
- $y$  - process output or controlled variable
- $y_{sp}$  - setpoint
- $y_t$  - value of process output at time  $t$
- $y^f$  - filtered output
- $\hat{y}(t + j)$  - prediction of  $y(t+j)$  at time  $t$
- $z$  - vertical coordinate of riser (m)
- $z_i$  - height of rapid fluidized bed (m) (Li et al., 1981)
- $\bar{z}$  - mean of series  $z_t$
- $z_0$  - reference height (m)
- $Z_o$  - characteristic length, defined by equation (2.7)

#### Greek Symbols

- $\alpha$  - thermal diffusivity (m<sup>2</sup>/s)
- $\beta$  - ratio of gas flow via bubble phase to total gas flow
- $\delta(y)$  - polynomial acting on  $y_t$  in linear difference equation of model (equation 5.4)
- $\Delta$  - differencing operator ( $1 - q^{-1}$ )
- $\Delta P_{cyc}$  - pressure drop across solids separator (kPa)
- $\Delta P_{lv}$  - pressure drop around L-valve to riser base (kPa)
- $\Delta P_{ris}$  - pressure drop across riser (kPa)

- $\Delta P_{st}$  - pressure drop across standpipe moving bed (kPa)
- $\Delta T_{LMTD}$  - overall log mean temperature difference (K)
- $\Delta T_o$  - temperature difference between gas phase and tube wall (K)
- $\Delta \rho$  - effective density ( $\rho_s - \rho_g$ ) (kg/m<sup>3</sup>)
- $\varepsilon$  - voidage
- $\varepsilon_a$  - voidage of particle clusters
- $\varepsilon_b$  - fraction of bed volume occupied by bubbles
- $\varepsilon_i$  - voidage at  $z_i$
- $\varepsilon_{mf}$  - voidage at minimum fluidization
- $\varepsilon_r$  - voidage at solids return level
- $\varepsilon'$  - voidage of dilute phase
- $\Phi$  - mechanism factor
- $\Gamma_j(q^{-1})$  - polynomial defined by equation (5.20)
- $\eta$  - ratio of  $k_c$  from equation (A.19) to that from equation (A.18)
- $\varphi$  - slip factor
- $\varphi(q^{-1})$  - demonimator polynomial of noise model
- $\lambda$  - forgetting factor
- $\lambda_j$  - control action weighting factor
- $\mu$  - gas viscosity (kg/m·s)
- $\mu_w$  - viscosity of water (kg/m·s)
- $v(q^{-1})$  - impulse response function
- $\theta(q^{-1})$  - numerator polynomial of noise model
- $\rho_{ave,hx}$  - average suspension density around heat exchange surface (kg/m<sup>3</sup>)
- $\rho_c$  - apparent density of char (kg/m<sup>3</sup>)
- $\rho_{cyc}$  - cyclone solids holdup (kg/m<sup>3</sup>)
- $\rho_g$  - gas density (kg/m<sup>3</sup>)
- $\rho_f$  - solids loading at top of profile (kg/m<sup>3</sup>)

- $\rho_p$  - particle density ( $\text{kg/m}^3$ )
- $\rho_s$  - apparent density of particles ( $\text{kg/m}^3$ )
- $\rho_w$  - density of water ( $\text{kg/m}^3$ )
- $\rho_z$  - solids loading at height  $z$  ( $\text{kg/m}^3$ )
- $\rho_0$  - reference solids loading at height  $z_0$  ( $\text{kg/m}^3$ )
- $\sigma$  - Stefan-Boltzmann constant,  $5.67 \times 10^{-8} \text{ W/m}^2 \cdot \text{K}^4$
- $\sigma_\xi^2$  - variance of white noise
- $\omega$  - segregation coefficient (Li and Kwauk, 1980)
- $\omega(u)$  - polynomial acting on  $u_i$  in linear difference equation of model (equation 5.4)
- $\xi$  - diffusive coefficient (Li and Kwauk, 1980)
- $\xi(t)$  - normally distributed white noise
- $\xi^f$  - filtered white noise
- $\zeta$  - decay constant in equation 4.1 ( $\text{m}^{-1}$ )

## **REFERENCES**

- Arena, U., A. Malandrino, and L. Massimilla**, "Modelling of Circulating Fluidized Bed Combustion of a Char", *CJChE*, V. 69, 860-868, Aug. 1991.
- Aström, K.J. and T. Bohlin**, "Numerical Identification of Linear Dynamic Systems from Normal Operating Records", in *Theory of Self-Adaptive Control Systems*, ed. P.H. Hammond, Plenum Press, 1966.
- Aström, K.J. and B. Wittenmark**, "On Self-Tuning Regulators", *Automatica*, V. 9, 185-199, 1973.
- Arthur, J.R.**, "Model Reactions Between Carbon and Oxygen", *Trans. Faraday Society*, 47, 164-178, 1951.
- Bader, R., J. Finlay, and T.M. Knowlton**, "Gas/Solids Flow Patterns in a 30.5-cm-diameter Circulating Fluidized Bed", in *Circulating Fluidized Bed Technology II*, ed. P. Basu and J.F. Large, Pergamon Press, Oxford, Boston, 123-137, 1988.
- Basu, P.**, "Heat Transfer in High Temperature Fast Fluidized Beds", *Chem. Eng. Sci.*, V. 45, N. 10, 3123-3136, 1990.
- Basu, P. and S.A. Fraser**, Circulating Fluidized Bed Boilers - Design and Operations, Butterworth-Heinemann, 1991.
- Behie, L.A.**, "ENCH 623 - Chemical Reactor Design", Course notes from University of Calgary, 1987.
- Beisswenger, H., S. Darling, L. Plass, and A. Wechsler**, "Burning Multiple Fuels and Following Load in the Lurgi/Combustion Engineering Circulating Fluid-Bed Boiler", in *Proceedings of the Eighth International Conference on Fluidised Bed Combustion*, V II, 619-633, 1985.
- Brereton, C.M.H.**, "Fluid Mechanics of High Velocity Fluidized Beds", Ph.D. Thesis, University of British Columbia, Canada, 1987.
- Box, G.E.P. and G.M. Jenkins**, Time Series Analysis - Forecasting and Control, Holden Day, San Francisco, 1970.
- Burkell, J.J., J.R. Grace, J. Zhao, and C.J. Lim**, "Measurement of Solids Circulation Rates in Circulating Fluidized Beds", in *Circulating Fluidized Bed Technology II*, ed. P. Basu and J.F. Large, Pergamon Press, Oxford, 501-509, 1988.

**Chakraborty, R.K. and J.R. Howard**, "Combustion of a Single Particle in Fluidized Beds of High-Density Alumina", J. Inst. Energy, V. 54, 55-58, 1981.

**Clarke, D.W. and P.J. Gawthrop**, "Self-Tuning Controller", Proc. IEEE, V. 122, 924, 1975, and V. 126, 633, 1979.

**Clarke, D.W., C. Mohtadi, and P. Tuffs**, "Generalized Predictive Control: Parts I and II", Automatica, V. 23, 137-148 and 149-160, 1987a, b.

**Cordero, A.O. and D.Q. Mayne**, "Deterministic Convergence of a Self-Tuning Regulator with Variable Forgetting Factor", IEE Proc.-PtD., V. 128, 19, 1981.

**Davidson, J.F., and Harrison, D.**, Fluidized Particles, Cambridge University Press, Cambridge, 1963.

**Dion, J.M., L. Dugard, Nguyen Minh Tri**, "MIMO Adaptive Generalized Predictive Control with Input-Output Constraints", Adaptive Control of Chemical Processes 1988, (ADCHEM '88), 2<sup>nd</sup> International IFAC Symposium, 15-20, 1988.

**Durbin, J.**, "The Fitting of Time Series Models", Rev. Int. Inst. Stat., V. 28, 233, 1960.

**Field, M.A., D.W. Gill, B.B. Morgan, and P.G.W. Hawksley**, Combustion of Pulverised Coal, The British Coal Utilisation Research Association, Leatherhead, 1967

**Fisher, D.G.**, "Process Control: An Overview and Personal Perspective", CJChE, V. 69, 5-26, 1991.

**Fortescue, T.R., L.S. Kershenbaum, and B.E. Ydstie**, "Implementation of Self-Tuning Regulators with Variable Forgetting Factors", Automatica, V. 17, 831-835, 1981.

**Glicksman, L.R.**, "Circulating Fluidized Bed Heat Transfer", in Circulating Fluidized Bed Technology II, ed. P. Basu and J.F. Large, Pergamon Press, Oxford, 13-29, 1988.

**Grace, J.R.**, "Fluidized Bed Heat Transfer", Chapter 8.2 in Handbook of Multiphase Systems, ed. G. Hetsroni, Hemisphere, Washington, 1982.

**Grace, J.R.**, "Heat Transfer in Circulating Fluidized Beds", in Circulating Fluidized Bed Technology, ed. P. Basu, Pergamon Press, Toronto, 63-81, 1986a.

**Grace, J.R.**, "Fluidized Beds as Chemical Reactors", in Gas Fluidization Technology, ed. D. Geldart, John Wiley and Sons, 1986b.

**Grace, J.R.**, "High-Velocity Fluidized Bed Reactors", Chem. Eng. Sci., V. 45, 1953-1966, 1990.

**Grace, J.R. and J. Tuot**, "Cluster Formation in Vertically Conveyed Suspensions of Intermediate Density", *Trans. Inst. Chem. Engrs.*, V. 57, 49-54, 1979.

**Halder, P.K. and P. Basu**, "Mass Transfer from a Coarse Particle to a Fast Bed of Fine Solids", *A.I.Ch.E. Symp. Ser.*, V. 84, N. 262, 58-67, 1988.

**Hartge, E.-U., D. Rensner, and J. Werther**, "Solids Concentration and Velocity Patterns in Circulating Fluidized Beds", in *Circulating Fluidized Bed Technology II*, ed. P. Basu and J.F. Large, Pergamon Press, Oxford, 165-180, 1988.

**Holman, J.P.**, Heat Transfer, 2<sup>nd</sup> edition, McGraw, New York, 1968.

**Horio, M.**, "Hydrodynamics of Circulating Fluidization - Present Status and Research Needs", in *Circulating Fluidized Bed Technology III*, ed. P. Basu, M. Horio, and M. Hasatani, Pergamon Press, Oxford, 3-14, 1991.

**Horio, M. and K. Morisita**, *Japanese J. Multiphase Flow*, V. 2, 117, 1988.

**Hyppanen, T., Y.Y. Lee, A. Kettunen, and J. Riiali**, "Dynamic Simulation of a CFB Based Utility Power Plant", in *Fluidized Bed Combustion, Proc. 12<sup>th</sup> Int. Fluidized Bed Combustion Conference*, ASME, 1121-1127, 1993.

**Kang, S.K., T.W. Kwon, and S.D. Kim**, "Hydrodynamic Characteristics of Cyclone Reactors", *Powder Tech.*, V. 58, 211-221, 1989.

**Karri, S.B.R and T.M. Knowlton**, "A Practical Definition of the Fast Fluidization Regime", in *Circulating Fluidized Bed Technology III*, ed. P. Basu, M. Horio, and M. Hasatani, Pergamon Press, Oxford, 67-72, 1991.

**Kehoe, P.W.K., and Davidson, J.F.**, *Inst. Chem. Eng. (Lond.) Symp. Ser.*, V. 33, 97, 1971.

**Knowlton, T.M.**, "Solids Transfer in Fluidized Systems", in *Gas Fluidization Technology*, ed. D. Geldart, John Wiley and Sons Ltd., Chichester, U.K., 341-314, 1986.

**Kobro and Brereton**, in *Circulating Fluidized Bed Technology*, ed. P. Basu, Pergamon Press, Toronto, 1986.

**Kullendorff, A., B. Jansson, and J. Olofsson**, "Operating Experience of Circulating Fluidized Beds", *Circulating Fluidized Bed Technology*, ed. P. Basu, Pergamon Press, Toronto, 83, 1986.

**Kunii, D. and O. Levenspiel**, "Flow Modeling of Fast Fluidized Beds", in *Circulating Fluidized Bed Technology III*, ed. P. Basu, M. Horio, and M. Hasatani, Pergamon Press, Oxford, 91-98, 1991.

**Lanneau, K.P.**, "Gas-Solids Contacting in Fluidized Beds", Trans. Instn. Chem. Engrs., V. 38, 125-143, 1960.

**Large, J.F., Y. Martinie, and, M.A. Bergougnou**, "Interpretive Model for Entrainment in a Large Gas Fluidized Bed", International Powder Bulk Solids Handling and Process Conference, May 1976.

**Leckner, B.**, "Heat Transfer in Circulating Fluidized Bed Boilers", in Circulating Fluidized Bed Technology III, ed. P. Basu, M. Horio, and M. Hasatani, Pergamon Press, Oxford, 27-38, 1991.

**Leith, D. and W. Licht**, "The Collection Efficiency of Cyclone Type Particle Collectors - A New Theoretical Approach", AIChE Symp. Ser., V. 68, N. 126, 196-206, 1972.

**Li, Y., and M. Kwauk**, "The Dynamics of Fast Fluidization", in Fluidization, ed. J.R. Grace and J.M. Matsen, Plenum Press, New York, 537-544, 1980.

**Li, Y., B. Chen, F. Wang, Y. Wang, and M. Guo**, "Rapid Fluidization", Int. Chem. Eng., V. 21, 670-678, 1981.

**MacGregor, J.F. and P.A. Taylor**, Advanced Process Control - an Intensive Short Course, McMaster University, Canada, 1990.

**Mahalingam, M. and A.K. Kolar**, "Experimental Investigations of Heat Transfer at the Wall of a Circulating Fluidized Bed", in Fluidized Bed Combustion, Proc. 12<sup>th</sup> Int. Fluidized Bed Combustion Conference, ASME, 341-348, 1993.

**Martin, H.**, "Heat and Mass Transfer in Fluidized Beds", Chem. Ing. Tech., V. 52, 199-209, 1980.

**Martin, H.**, "Heat Transfer Between Gas Fluidized Beds of Solid Particles and the Surfaces of Immersed Heat Exchanger Elements, Part I", Chem. Eng. Process., V. 18, 157-169, 1984.

**McIntosh, A.R., S.L. Shah, and D.G. Fisher**, "Analysis and Tuning of Adaptive Generalized Predictive Control", CJChE, V. 69, N. 1, 97-110, 1991.

**Monceaux, L., M. Azzi, Y. Molodtsov, and J.F. Large**, "Overall and Local Characterization of Flow Regimes in a Circulating Fluidized Bed", in Circulating Fluidized Bed Technology, ed. P. Basu, Pergamon Press, Toronto, 185-191, 1986.

**Mori, S., K. Narukawa, I. Yamada, T. Takebayashi, H. Tanii, Y. Tomoyasu, and T. Mii**, "Dynamic Model of a Circulating Fluidized Bed Coal Fired Boiler", in Fluidized Bed Combustion, Proc. 11<sup>th</sup> Int. Fluidized Bed Combustion Conference, ASME, 1261-1266, 1991.

**Orcutt, J.C., J.F. Davidson, and R.L. Pigford**, "Reaction Time Distributions in Fluidized Catalytic Reactors", Chem. Eng. Prog., Symp. Ser., V. 58, N. 38, 1-15, 1962.

**Patience, G.S., J. Chaouki, F. Berruti, and R. Wong**, "Scaling Considerations for Circulating Fluidized Bed Risers", Powder Tech., V. 72, 31-37, 1992.

**Perry, R.H. and D.W. Green**, Perry's Chemical Engineer's Handbook, 6<sup>th</sup> edition, McGraw-Hill, Inc., 1984.

**Reh, L.**, personal communication with J.R. Grace, 1985.

**Rhodes, M.J. and D. Geldart**, "A Model for the Circulating Fluidized Bed", Powder Tech., V. 53, 155-162, 1987.

**Rhodes, M.J., P. Lausmann, F. Villain, and D. Geldart**, "Measurement of Radial and Axial Solids Flux Variations in the Riser of a Circulating Fluidized Bed", Circulating Fluidized Technology II, ed. P. Basu and J.F. Large, Pergamon Press, Oxford, 155-164, 1988.

**Sahagian, J.**, "Circulating Combustion", in First Annual Fluidized Bed Conference Proceedings, Council of Industrial Boiler Owners, Fairfax Station, Virginia, 1985.

**Senior, R.C.**, "Circulating Fluidised Bed Fluid and Particle Mechanics: Modelling and Experimental Studies with Application to Combustion", Ph.D. Thesis, University of British Columbia, Canada, 1992.

**Senior, R.C. and C. Brereton**, "Modeling of Circulating Fluidized Bed Solids Flow and Distribution", Chem. Eng. Sci., V. 47, 281-296, 1992.

**Sleicher, C.A. and M.W. Rouse**, "A Convenient Correlation for Heat Transfer to Constant and Variable Property Fluids in Turbulent Pipe Flow", Int. J. Heat Mass Transfer, V. 18, 677, 1975.

**Stephanopoulos, G.**, Chemical Process Control - An Introduction to Theory and Practice, PTR Prentice Hall, N.J., 1984

**Stockdale, W. and R. Stonebridge**, "The Design and Commercialisation of Multi-Solids Fluidised Bed Combustors", in Coaltech '85, London 5<sup>th</sup> Int. Conf. and Ex. on Coal Util. and Trade, 1985.

**Subbarao, D. and P. Basu**, "Heat Transfer in Circulating Fluidized Beds", in Circulating Fluidized Bed Technology, ed. P. Basu, Pergamon Press, Toronto, 281-286, 1986.

**Thomas, J.F., R.W. Gregory, and M. Takayasu**, "Atmospheric Fluidized Bed Boilers for Industry", IEA Coal Research Report # 1CTIS/TR35, IEA Coal Research, London, Nov. 1986.



**Weinstein, H., M. Shao, and M. Schnitzlein**, "Radial Variation in Solid Density in High Velocity Fluidization", in *Circulating Fluidized Bed Technology*, ed. P. Basu, Pergamon Press, Toronto, 201-206, 1986.

**Weiss, V. and Fett, F.N.**, "Modeling the Decomposition of Sodium Bicarbonate in a Circulating Fluidized Bed Reactor", in *Circulating Fluidized Bed Technology*, ed. P. Basu, Pergamon Press, Toronto, 167-172, 1986.

**Weiss, V., J. Schöler, and F.N. Fett**, "Mathematical Modeling of Coal Combustion in a Circulating Fluidized Bed Reactor", in *Circulating Fluidized Bed Technology II*, ed. P. Basu and J.F. Large, Pergamon Press, Oxford, 289-298, 1988.

**Wen, C.Y. and L.H. Chen**, "Fluidized Bed Freeboard Phenomena: Entrainment and Elutriation", *AIChE J.*, V. 28, 117-128, 1982.

**Wu., R.L.**, "Heat Transfer in Circulating Fluidized Beds", Ph.D. Thesis, University of British Columbia, Canada, 1989.

**Wu, R.L., C.J. Lim, and J.R. Grace**, "The Measurement of Instantaneous Local Heat Transfer Coefficients in a Circulating Fluidized Bed", *CJChE*, V. 67, 301-307, 1989.

**Yerushalmi, J., N.T. Cankurt, D. Geldart, and B. Liss**, "Flow Regimes in Vertical Gas-Solid Contact Systems", *AIChE Symp. Ser.*, V. 74, N. 176, 1-13, 1978.

**Yerushalmi, J.**, "High Velocity Fluidized Beds", in *Gas Fluidization Technology*, ed. D. Geldart, John Wiley and Sons, Chichester, U.K., 1986.

**Yu, L.**, "Experimental Study on the Hydrodynamics in Circulating Fluidized Bed with Wide Particle Size Distribution", M.Sc. Thesis, Tsinghua University, China, 1989.

**Zhang, L., T.D. Li, Q.Y. Zheng, and C.D. Lu**, "A General Dynamic Model for Circulating Fluidized Bed Combustion Systems with Wide Particle Size Distributions", in *Fluidized Bed Combustion*, Proc. 11<sup>th</sup> Int. Fluidized Bed Combustion Conference, ASME, 1289-1294, 1991.

## **APPENDIX A - KINETIC RATE DETERMINATION**

### **A.1 Introduction**

This appendix documents supplemental experimental work undertaken to provide some estimation of the "reactivity" of the various fuels used in this research, in particular for simulation of results obtained on the UBC pilot CFBC. The method used is not considered ideal by any means. It is only intended to provide rough estimates for rates of combustion.

The chemistry and physics of coal reactions are fundamental to the combustion process. Significant emphasis has been given to the modeling of these reactions, but the work has not yet advanced to the point where it is widely used in process development (Smoot, 1984).

A coal particle, as described by its proximate analysis, is comprised of moisture, volatiles, carbon and ash. The proportion of each of these components depends on the source and rank of the coal.

The burning process is typically considered as taking place in three interactive stages - drying (driving off moisture), devolatilization (gasification of volatiles), and char combustion (oxidation of char). The time required for char combustion is usually at least an order of

magnitude greater than for drying and devolatilization so that char combustion is considered the controlling factor in the burnout times of coal. In this study, experiments were carried out on char particles (dried and devolatilized coal) and on petroleum coke where it can be assumed that drying and devolatilization have been already completed.

## **A.2 Experimental Apparatus**

This work was carried out to provide estimates of the kinetic parameters ( $n$ ,  $A$ , and  $E_a$ ) used in the dynamic model presented in Chapter 4. The tests were performed in a bench scale fluid bed facility.

Figure A.1 gives a flow diagram of the system, while Figures A.2 and A.3 provide further details of the unit. The reactor is a 50 mm OD quartz cylindrical tube, with an ID of 46 mm and a height of 1020 mm. It is supported via ceramic sleeves by two stainless steel plates. The bed is heated by two electrical heaters surrounding the column. A gas preheater (usually maintained at 300 °C) is employed to heat the inlet gas.

Due to heater positioning, it was found that the region close to the distributor was not adequately heated. To ensure a uniform bed temperature, ceramic fibre, supported by a stainless steel rod, was placed on top of the silicon carbide distributor to raise the bed approximately 500 mm up the column to a location where it could be adequately heated. This fibre became the distributor for the fluid bed.

The inlet gas was a mixture of nitrogen and air, mixed prior to introduction to the reactor. Nitrogen, air, and total gas flowrates were measured with rotameters. The oxygen concentration of the inlet mixture was determined by running a sample of the inlet stream through the analyser bank. The gas analysis system was the same as that used for the CFB combustor and is fully described in Chapter 3.

The reactor temperature is measured via a thermocouple immersed in the center of the fluid bed. This temperature is maintained by the setting of the external electrical heaters. The flue gas from the reactor exits to the exterior of the building, after sampling, via a duct and a

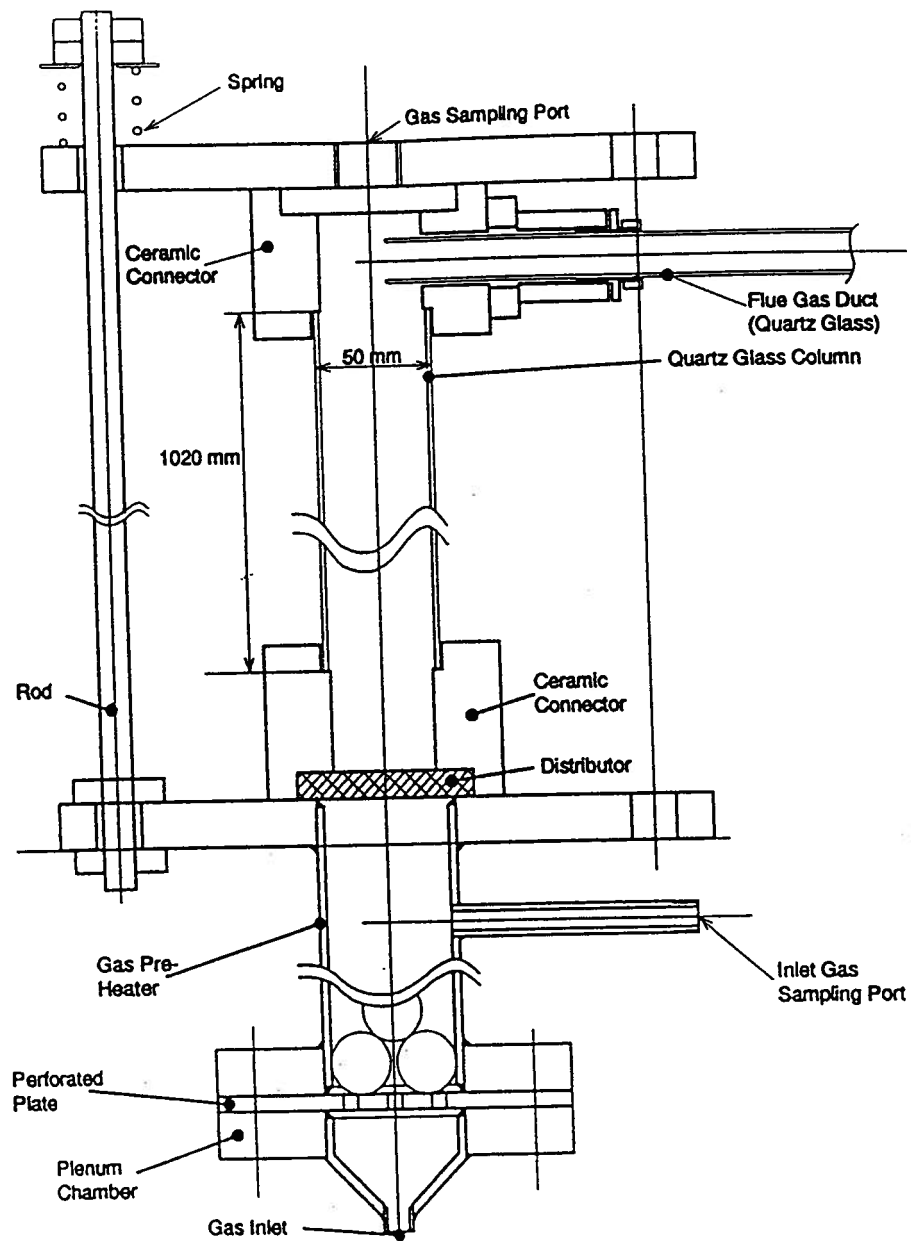
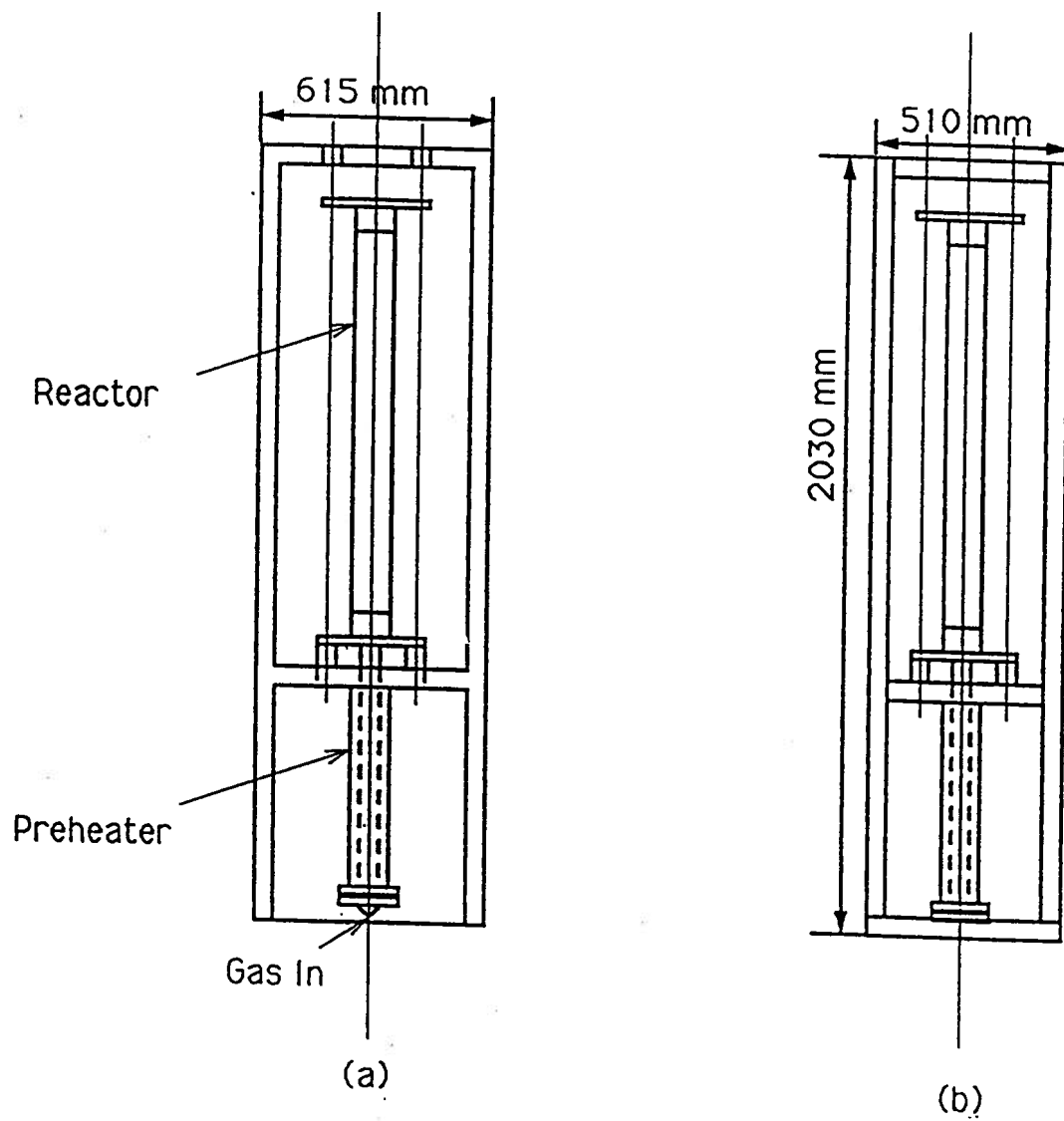


Figure A.1: Schematic Diagram of the 50 mm Diameter Fluidized Bed Reactor



**Figure A.2: External View of the Bench Scale Fluidized Bed Reactor - (a) front view and  
(b) side view**

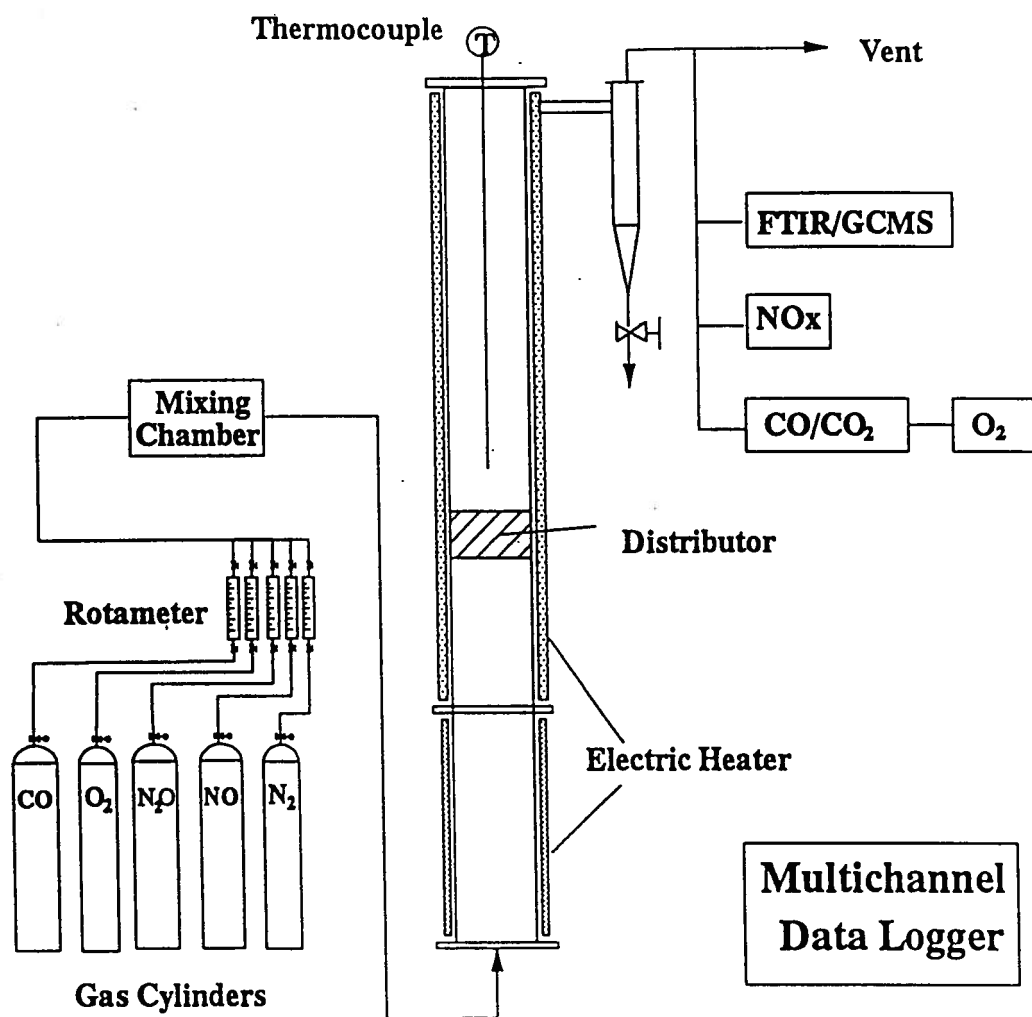


Figure A.3: Details of the Bench Scale Reactor Setup

porous stainless steel filter. A valve is located after this filter to allow pressurization of the system to maintain sufficient positive pressure (approx. 300 mm or 12" H<sub>2</sub>O) to assure adequate flow through the gas sample train.

For each experiment, the reactor was charged with 200 g of 500  $\mu\text{m}$  silica sand as bed material, and the unit was heated to the desired operating temperature with the electrical heaters. During this process, air was passed through the reactor to ensure uniform heating of the bed. Once the desired temperature had been reached, the air was shut off and nitrogen was passed through the bed to purge all oxygen from the system.

When all oxygen had been removed, as indicated by the analysers, the char sample was dropped into the bed through a port at the top of the column. The appropriate amount of air was then mixed with the nitrogen to begin the test. Nominally, the inlet gas flowrate was 1.5 kg/h using a 5% O<sub>2</sub>, N<sub>2</sub> mixture. Rather than sample the inlet gas to determine the true inlet O<sub>2</sub> concentration, the flue gas oxygen concentration measured after burnout of the char is taken as the inlet concentration.

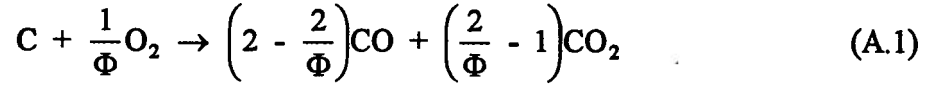
There is a certain delay in, and filtering of, concentration measurements, related to the length of the sample lines and analyser cell volumes and responses. The CO/CO<sub>2</sub> analyser is the first in the sample train. Measurements indicate a delay of 15 to 20 s followed by a 95% rise time of approximately 3 to 4 s. The time delay is accommodated by taking the time of start of combustion as the time when the CO/CO<sub>2</sub> analyser first responds, as opposed to the time of introduction of air. Burnout times are of the order of 10 min. This duration of the burnout period is much larger than the response time of this analyser, so that the latter can safely be ignored.

On the other hand, the oxygen analyser at the end of the sample train experiences a delay of over 1 minute, with samples reaching it filtered by the preceding 4 analysers. The high degree of prefiltering makes dynamic measurements, on the time scale experienced here, somewhat difficult to analyze. For this reason, as explained later, only the CO<sub>2</sub> response is used (and needed) to analyze the results; the oxygen analysis is only used to determine a final

steady-state oxygen concentration (after the char burnout); this concentration is assumed to correspond to the inlet concentration.

### **A.3 Theory and Analysis**

The dynamic model (Chapter 4) assumes that all carbon reacting with oxygen is converted to CO<sub>2</sub>, and the same assumption is applied in this work. In practice, char burns according to the reaction:



Field et. al (1967), in the modelling of pulverized coal firing, define  $\Phi$  as a mechanism factor with a value of 1 for CO<sub>2</sub> transport away from the particle, and 2 for CO transport. The mechanism factor increases with increasing temperature and decreasing char size. Field et. al (1967) express  $\Phi$  as:

$$\Phi = \frac{2p+2}{p+2} \quad \text{for } d_c < 0.005 \text{ cm} \quad (A.2)$$

and

$$\Phi = \frac{(2p+2) - p(d_c - 0.005)/0.095}{p+2} \quad \text{for } 0.005 < d_c < 0.1 \text{ cm} \quad (A.3)$$

where  $p$  is the ratio of carbon monoxide to carbon dioxide formed. From Arthur (1951):

$$p = 2500 \exp\left(\frac{-5.19 \times 10^7}{RT}\right) \quad (A.4)$$



At typical CFB combustion temperatures (850 °C),  $\Phi$  takes on a value of 1 for 1 mm char particles and 1.3 for 0.6 mm char particles. The data analysis usually does not include estimated particle diameters of less than 700  $\mu\text{m}$ . Hence, the assumption of complete conversion to  $\text{CO}_2$ , which is consistent with the assumptions applied in the model, is not expected to create any significant error in the analysis.

The reactor itself is assumed to be well mixed, with adequate exchange of gas between the emulsion and bubble phase to allow for a mean gas concentration throughout the reactor. This assumption is reasonable since the size of the bed employed here is much smaller than required for fully developed fluidization. It is doubtful that the bed is deep enough to allow for extensive growth and coalescence of bubbles. The flow is entirely transitional over the length of the bed. There is little advantage in employing an advanced reactor model and incorporating its inherent errors into the analysis when the intended application requires only order of magnitude estimates.

Rather than attempting to justify this assumption through characterization of the movement of gas and solids phases, a more direct approach provides partial justification - the comparison of conversion based on a bubbling bed model, and that based on a well mixed reactor model assumed here. For this comparison, one of the models of Orcutt et al. (1962) has been selected for the bubbling bed. The key assumptions of this model are that no solids are associated with the bubble phase, while gas is assumed to be in plug flow in the bubble phase and perfectly mixed in the dense phase.

For a first order, irreversible reaction, the Orcutt model yields a dimensionless outlet concentration of:

$$\frac{C_{Aout}}{C_{Ain}} = \frac{1 - \beta e^{-X} + \beta k_1' e^{-X}}{1 - \beta e^{-X} + k_1'} \quad (\text{A.5})$$

where:  $C_{Aout}/C_{Ain}$  = dimensionless outlet concentration  
 $k_1'$  = dimensionless kinetic rate constant (Grace, 1986b)  
 $X$  = number of mass transfer units  
 $\beta$  = ratio of gas flow via bubble phase to total gas flow

The gas flow ratio,  $\beta$ , is assumed to be given by:

$$\beta = \frac{U_g - U_{mf}}{U_g} \quad (A.6)$$

where:  $U_g$  = superficial gas velocity (m/s)  
 $U_{mf}$  = superficial gas velocity at conditions of minimum fluidization (m/s)

The number of mass transfer units,  $X$ , here is given by:

$$X = \frac{k_q a_b \epsilon_b L_f}{\beta u_A} \quad (A.7)$$

where:  $a_b$  = interfacial bubble area per unit bubble volume ( $m^2/m^3$ )  $\approx 6/d_b$   
 $k_q$  = bubble phase to dense phase mass transfer coefficient ( $ms^{-1}$ )  
 $\epsilon_b$  = fraction of bed volume occupied by bubbles  $\approx (U - U_{mf})/u_A$   
 $u_A$  = absolute rise velocity of bubbles (m/s)

with  $k_q$  (Davidson and Harrison, 1963) given by:

$$k_q = 0.75U_{mf} + \frac{0.975g^{0.25}D^{0.5}}{d_b^{0.25}} \quad (A.8)$$

where :  $D$  = diffusivity ( $m^2/s$ )  
 $d_b$  = bubble diameter (m)  
 $g$  = acceleration due to gravity =  $9.81 \text{ m/s}^2$

Application of these equations to this equipment provides a rough estimate of 2 for  $X_b$ .

The limiting case of a perfectly mixed reactor for a first-order reaction is given (Grace, 1986b) by:

$$\frac{C_{Aout}}{C_{Ain}} = \frac{1}{1 + k_1'} \quad (A.9)$$

Table A.1 provides a comparison of the kinetic rate  $k_1'$  calculated from the Orcutt model (equation A.5) and the well mixed reactor (equation A.9) for various conversions. The % difference in this table is determined from:

$$\% \text{difference} = 100 \left( 1 - \frac{\left( \frac{C_{Aout}}{C_{Ain}} \right)_{eq.(A.9)}}{\left( \frac{C_{Aout}}{C_{Ain}} \right)_{eq.(A.5)}} \right) \quad (A.10)$$

Typical dimensionless outlet concentrations in the experiments reported here range from 0.6 to 1. Table A.1 indicates that, if the Orcutt model provides an effective description of the hydrodynamics, the well-mixed reactor assumption is likely to incorporate no more than 6 % difference in kinetic rates calculated from the data.

The rate of char combustion is directly related to the flow rate of  $CO_2$  leaving the system. This reaction rate can be expressed in terms of an exposed char external surface area by:

**Table A.1: Comparison of Orcutt Bubbling Bed Model with Well-Mixed Assumption**

$\frac{C_{Aout}}{C_{Ain}}$	$k_1'_{(eq.(A.5))}$	$k_1'_{(eq.(A.9))}$	% difference
0.2	6.6	4	39
0.4	1.76	1.5	14
0.6	0.71	0.67	6
0.8	0.26	0.25	4
1.0	0	0	0

$$r_c = \frac{m_{CO_2}}{\pi d_c^2 n_c} \quad (A.11)$$

where:  $d_c$  = char particle diameter (m)  
 $m_{CO_2}$  = molar flow of  $CO_2$  (kmol/s)  
 $n_c$  = number of char particles  
 $r_c$  = rate of combustion based on external surface area (kmol/m<sup>2</sup>.s)

The number of particles in the reactor,  $n_c$ , is estimated from the mass introduced to the reactor and the mass of a single particle:

$$n_c = \frac{M_{ci}}{M_p} \quad (A.12)$$

where:  $M_{ci}$  = mass charged to reactor (kg)  
 $M_p$  = mass of single particle charged to reactor (kg)

The mass of each particle introduced (assuming it is spherical) is defined by:

$$M_p = \frac{\pi}{6} d_{ci}^3 \rho_c \quad (A.13)$$

where:  $d_{ci}$  = initial char particle diameter (m)  
 $\rho_c$  = apparent char density (kg/m<sup>3</sup>)

It is assumed that the majority of combustion takes place at or near the particle surface. Hence a shrinking particle type model is utilized. Ash is assumed to break off the particle due to rigorous agitation within this bed, and thus no ash layer is considered.

The particle diameter at any time  $t$  can then be estimated from:

$$d_{\alpha} = \left( \frac{6M_{\alpha}}{\pi n_c \rho_c} \right)^{1/3} \quad (\text{A.14})$$

where:  $d_{\alpha}$  = char particle diameter at time  $t$  (m)  
 $M_{\alpha}$  = mass of char at time  $t$  (kg)  
 $= M_{ci} - \int_0^t m_{\infty_2} dt$

At pseudo steady-state, the rate of diffusion of oxygen to the particle surface is the same as the kinetic rate at the surface:

$$r_c = k_c c_s^n = k_d (c_b - c_s) \quad (\text{A.15})$$

where:  $c_b$  = bulk oxygen concentration (kmol/m<sup>3</sup>)  
 $c_s$  = surface concentration of oxygen (kmol/m<sup>3</sup>)  
 $k_c$  = kinetic rate constant (kmol/m<sup>2</sup>·s·(kmol/m<sup>3</sup>) <sup>$n$</sup> )  
 $k_d$  = mass transfer coefficient (m/s)  
 $n$  = apparent reaction order

The apparent reaction order,  $n$ , takes into account and corrects for assumptions applied in this analysis and in the dynamic model. Specifically, it corrects for errors in the assumption of complete conversion to CO<sub>2</sub> at the particle surface, as well as the assumption of combustion occurring on the external surface of the particle, neglecting any internal diffusion.

The surface concentration of oxygen,  $c_s$ , can be determined by equating the rate from equation (A.11) to the expression on the right hand side of equation (A.15).

From the definition of the Sherwood number, the mass transfer coefficient,  $k_d$ , is:

$$k_d = \frac{\text{Sh}D}{d_p} \quad (\text{A.16})$$

where:  $D$  = diffusivity ( $\text{m}^2/\text{s}$ )  
 $Sh$  = Sherwood number

For a fluidized bed, the Sherwood number,  $Sh$ , is given by Halder and Basu (1988) by:

$$Sh = 2\epsilon_{mf} + 0.69 \left( \frac{Re}{\epsilon_{mf}} \right)^{1/2} Sc^{1/3} \quad (\text{A.17})$$

where:  $Sc$  = Schmidt number =  $\mu_g / \rho_g D$

Having determined the oxygen surface concentration,  $c_s$ , from equation (A.15), the kinetic rate constant,  $k_c$ , can be calculated from:

$$k_c = \frac{r_c}{c_s^n} \quad (\text{A.18})$$

The temperature dependency of  $k_c$  is given by the Arrhenius expression:

$$k_c = A \exp \left( \frac{-E_a}{RT_p} \right) \quad (\text{A.19})$$

where:  $A$  = frequency factor ( $\text{kmol}/\text{m}^2 \cdot \text{s} \cdot (\text{kmol}/\text{m}^3)^n$ )

$E_a$  = activation energy ( $\text{kJ}/\text{kmol}$ )

$R$  = gas constant =  $8.314 \text{ (kJPa} \cdot \text{m}^3/\text{kmol} \cdot \text{K)}$

$T_p$  = particle temperature ( $\text{K}$ )

A particle heat balance is required to estimate the temperature of the burning particle,  $T_p$ , in equation (A.19). This is accomplished by equating the heat released on combustion to the convective and radiative heat transfer:

$$r_c \Delta H_c = h_p (T_p - T_g) + \epsilon_{mf} \sigma (T_p^4 - T_g^4) \quad (\text{A.20})$$

where:  $h_p$  = convective heat transfer coefficient ( $\text{W/m}^2 \cdot \text{K}$ )  
 $\Delta H_c$  = heat of combustion ( $\text{J/kmol}$ )  
 $T_g$  = temperature of surroundings ( $\text{K}$ )  
 $\epsilon_{mf}$  = emissivity  
 $\sigma$  = Stephan-Boltzmann constant ( $5.670 \times 10^{-8} \text{ W/m}^2 \cdot \text{K}^4$ )

This equation can be solved to determine  $T_p$ .

It is assumed here that the char particle is at a uniform temperature. This assumption is justified in the same manner as for the sand in Section 4.3.4. In this case,  $h(V/A)/k$  is estimated to be 0.036 to justify this assumption.

The analysis of the data can now be summarized. Given a particular data set, values for  $A$  and  $E_a$  in Equation (A.19) as well as  $n$  in Equation (A.18) are assumed. The overall rate of combustion,  $r_c$ , is determined from the data using Equation (A.11). The surface concentration of oxygen,  $c_s$ , is then estimated by means of Equation (A.15) and used to calculate the experimental kinetic rate coefficient,  $k_c$ , from Equation (A.18) based on the assumed reaction order  $n$ . A calculated value for  $k_c$  is next obtained from Equation (A.19) and the particle heat balance (Equation A.20). The experimental and calculated kinetic rate constants can then be compared in terms of the ratio:

$$\eta = \frac{k_{c(\text{eq. (A.19)})}}{k_{c(\text{eq. (A.18)})}} \quad (\text{A.21})$$



This procedure is carried out on the data set from the start of combustion to a point where the estimated particle size is approximately 700  $\mu\text{m}$ .

An appropriate value for the reaction order,  $n$ , can be determined by investigation of  $\eta$  over the burnout period selected;  $n$  is adjusted until a constant  $\eta$  with respect to time is obtained.

Experimental values of  $k_c$  from equation (A.18) can then be substituted into equation (A.19) from two data sets at different temperatures to solve for  $A$  and  $E_a$ . A third independent data set at an intermediate temperature is then analysed to ensure that  $\eta$ , defined by equation (A.21) is in fact 1 over the burnout time, implying a good fit.

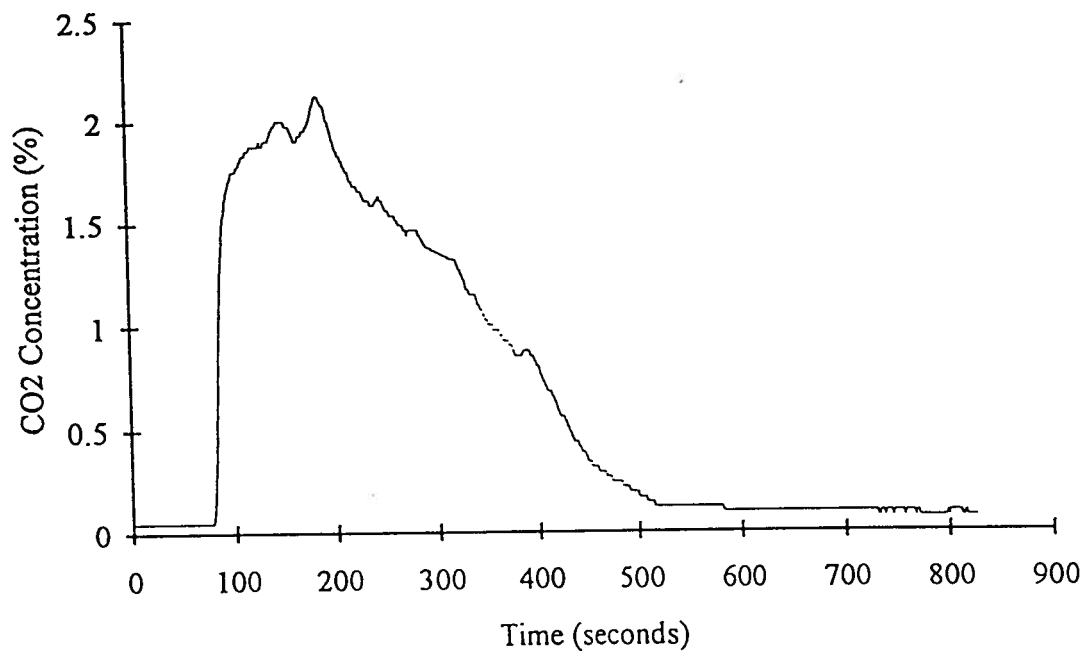
#### **A.4 Results**

Two fuels were investigated for the purpose of kinetic parameter estimation - Highvale coal char and Conoco coke. Various fuel sizes were employed for these tests. Average char diameters were 1.6 mm and 0.95 mm for Highvale Coal char, and 2.2 mm, 1.5 mm, 1.1 mm, and 0.8 mm for the Conoco Coke. The burnout for each particle size for each fuel was measured at three different reactor temperatures: 800 °C, 850 °C and 900 °C.

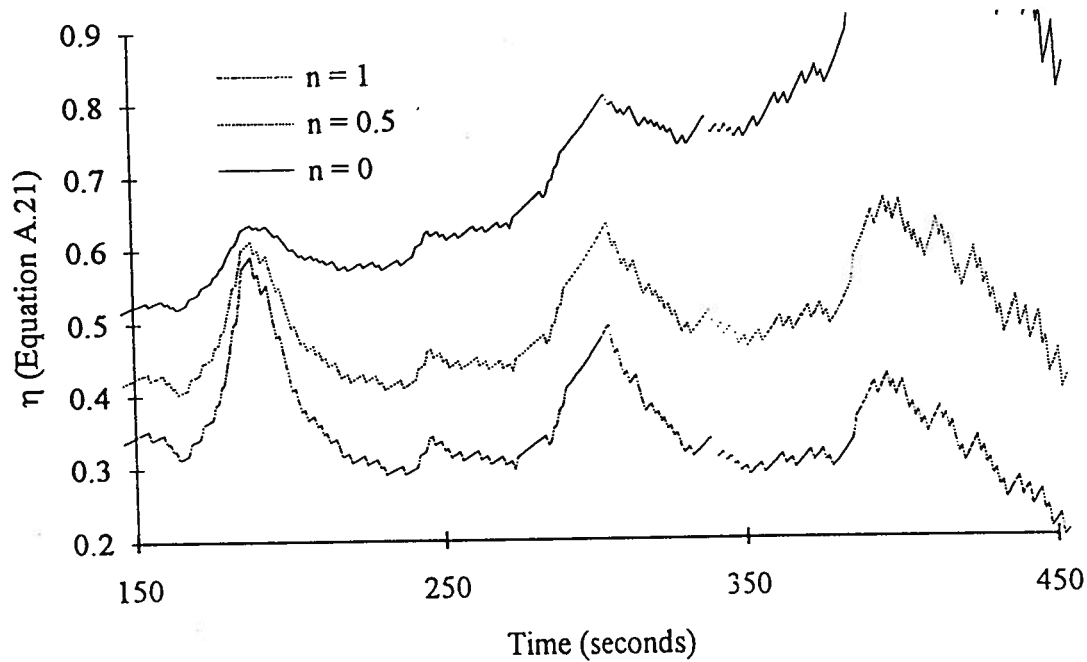
A typical  $\text{CO}_2$  vs. time profile is presented in Figure A.4. In the test portrayed, a 1 g sample of 1.6 mm diameter Highvale char particles was charged to the reactor operating at 810 °C and the corresponding burnout recorded.

The effect of reaction order,  $n$ , in this particular test is shown in Figure A.5. As mentioned before,  $n$  is a fitting parameter which covers applied assumptions. Since a value of  $n = 0.9$  provides a relatively constant ratio,  $\eta$ , over the analysed burnout time range, this value was selected to represent Highvale coal char.

Figure A.6 presents the effect of temperature on the ratio. This error at different temperatures is then applied to equation (A.19) which provides values for  $A$  and  $E_a$ .

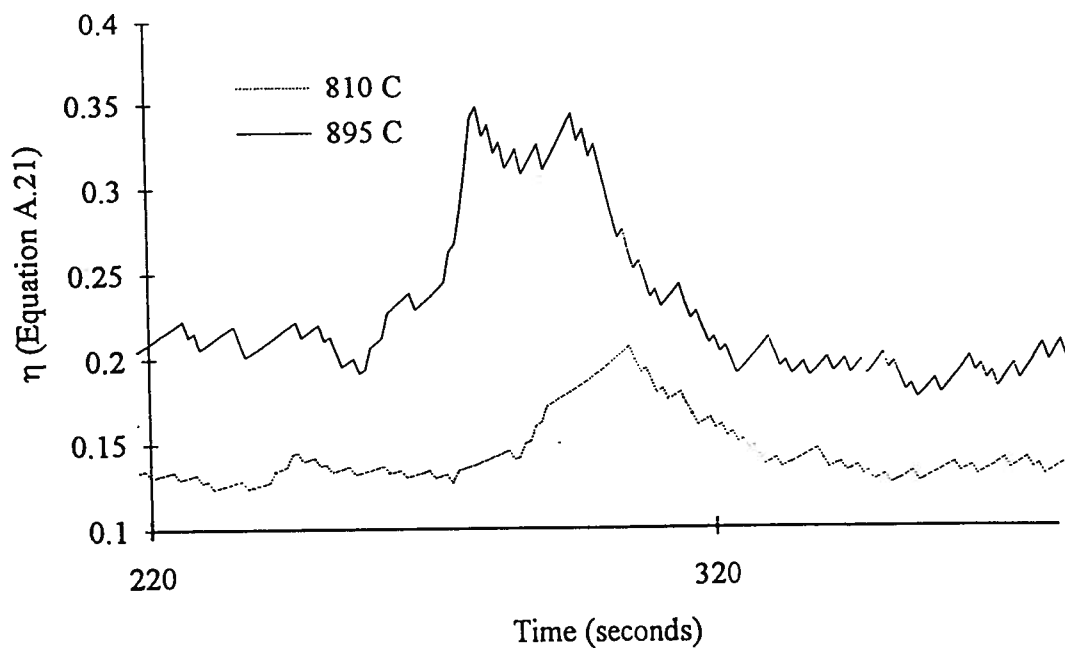


**Figure A.4: Typical Carbon Dioxide Response Curve during Char Burnout for Highvale Coal Char Particles of Initial Diameter of 1.6 mm at a Reactor Temperature of 810 °C**



**Figure A.5: Effect of Reaction Order,  $n$ , on Calculated Kinetic Rate Constant**

- Highvale Coal Char with Initial Particle Diameter of 1.6 mm
- Reactor Temperature = 810 °C
- $A = 6 \text{ (kmol/m}^2\cdot\text{s}\cdot\text{(kmol/m}^3)^n\text{)}, E_a = 1 \times 10^7 \text{ (kJ/kmol)}$

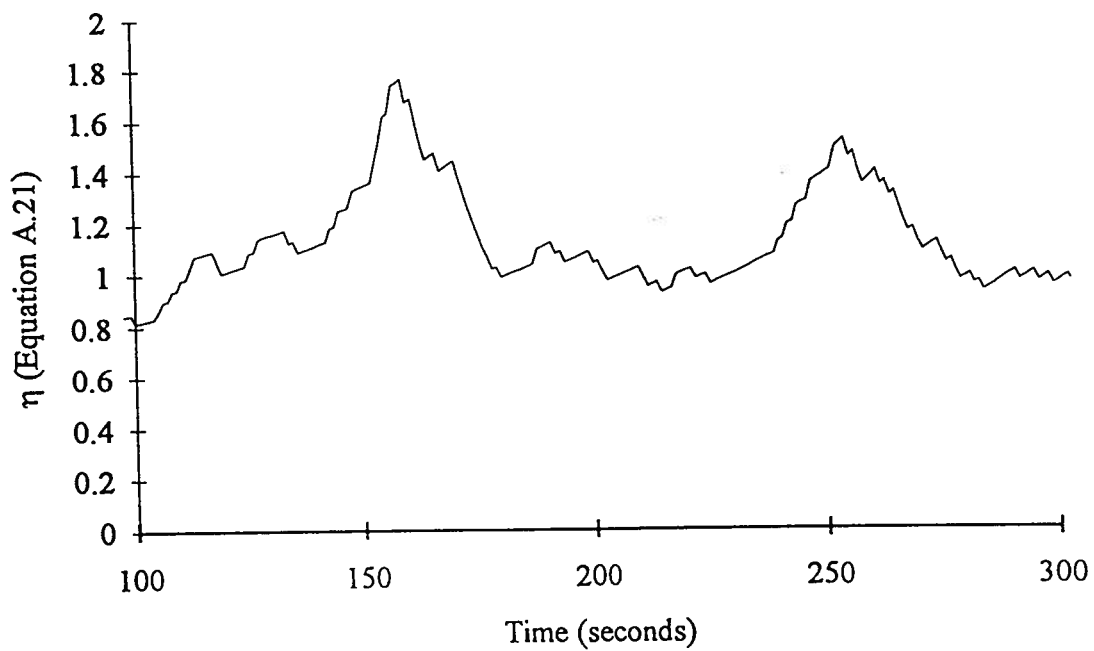


**Figure A.6: Effect of Temperature on Kinetic Rate Constant**

- Highvale Coal Char with Initial Particle Diameter of 1.6 mm
- $n = 0.9$ ,  $A = 6 \text{ (kmol/m}^2\text{-s}\cdot\text{(kmol/m}^3\text{)}^n\text{)}$ ,  $E_a = 1 \times 10^7 \text{ (kJ/kmol)}$

Analysis of a third data set at yet another temperature was performed to confirm the above determined values. This confirmation is achieved as shown in Figure A.7 by a relatively constant  $\eta$  of 1.

This approach was applied to both fuels. The parameters determined in this manner are presented in Table A.2.



**Figure A.7: Confirmation of Determined Kinetic Parameters**

- Highvale Coal Char with Initial Particle Diameter of 1.6 mm
  - Reaction Temperature = 850 °C
- $n = 0.9$ ,  $A = 1092 \text{ (kmol/m}^2\cdot\text{s}\cdot\text{(kmol/m}^3)^n\text{)}$ ,  $E_a = 8.0733 \times 10^7 \text{ (kJ/kmol)}$ .

**Table A.2: Results of Fuel Reactivity Experiments**

Fuel	Order, n	Frequency Factor, A ( $\text{kmol/m}^2 \cdot \text{s} \cdot (\text{kmol/m}^3)^n$ )	Activation Energy, $E_a$ ( $\text{kJ/kmol}$ )
Highvale Coal	0.9	1092	$8.0733 \times 10^7$
Conoco Coke	1.0	44.4	$4.99 \times 10^7$

## **APPENDIX B - SOLIDS CIRCULATION RATE MEASUREMENT**

This appendix summarizes work performed to measure the solids circulation rate in the UBC pilot CFB combustor. The solids circulation rate determines the bed carbon inventory and the rate at which this bed carbon is returned to the riser. This information is necessary for the modeling described in Chapter 4. Burkell et al. (1988) review various techniques for determining solids circulation rates in circulating fluidized beds.

### **B.1 Theory**

This work uses a time-of-descent method to determine the solids circulation rate. Time-of-descent involves measurement of the velocity of the particles in the moving packed bed within the standpipe.

The particle velocity measurement is commonly made by observing individual particles through the wall of a plexiglass column, or through windows installed in the standpipe. Plug flow is generally assumed so that this velocity is taken as the descent velocity of all particles in the moving packed bed,  $v_d$  (m/s).



The moving bed is assumed to have a voidage equal to that at minimum fluidization,  $\epsilon_{mf}$ . The volumetric flowrate of solids,  $\dot{q}_s$  (m<sup>3</sup>/s), is therefore:

$$\dot{q}_s = v_d A_{st} (1 - \epsilon_{mf}) \quad (B.1)$$

where  $A_{st}$  is the cross-sectional area of the standpipe (m<sup>2</sup>).

The circulation rate,  $\dot{m}_s$  (kg/s), is obtained by multiplying the volumetric flow of solids by the solids density,  $\rho_s$  (kg/m<sup>3</sup>):

$$\dot{m}_s = v_d \rho_s (1 - \epsilon_{mf}) A_{st} \quad (B.2)$$

Finally, the net solids flux in the riser,  $G_s$  (kg/m<sup>2</sup>·s), can be calculated by dividing the solids circulation rate by the cross-sectional area of the riser,  $A_{ris}$  (m<sup>2</sup>):

$$G_s = v_d \rho_s (1 - \epsilon_{mf}) \frac{A_{st}}{A_{ris}} \quad (B.3)$$

## **B.2 Experimental Apparatus**

Visual observation is unreliable at the temperatures of combustion used in this work (800 to 900 °C). Although a quartz glass window (400 mm tall by 50 mm wide) was in place in the lower portion of the standpipe, it was difficult to discern and track individual particles due to the red glowing of the solids at these temperatures. As well, although plug flow is assumed, the flat wall may influence particle velocities in its vicinity so that particles at the wall may not be representative of the bed.

This work employed a method, referred to here as the line-and-sinker technique, whereby the velocity of the bed could be measured as a bulk velocity from within the moving mass. This employed a stainless steel cage, suspended by a chain, inserted into the bed. The

rate at which this cage is pulled downward with the moving packed bed is taken to be the velocity of the bed itself.

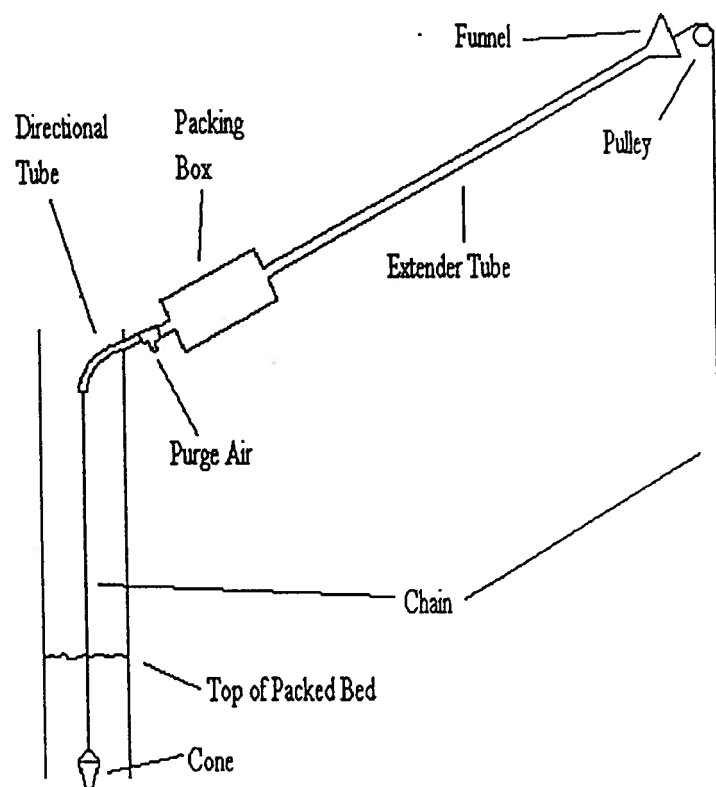
Figure B.1 is a sketch of the line-and-sinker system. A view port located near the top of the standpipe was replaced with the apparatus shown in this figure.

A stainless steel welded link chain of an approximate 6 mm outer diameter is used to suspend a conical cage within the moving bed. The stainless steel hollow cone is approximately 40 mm in diameter at the top and 10 mm at the bottom with six 5 mm holes drilled around its circumference. This design, developed through trial and error, allows the cone to be caught by the bed and pulled downward with the moving mass while permitting it to be raised through the bed without excessive force.

The external, free hanging end of the chain passes over a pulley and through a funnel into the extender tube. The pulley allows for a smooth change of direction from the vertical hanging position to that of the entry port to the standpipe. The funnel ensures that the chain is guided into the extender tube freely, without getting caught on any sharp, or abrupt edges.

The extender tube, approximately 2 m long, serves two purposes. First it moves the hanging portion of the chain beyond the pilot plant structure so that it may hang freely and not become caught in various ancillaries. Second, it provides containment of hot portions of the chain. When, during testing, the chain is pulled to raise the cage, the hot portion of the chain being extracted from the standpipe remains in this feeding structure preventing the possibility of operator contact.

The packing box is a large cylinder, 400 mm long and 100 mm in diameter, packed with Fibrefrax Durablanket insulation. This provides a crude seal between the standpipe and the atmosphere. During installation, the chain is wrapped tightly with insulation which is then stuffed into the cylinder. The chain is pulled back and forth through the insulation until it moves freely with no resistance. The packing box provides tight restriction in the line, so that some backpressure may be maintained while providing minimal resistance to chain movement.



**Figure B.1: Schematic of Line-and-Sinker System**

Prior to entry into the standpipe, a tee allows purge air to be introduced. This air increases the pressure in the feedline to minimize escape of hot gas to the atmosphere, and to keep solids from flowing into the chain-containment system where they could cause blockages and restrict free movement of the chain.

The directional tube enables the smooth transition of the chain to the vertical direction in the standpipe, helping to minimize restrictions to free movement. As well, it is located such that the cage, welded to the end of the chain, is located approximately along the centreline of the moving bed.

For each experiment, the cage is located such that it is approximately 1.5 to 2 m above the vertex of the L-valve. A very strong pulling indicates that the bed is, in fact, under packed conditions and that the cage is within this bed. The chain is then released and the time required for a 1 m length to pass into the system is measured. The chain can then be pulled to raise the cage and repeat the measurement. In this work, the measurement was repeated 5 times for each condition considered.

### **B.3 Experimental Results**

For all of the tests reported here, as the chain passed into the bed, it would typically move 5 to 10 cm, pause, then move again. This is indicative of stick/slip flow often reported for these types of systems. Consequently, the reported measurements are time averages of the solids circulation rate.

The three factors which determine the solids circulation rate, or net solids flux, are gas velocity, primary to secondary (P/S) air ratio, and average riser solids loading. Run #33 employed the line-and-sinker method to measure solids rates as a function of these variables. The results are presented, in terms of net riser solids flux, in Table B.1. The values reported here were found to be reproducible to  $\pm 5\%$ .

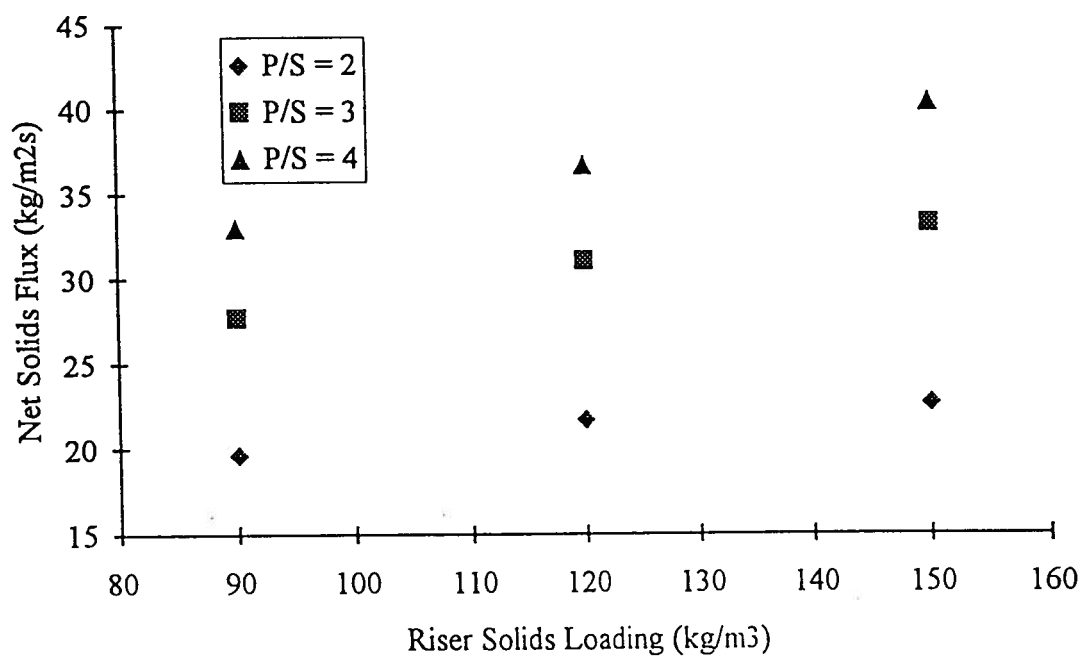
Figures B.2 to B.10 present these results graphically. Figures B.2 to B.4 show the effects of solids loading in the riser; Figures B.5 to B.7 illustrate the effects of P/S air ratio;

Figures B.8 to B.10 present the effects of gas velocity. It is seen that there is very little effect of solids loading on the net solids flux. This is indicative of a high sensitivity of riser solids loading to changes in the solids circulation rate.

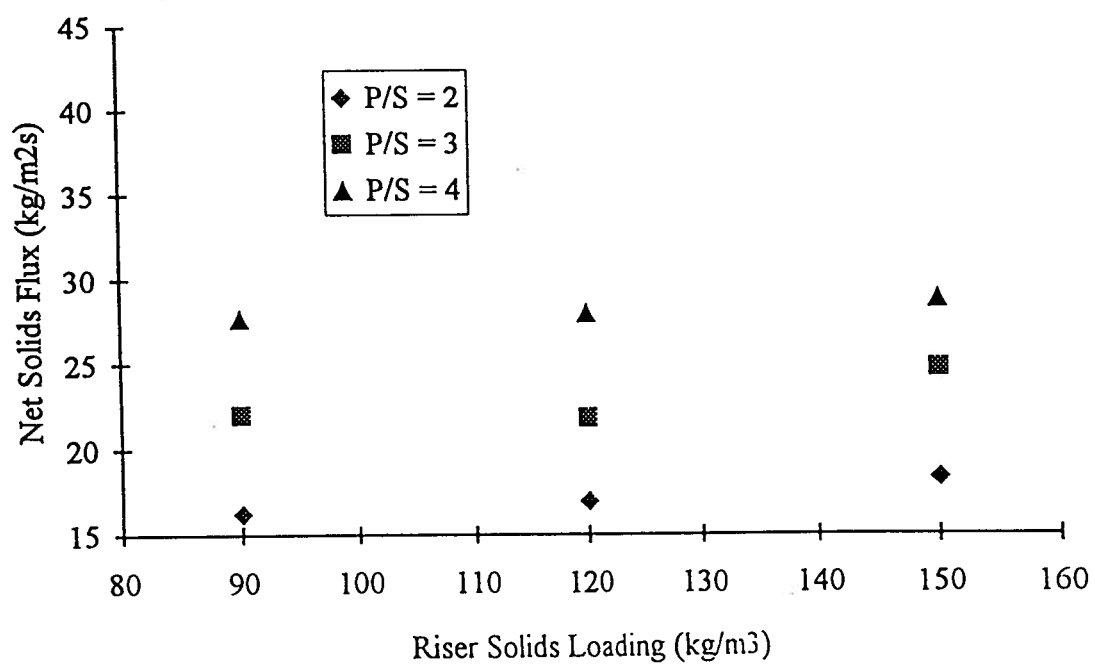
In contrast, the gas velocity and P/S air ratio have a more significant effect on solids flux. On increasing the P/S air ratio, more air is introduced at the base of the riser. This increased primary zone gas velocity helps to move the solids up the riser and out the exit. Therefore, a higher circulation rate is required in order to maintain a given riser solids loading. Increasing the total gas velocity is seen to have a similar effect.

**Table B.1: Results of Circulation Rate Experiments**

P/S Air Ratio	Gas Velocity (m/s)	Solids Loading (kg/m <sup>3</sup> )	Net Solids Flux (kg/m <sup>2</sup> ·s)
2	8	90	19.6
2	8	120	21.7
2	8	150	22.6
3	8	90	27.7
3	8	120	31.0
3	8	150	33.2
4	8	90	33.0
4	8	120	36.7
4	8	150	40.3
2	7.3	90	16.2
2	7.3	120	17.0
2	7.3	150	18.3
3	7.3	90	22.0
3	7.3	120	21.8
3	7.3	150	24.7
4	7.3	90	27.7
4	7.3	120	28.0
4	7.3	150	28.8
2	6.7	90	15.7
2	6.7	120	15.2
2	6.7	150	14.9
3	6.7	90	19.9
3	6.7	120	20.8
3	6.7	150	20.4
4	6.7	90	23.1
4	6.7	120	23.5
4	6.7	150	25.3

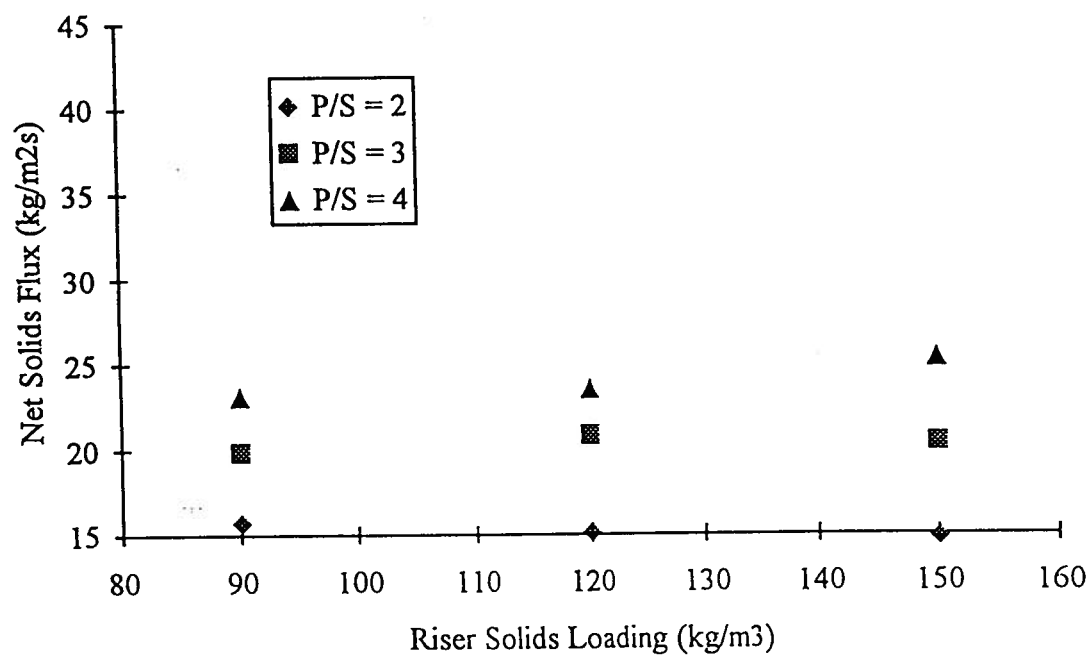


**Figure B.2: Effect of Riser Solids Loading on Net Solids Flux for a Gas Velocity of 8 m/s (Ottawa 50 sand, total inventory = 100 kg, combustor temperature = 850 °C)**

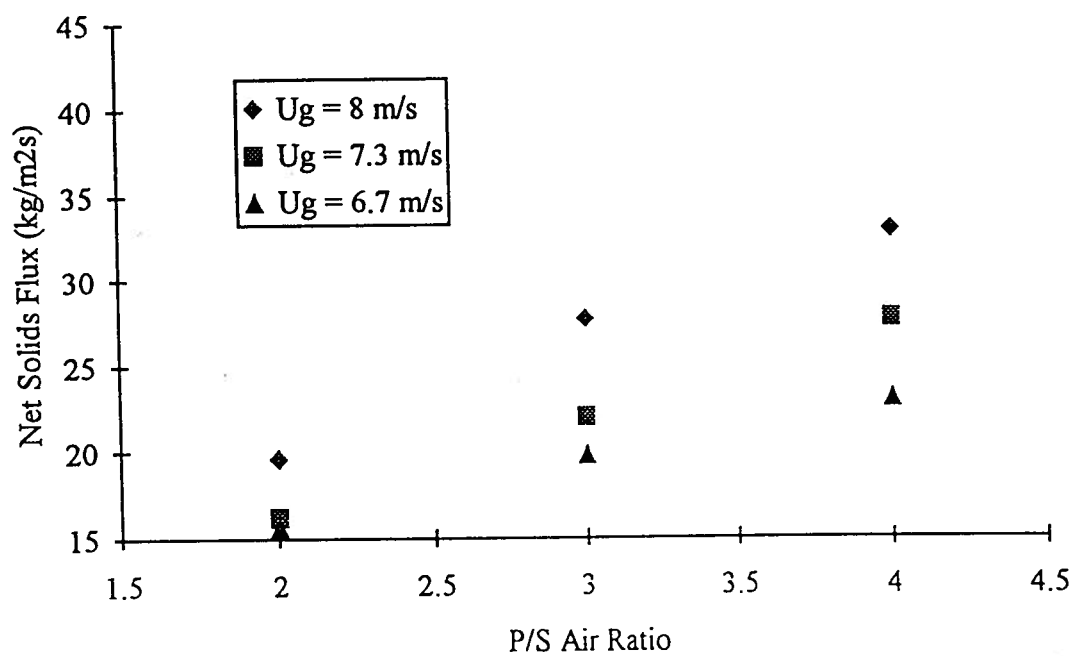


**Figure B.3: Effect of Riser Solids Loading on Net Solids Flux for a Gas Velocity of 7.3 m/s (Ottawa 50 sand, total inventory = 100 kg, combustor temperature = 850 °C)**

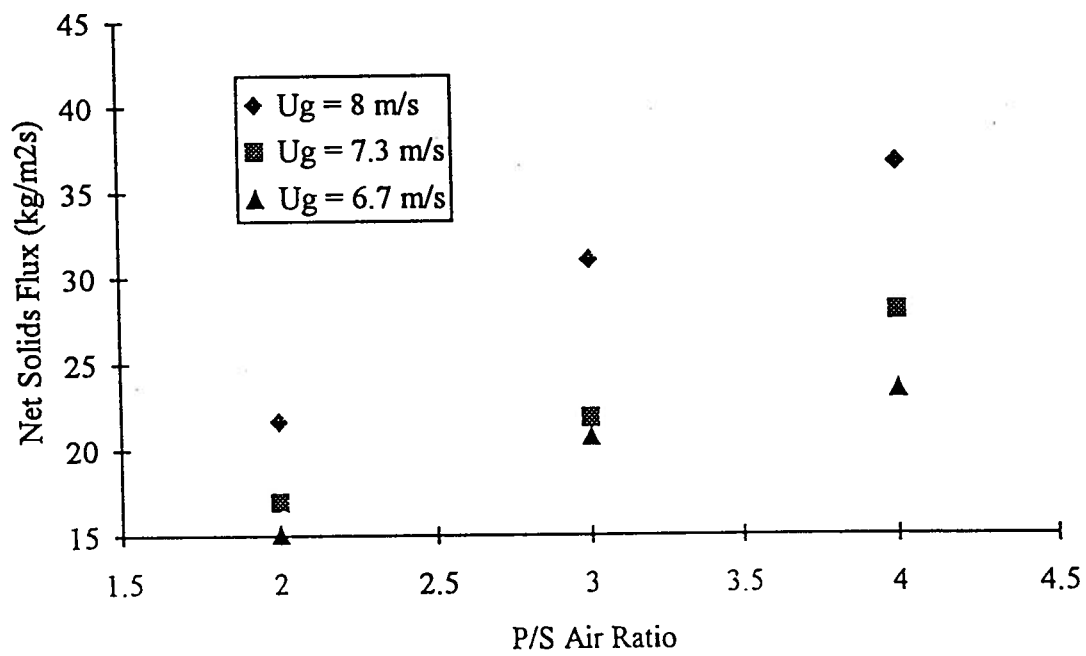




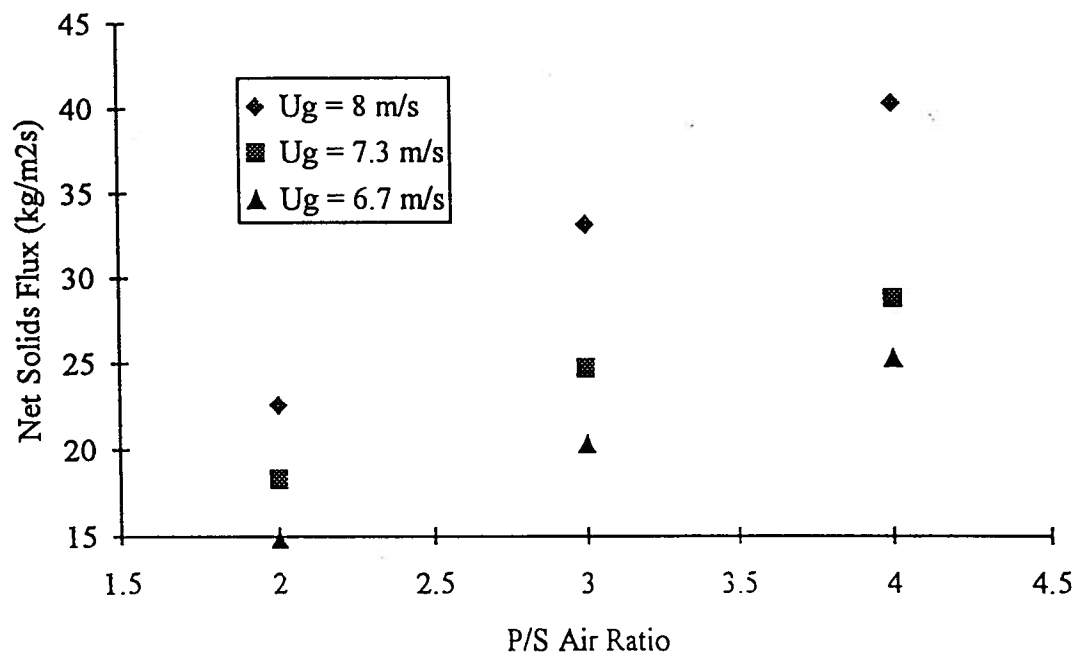
**Figure B.4: Effect of Riser Solids Loading on Net Solids Flux for a Gas Velocity of 6.7 m/s (Ottawa 50 sand, total inventory = 100 kg, combustor temperature = 850 °C)**



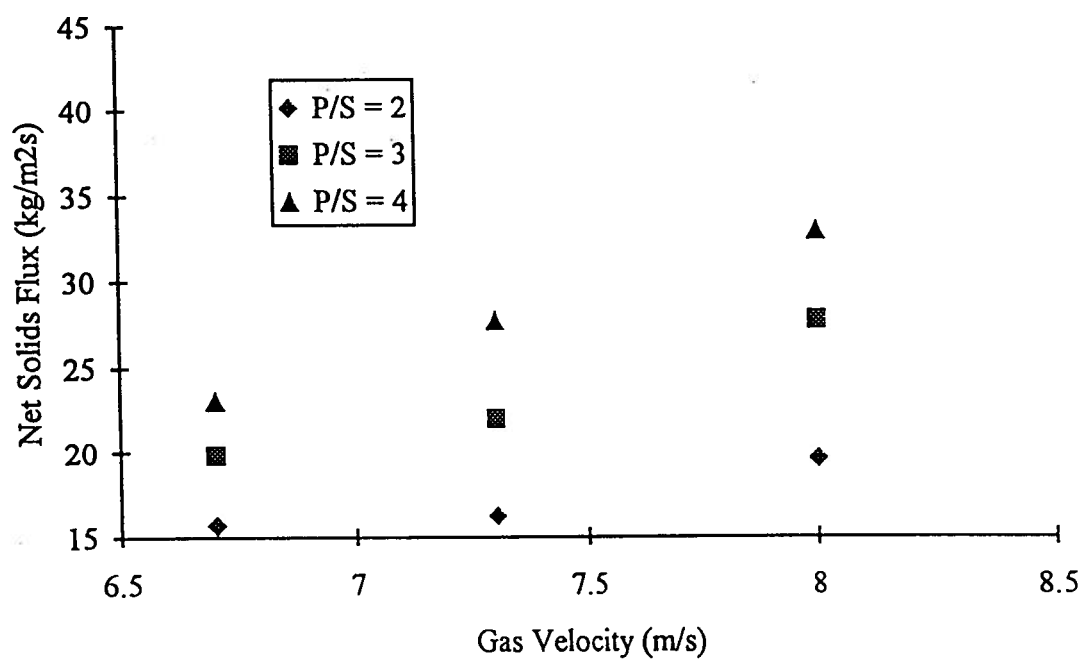
**Figure B.5: Effect of P/S Air Ratio on Net Solids Flux for a Riser Solids Loading of  $90 \text{ kg/m}^3$  (Ottawa 50 sand, total inventory = 100 kg, combustor temperature =  $850^\circ \text{C}$ )**



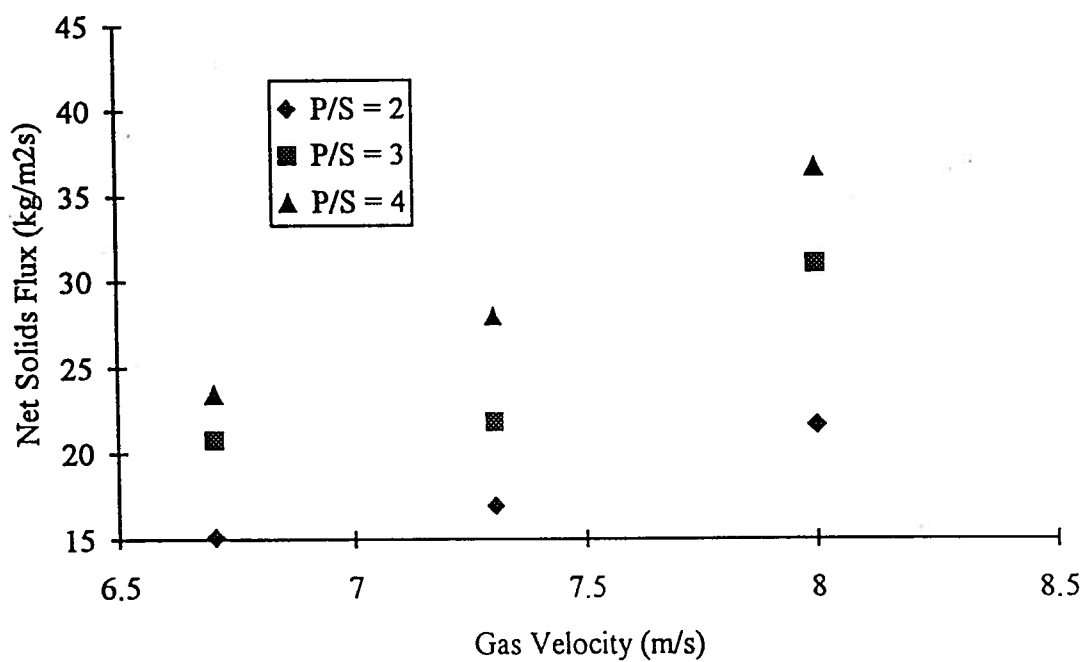
**Figure B.6: Effect of P/S Air Ratio on Net Solids Flux for a Riser Solids Loading of  $120 \text{ kg/m}^3$  (Ottawa 50 sand, total inventory = 100 kg, combustor temperature =  $850^\circ\text{C}$ )**



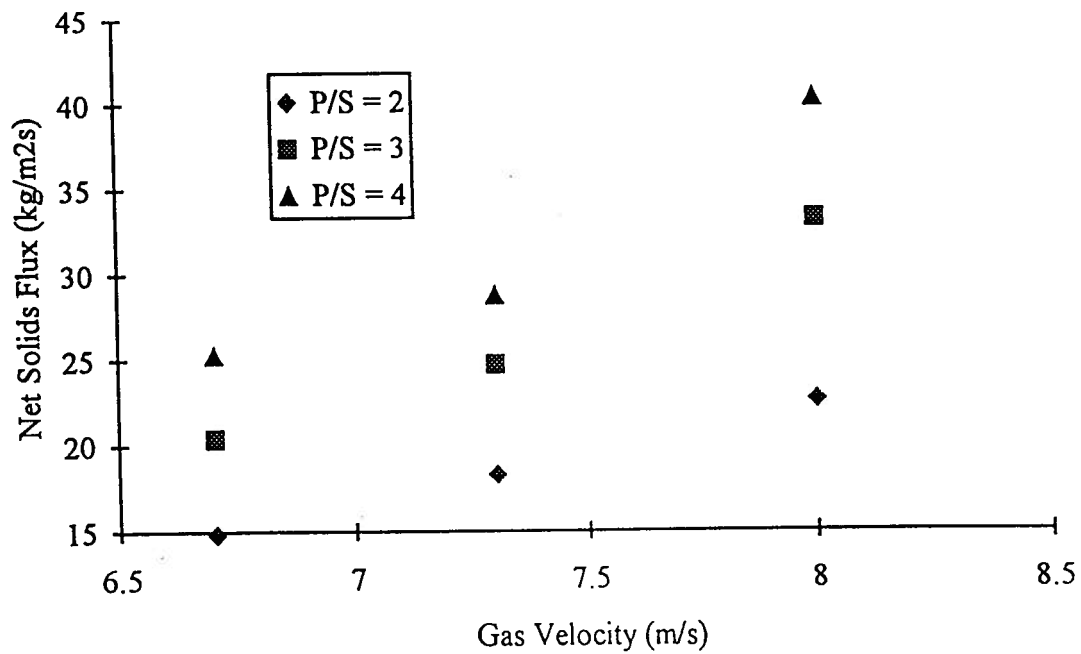
**Figure B.7: Effect of P/S Air Ratio on Net Solids Flux for a Riser Solids Loading of  $150 \text{ kg/m}^3$  (Ottawa 50 sand, total inventory = 100 kg, combustor temperature =  $850^\circ\text{C}$ )**



**Figure B.8: Effect of Gas Velocity on Net Solids Flux for a Riser Solids Loading of 90 kg/m<sup>3</sup>(Ottawa 50 sand, total inventory = 100 kg, combustor temperature = 850 °C)**



**Figure B.9: Effect of Gas Velocity on Net Solids Flux for a Riser Solids Loading of 120 kg/m<sup>3</sup> (Ottawa 50 sand, total inventory = 100 kg, combustor temperature = 850 °C)**



**Figure B.10: Effect of Gas Velocity on Net Solids Flux for a Riser Solids Loading of 150 kg/m<sup>3</sup> (Ottawa 50 sand, total inventory = 100 kg, combustor temperature = 850 °C)**

## **APPENDIX C - PROGRAM LISTING - DATA LOGGING AND CONTROL FOR THE UBC PILOT CFB COMBUSTOR**

This appendix provides the listing of the QuickBasic program written for data logging and control of the UBC pilot CFB combustor. It consists of two programs - T1 performs the data logging operations and T2 is the GPC control algorithm for combustion temperature control. Information is passed between the two through the COMMON variable storage block.



# COMMON.BI

defines variables to be saved in ram when loading and unloading T1 and T2

```

DIM a(1 TO 3, 1 TO 2)
DIM b(0 TO 4, 1 TO 2, 1 TO 2)
DIM cc(1 TO 4)
DIM info%(15)
DIM k(1 TO 10)
DIM lam(1 TO 20, 1 TO 2)
DIM ma(5)
DIM modifiers%(1)
DIM offset%(3, 15)
DIM positions%(15)
DIM p(1 TO 10, 1 TO 10, 1 TO 2)
DIM pt(1 TO 10, 1 TO 10)
DIM sp(5)
DIM u(-5 TO 0, 1 TO 2)
DIM y(-5 TO 0, 1 TO 2)
DIM uf(-5 TO 0, 1)
DIM yf(-5 TO 0, 1)
DIM du(-5 TO 0, 1)
DIM dyf(-5 TO 0, 1)
DIM theta(1 TO 10)
DIM fi(1 TO 10)
DIM wrls(1 TO 10, 1 TO 30)
DIM alpha(1 TO 2)
DIM setpnt(1 TO 2)
DIM tube%(1 TO 3)
COMMON SHARED errors%, address%, command%, positions%(),
modifiers%(), info%()
COMMON SHARED adapt%, arat, b(), cont%, coalo, control%, elm1p, elm2p
COMMON SHARED elm1s, elm2s, elm1l, elm2l, fly%, gout%, k(), loado, limeo
COMMON SHARED lam(), ma(), nu%, p(), pt(), sp(), samtime, saver%, rango
COMMON SHARED rangco, range, rangch, rangno, rangso, risvel, tsamp%,
times%
COMMON SHARED wrls(), warnup%, y(), u()
COMMON SHARED theta(), fi(), ffr, load, templ, a()
COMMON SHARED change%, istme%, ppmco, tube%()

```

```

COMMON SHARED oxyg, alph(), setpnt()
COMMON SHARED oxy%, water%, oxysp, twosp, twof
COMMON SHARED kox, tauio, taudo, kl, tauil, taucl
COMMON SHARED kw, tauiw, tauclw, ewm1w, ewm2w
COMMON SHARED eom1o, eom2o, alphw, ma3high, ma3low, limit%
COMMON SHARED sphigh, splow, prbst%, prbs%, prbstint%, hopper%
COMMON SHARED cc(), uf(), yf(), duf(), dyf()
COMMON SHARED pidt%, k4, taui4, taucl4, e4, elm14, elm24
COMMON SHARED limitc%, ma4high, ma4low, ccount%, trace, lamrls

```

```

T1
    program to perform data logging operations

' $STATIC
' $INCLUDE: 'c:\opto\q\progs\common.bi'

' Declare common variables to be passed to optoware driver

' Declaration of subroutines
DECLARE SUB OPTOWARE (a%, b%, C%, BYVAL D%, BYVAL E%, BYVAL
F%)
DECLARE SUB fmain ()
DECLARE SUB getdata ()
DECLARE SUB saveit ()
DECLARE SUB numdisp ()
DECLARE SUB toolor ()
DECLARE SUB aircont ()
DECLARE SUB loadcell ()
DECLARE FUNCTION orifice! (dor, dpip, p1, p2)
DECLARE SUB powerup ()
DECLARE SUB hello ()
DECLARE SUB initialsp ()
DECLARE FUNCTION seconds! ()
DECLARE SUB opto ()
DECLARE SUB errmsg ()

' Dimension array variables
DIM value(3, 15)
DIM offer(3, 15)

' Definition of logged data variables

'VALUE(1,0) = L-2
'VALUE(1,1) = L-3 bottom temperature
'VALUE(1,2) = L-4
'VALUE(1,3) = L-6
'VALUE(1,7) = L-12 temperature just above inbed hx

' Assign data constants
offer(2, 13) = -.045177
offer(3, 6) = .015873
offer(2, 14) = .008547
offer(3, 7) = .006105
offset%(1, 0) = -30
offset%(1, 1) = -30
offset%(1, 2) = -30
offset%(1, 3) = -30
offset%(1, 4) = -25
offset%(1, 5) = -25
offset%(1, 6) = -20
offset%(1, 7) = -20
offset%(1, 8) = -20
offset%(1, 9) = -20
offset%(1, 10) = -9
offset%(1, 11) = -12
offset%(1, 12) = -11
' offset%(1, 13) = 0
offset%(1, 14) = 3
offset%(1, 15) = 4
' offset%(2, 0) = 0
offset%(2, 1) = 4
' offset%(2, 2) = 0
offset%(2, 3) = 1
' offset%(2, 4) = 0
' offset%(2, 5) = 0
offset%(2, 6) = -5
offset%(2, 7) = -6
offset%(2, 8) = -24
offset%(2, 9) = -12
offset%(2, 10) = -9
offset%(2, 11) = -3
offset%(2, 12) = -8
offset%(2, 13) = 23
offset%(2, 14) = 6
offset%(2, 15) = 2
' offset%(3, 0) = 0
' offset%(3, 1) = 0
offset%(3, 2) = 1
offset%(3, 3) = -59
offset%(3, 4) = 3
' offset%(3, 5) = 0

```

```

offset%(3, 6) = 99
offset%(3, 7) = 4
offset%(3, 8) = 13
'offset%(3, 9) = 0
'offset%(3, 10) = 0
'offset%(3, 11) = 0
'offset%(3, 12) = 0
'offset%(3, 13) = 0
'offset%(3, 14) = 0
'offset%(3, 15) = 0

' define optimum command numbers
setpnum% = 102
conport% = 104
pupclr% = 0
reset% = 1
conout% = 8
probt% = 76
setoff% = 53
rtmp% = 77
stavin% = 47
racb% = 48
ravin% = 49
rain% = 37
ranout% = 36
waout% = 35

' Set air properties
MU = .1812
RO1 = 1.2928
cp = 1.403
cv = 1

' set orifice run parameters
dorprim = .034925
dpprim = .0508
dorsec = .028575
dipsec = .0381

kp = 50
ks = 50

gam1 = .1
alphw = .5

' Set serial port number
command% = setpnum%
info%(0) = 2
CALL opto

' configure serial port
command% = conport%
info%(0) = 9600
CALL opto

IF times% = 1 THEN GOTO 145

kl = 1 / 50

taul = 20
taudl = .12
kw = .26
tauw = 30
taudw = .06
kox = 8
tauo = 150
taudo = .25
taui4 = 10000000000000#

rannum = 1
RANDOMIZE (rannum)
' initialize variables

nu% = 1
FOR i% = 0 TO 15
    positions%(i%) = 0
    info%(i%) = 0
NEXT i%

rango = 10
rangco = 1000

```

```

rangc = 20
rangch = .2
rangno = 500
rangso = 1000

tsamp% = 120
sp(4) = 4
sp(5) = 4

' Check if power up clear is required (ie. has hardware been shut off)
address% = 1
command% = rtemp%
positions%(0) = 0
positions%(1) = -1
CALL opto

IF errors% <> 0 THEN CALL powerup

' Initial entry screen and file initialization
CALL hello

6
' Determine set points for flow controllers
IF cont% = 1 THEN CALL initials

COLOR 7, 0

' Initialize display to number format
gout% = 1

' Read the load cells
CALL loadcell
' coalo = coal
' limeo = lime

TIME$ = "00:00:00"
samtime = seconds

145
' Set up function keys for main menu
GOSUB 7000

150
WHILE (seconds - samtime) < tsamp%
IF change% = 1 THEN
'
' arat / (arat + 1) * airset - 37
' sp(2) = 1 / (arat + 1) * airset
' sp(1) = 0
' sp(2) = 0
command% = waout%
address% = 3
positions%(1) = -1
positions%(0) = 14
info%(0) = INT((sp(4) - 4) / 16 * 4095)
CALL opto
change% = 0
END IF
COLOR 7, 0: LOCATE 22, 1: PRINT "ELAPSED TIME", TIME$
COLOR 7, 0: LOCATE 23, 1: PRINT "SAMPLE RATE (S).....", tsamp%
IF cont% = 1 THEN CALL aircont

'calculate riser velocity
veloc = 8.31434 * (value(1, 1) + 273) * (pair + sair + 37) / 28.82 / 3600 / (.1524) ^
2 / 100

IF gout% = 1 THEN
COLOR 14, 0
LOCATE 3, 10: PRINT USING "###.##", tairi
LOCATE 4, 16: PRINT USING "###.##", pair
LOCATE 13, 50: PRINT USING "###.##", ma(1)
LOCATE 5, 16: PRINT USING "###.##", sair
LOCATE 14, 50: PRINT USING "###.##", ma(2)
LOCATE 7, 16: PRINT USING "###.##", value(2, 10)
LOCATE 7, 21: PRINT sp(3)
LOCATE 15, 50: PRINT USING "###.##", ma(3)
LOCATE 2, 16: PRINT USING "###.##", value(3, 0)
LOCATE 3, 16: PRINT USING "###.##", airset
LOCATE 8, 16: PRINT USING "###.##", two
LOCATE 9, 16: PRINT USING "###.##", veloc
LOCATE 12, 16: PRINT USING "###.##", value(1, 1)
LOCATE 10, 16: PRINT USING "###.##", ma(4)
END IF

```



```

IF limit% = 1 THEN
    limit% = 0
ELSE
    limit% = 1
CLS
INPUT "high ma setting"; ma3high
INPUT "low ma setting"; ma3low
CALL numdisp
GOSUB 7300
END IF
RETURN

*****
'*****
*****
7010
'SUBROUTINE TO CHANGE FUNCTION KEYS FOR SETUP CONTROL
COLOR 10, 0
KEY 1, "SPLRAT"
KEY 2, "RANGE"
IF cont% = 0 THEN
    KEY 3, "CONTRL"
ELSE
    KEY 3, "NO CON"
END IF
KEY 4, "COAL"
KEY 5, "VEL"
KEY 6, "O2"
KEY 7, "TWO"
KEY 8, "LOAD"
KEY 9, "TUBES"
KEY 10, "RETRN"

KEY(1) ON: KEY(2) ON: KEY(3) ON: KEY(4) ON: KEY(5) ON
KEY(6) ON: KEY(7) ON: KEY(8) ON: KEY(9) ON: KEY(10) ON

ON KEY(1) GOSUB 7020
ON KEY(2) GOSUB 7040
ON KEY(3) GOSUB 7100
ON KEY(4) GOSUB 8500
ON KEY(5) GOSUB 7110

*****
*****
7013
' subroutine to work on oxygen controller
COLOR 10, 0
IF oxy% = 0 THEN
    KEY 1, "CONTRL"
ELSE
    KEY 1, "NO CON"
END IF
KEY 2, ""
KEY 3, "SP"

KEY 4, ""
KEY 5, "K"
KEY 6, "TAUT"
KEY 7, "TAUD"
KEY 8, ""
KEY 9, ""
KEY 10, "RETRN"

KEY(1) ON: KEY(2) ON: KEY(3) ON: KEY(4) ON: KEY(5) ON
KEY(6) ON: KEY(7) ON: KEY(8) ON: KEY(9) ON: KEY(10) ON

ON KEY(1) GOSUB 7014
ON KEY(2) GOSUB 9990
ON KEY(3) GOSUB 7015
ON KEY(4) GOSUB 9990

ON KEY(6) GOSUB 7013
ON KEY(7) GOSUB 7093
ON KEY(8) GOSUB 7032
ON KEY(9) GOSUB 7099
ON KEY(10) GOSUB 7000

KEY ON
COLOR 7, 0

RETURN
'*****
*****
*****

```

```

ON KEY(5) GOSUB 7016
ON KEY(6) GOSUB 7017
ON KEY(7) GOSUB 7018
ON KEY(8) GOSUB 9990
ON KEY(9) GOSUB 9990
ON KEY(10) GOSUB 7010

```

```

KEY ON
COLOR 7, 0

```

```

RETURN

```

```

7014
IF oxy% = 1 THEN
  oxy% = 0
ELSE
  oxy% = 1
END IF
GOSUB 7013
RETURN

```

```

7015
CLS
INPUT "input new oxygen setpoint (%)", oxy%
CALL numdisp
GOSUB 7300
RETURN

```

```

7016
CLS
INPUT "input new oxygen gain", kox
CALL numdisp
GOSUB 7300
RETURN

```

```

7017
CLS
INPUT "input new oxygen tau", tauio
CALL numdisp
GOSUB 7300
RETURN

```

```

7018
CLS
INPUT "input new oxygen tau", tauo
CALL numdisp
GOSUB 7300
RETURN

```

```

*****
*****

```

```

7032
' subroutine to work on load controller
COLOR 10, 0
KEY 1, ""
KEY 2, ""
KEY 3, ""
KEY 4, ""
KEY 5, "K"
KEY 6, "TAU"
KEY 7, "TAUD"
KEY 8, ""
KEY 9, ""
KEY 10, "RETRN"

```

```

KEY(1) ON: KEY(2) ON: KEY(3) ON: KEY(4) ON: KEY(5) ON
KEY(6) ON: KEY(7) ON: KEY(8) ON: KEY(9) ON: KEY(10) ON

```

```

ON KEY(1) GOSUB 9990
ON KEY(2) GOSUB 9990
ON KEY(3) GOSUB 9990
ON KEY(4) GOSUB 9990
ON KEY(5) GOSUB 7033
ON KEY(6) GOSUB 7034
ON KEY(7) GOSUB 7035
ON KEY(8) GOSUB 9990
ON KEY(9) GOSUB 9990
ON KEY(10) GOSUB 7010

```

```

KEY ON
COLOR 7, 0

```

```

RETURN

7033
CLS
INPUT "input new load gain", kl
CALL numdisp
GOSUB 7300
RETURN

7034
CLS
INPUT "input new load tau", tauil
CALL numdisp
GOSUB 7300
RETURN

7035
CLS
INPUT "input new load taud", tauld
CALL numdisp
GOSUB 7300
RETURN

*****
*****
*****
*****
7093
' subroutine to work on water controller
COLOR 10, 0
IF water% = 0 THEN
    KEY 1, "CONTRL"
ELSE
    KEY 1, "NO CON"
END IF
KEY 2, ""
KEY 3, "SP"

KEY 4, ""
KEY 5, "K"
KEY 6, "TAUI"
KEY 7, "TAUD"
KEY 8, ""
KEY 9, "Filtr"
KEY 10, "RETRN"

KEY(1) ON: KEY(2) ON: KEY(3) ON: KEY(4) ON: KEY(5) ON
KEY(6) ON: KEY(7) ON: KEY(8) ON: KEY(9) ON: KEY(10) ON

ON KEY(1) GOSUB 7094
ON KEY(2) GOSUB 9990
ON KEY(3) GOSUB 7095
ON KEY(4) GOSUB 9990
ON KEY(5) GOSUB 7096
ON KEY(6) GOSUB 7097
ON KEY(7) GOSUB 7098
ON KEY(8) GOSUB 9990
ON KEY(9) GOSUB 7092
ON KEY(10) GOSUB 7010

KEY ON
COLOR 7, 0

RETURN

7094
IF water% = 1 THEN
    water% = 0
ELSE
    water% = 1
END IF
GOSUB 7093
RETURN

7095
CLS
INPUT "input new water setpoint (%)", twosp
CALL numdisp
GOSUB 7300

```



RETURN

7096

CLS

INPUT "input new water gain", kw

CALL numdisp

GOSUB 7300

RETURN

7097

CLS

INPUT "input new water tau", tauw

CALL numdisp

GOSUB 7300

RETURN

7098

CLS

INPUT "input new water tau", tauw

CALL numdisp

GOSUB 7300

RETURN

7092

CLS

INPUT "input new water filter factor (0 to 1)", alphw

CALL numdisp

GOSUB 7300

RETURN

\*\*\*\*\*  
\*\*\*\*\*

7099

'subroutine to toggle on/off for tubes

CLS

INPUT "which tube to toggle on/off", xx%

IF tube%(xx%) = 1 THEN

    tube%(xx%) = 0

ELSE

    tube%(xx%) = 1

END IF

CALL getdata  
CALL numdisp  
GOSUB 7010  
RETURN

REM\*\*\*\*\*  
\*\*\*\*\*

7020

' SUBROUTINE TO CHANGE SAMPLE RATE

COLOR 10, 0

KEY 1, "-30 S"

KEY 2, "-5 S"

KEY 3, "-15 S"

KEY 4, "-1 M"

KEY 5, "-2 M"

KEY 6, "-3 M"

KEY 7, "-5 M"

KEY 8, ""

KEY 9, "user"

KEY 10, "RETRN"

KEY(1) ON: KEY(2) ON: KEY(3) ON: KEY(4) ON: KEY(5) ON  
KEY(6) ON: KEY(7) ON: KEY(8) ON: KEY(9) ON: KEY(10) ON

ON KEY(1) GOSUB 7025

ON KEY(2) GOSUB 7026

ON KEY(3) GOSUB 7027

ON KEY(4) GOSUB 7028

ON KEY(5) GOSUB 7029

ON KEY(6) GOSUB 7030

ON KEY(7) GOSUB 7031

ON KEY(8) GOSUB 9990

ON KEY(9) GOSUB 8033

ON KEY(10) GOSUB 7010

KEY ON

COLOR 7, 0

RETURN

7025   tsamp% = 30

```

GOSUB 7010
RETURN

7026  tsamp% = 5
GOSUB 7010
RETURN

7027  tsamp% = 15
GOSUB 7010
RETURN

7028  tsamp% = 60
GOSUB 7010
RETURN

7029  tsamp% = 120
GOSUB 7010
RETURN

7030  tsamp% = 180
GOSUB 7010
RETURN

7031  tsamp% = 300
GOSUB 7010
RETURN

8033  CLS
      INPUT "input sample time in seconds"; tsamp%
      IF tsamp% < 10 THEN GOTO 8033
      CLS
      CALL numdisp
      GOSUB 7300
      GOSUB 7010
      RETURN

REM*****
*****

7040  ' SUBROUTINE TO CHANGE ANALYSER BANK RANGES

      COLOR 10, 0
      KEY 1, "O2"
      KEY 2, "CH4"
      KEY 3, "NOX"
      KEY 4, "SO2"
      KEY 5, ""
      KEY 6, ""
      KEY 7, ""
      KEY 8, ""
      KEY 9, ""
      KEY 10, "RETRN"

      KEY(1) ON: KEY(2) ON: KEY(3) ON: KEY(4) ON: KEY(5) ON
      KEY(6) ON: KEY(7) ON: KEY(8) ON: KEY(9) ON: KEY(10) ON

      ON KEY(1) GOSUB 7045
      ON KEY(2) GOSUB 7050
      ON KEY(3) GOSUB 7055
      ON KEY(4) GOSUB 7060
      ON KEY(5) GOSUB 9990
      ON KEY(6) GOSUB 9990
      ON KEY(7) GOSUB 9990
      ON KEY(8) GOSUB 9990
      ON KEY(9) GOSUB 9990
      ON KEY(10) GOSUB 7010

      KEY ON
      COLOR 7, 0
      RETURN

REM*****
*****

REM*****
*****

```

```

7045      ' SUBROUTINE TO CHANGE O2 ANALYSER RANGE
          COLOR 10, 0
          KEY 1, "-5%"
          KEY 2, "-10%"
          KEY 3, "-25%"
          KEY 4, ""
          KEY 5, ""
          KEY 6, ""
          KEY 7, ""
          KEY 8, ""
          KEY 9, ""
          KEY 10, "RETRN"

          KEY(1) ON: KEY(2) ON: KEY(3) ON: KEY(4) ON: KEY(5) ON
          KEY(6) ON: KEY(7) ON: KEY(8) ON: KEY(9) ON: KEY(10) ON

          ON KEY(1) GOSUB 7046
          ON KEY(2) GOSUB 7047
          ON KEY(3) GOSUB 7048
          ON KEY(4) GOSUB 9990
          ON KEY(5) GOSUB 9990
          ON KEY(6) GOSUB 9990
          ON KEY(7) GOSUB 9990
          ON KEY(8) GOSUB 9990
          ON KEY(9) GOSUB 9990
          ON KEY(10) GOSUB 7040

          KEY ON
          COLOR 7, 0
          RETURN

7046      ' CHANGE O2 TO 0 TO 10%
          rango = 5
          GOSUB 7040
          RETURN

7047      ' CHANGE O2 TO 0 TO 10%

7048      ' CHANGE O2 TO 25%
          rango = 25
          GOSUB 7040
          RETURN

          REM*****
          REM*****
          REM*****

7050      ' SUBROUTINE TO CHANGE CH4 ANALYSER RANGE
          COLOR 10, 0
          KEY 1, "-0.2%"
          KEY 2, "-0.5%"
          KEY 3, ""
          KEY 4, ""
          KEY 5, ""
          KEY 6, ""
          KEY 7, ""
          KEY 8, ""
          KEY 9, ""
          KEY 10, "RETRN"

          KEY(1) ON: KEY(2) ON: KEY(3) ON: KEY(4) ON: KEY(5) ON
          KEY(6) ON: KEY(7) ON: KEY(8) ON: KEY(9) ON: KEY(10) ON

          ON KEY(1) GOSUB 7052
          ON KEY(2) GOSUB 7053
          ON KEY(3) GOSUB 9990
          ON KEY(4) GOSUB 9990
          ON KEY(5) GOSUB 9990
          ON KEY(6) GOSUB 9990
          ON KEY(7) GOSUB 9990

```

```

ON KEY(8) GOSUB 9990
ON KEY(9) GOSUB 9990
ON KEY(10) GOSUB 7040

KEY ON
COLOR 7,0
RETURN

7052
' CHANGE CH4 ANALYSER TO 0 TO 0.2%
RANGECH = .2
GOSUB 7040
RETURN

7053
' CHANGE CH4 ANALYSER TO 0 TO 0.5%
rangch = .5
GOSUB 7040
RETURN

REM*****
*****

REM*****
*****

7055
' SUBROUTINE TO CHANGE NOX ANALYSER RANGE
COLOR 10,0
KEY 1, "-2.5"
KEY 2, "-5"
KEY 3, "-10"
KEY 4, "-25"
KEY 5, "-50"
KEY 6, "-100"
KEY 7, "-250"
KEY 8, "-500"
KEY 9, ""
KEY 10, "RETRN"

KEY(1) ON: KEY(2) ON: KEY(3) ON: KEY(4) ON: KEY(5) ON
KEY(6) ON: KEY(7) ON: KEY(8) ON: KEY(9) ON: KEY(10) ON

ON KEY(1) GOSUB 7076
ON KEY(2) GOSUB 7077
ON KEY(3) GOSUB 7078
ON KEY(4) GOSUB 7079
ON KEY(5) GOSUB 7081
ON KEY(6) GOSUB 7082
ON KEY(7) GOSUB 7083
ON KEY(8) GOSUB 7084
ON KEY(9) GOSUB 9990
ON KEY(10) GOSUB 7040
COLOR 7,0
KEY ON

RETURN

7076
' CHANGE NOX ANALYSER TO 0 TO 2.5 PPM
rangno = 2.5
GOSUB 7040
RETURN

7077
' CHANGE NOX ANALYSER TO 0 TO 5 PPM
rangno = 5
GOSUB 7040
RETURN

7078
' CHANGE NOX ANALYSER TO 0 TO 10 PPM
rangno = 10
GOSUB 7040
RETURN

7079
' CHANGE NOX ANALYSER TO 0 TO 25 PPM
rangno = 25
GOSUB 7040
RETURN

```

```

7081      KEY 8, ""
      KEY 9, ""
      KEY 10, "RETRN"

      KEY(1) ON: KEY(2) ON: KEY(3) ON: KEY(4) ON: KEY(5) ON
      KEY(6) ON: KEY(7) ON: KEY(8) ON: KEY(9) ON: KEY(10) ON

      ON KEY(1) GOSUB 7062
      ON KEY(2) GOSUB 7063
      ON KEY(3) GOSUB 7064
      ON KEY(4) GOSUB 9990
      ON KEY(5) GOSUB 9990
      ON KEY(6) GOSUB 9990
      ON KEY(7) GOSUB 9990
      ON KEY(8) GOSUB 9990
      ON KEY(9) GOSUB 9990
      ON KEY(10) GOSUB 7040

      KEY ON
      COLOR 7, 0
      RETURN

7062      ' CHANGE SO2 ANALYSER TO 0 TO 1000 PPM
      rangso = 1000
      GOSUB 7040
      RETURN

7063      ' CHANGE SO2 ANALYSER TO 0 TO 3000 PPM
      rangso = 3000
      GOSUB 7040
      RETURN

7064      ' CHANGE SO2 ANALYSER TO 0 TO 5000 PPM
      rangso = 5000
      GOSUB 7040
      RETURN

7082      ' CHANGE NOX ANALYSER TO 0 TO 100 PPM
      rangno = 100
      GOSUB 7040
      RETURN

7083      ' CHANGE NOX ANALYSER TO 0 TO 250 PPM
      rangno = 250
      GOSUB 7040
      RETURN

7084      ' CHANGE NOX ANALYSER TO 0 TO 500 PPM
      rangno = 500
      GOSUB 7040
      RETURN

      REM*****
      *****

      REM*****
      *****

7060      ' SUBROUTINE TO CHANGE SO2 ANALYSER RANGE
      COLOR 10, 0
      KEY 1, "-1000"
      KEY 2, "-3000"
      KEY 3, "-5000"
      KEY 4, ""
      KEY 5, ""
      KEY 6, ""
      KEY 7, ""

```



```

*****
REM*****
*****
REM*****
*****
7300
' SUBROUTINE FOR UNSCHEDULED SAMPLING OF OPTOMUX
  TMS$ = TIMES$
  CALL getdata
  RETURN
REM*****
*****
*****
7400
' SUBROUTINE TO DISPLAY FUNCTION KEYS FOR VALVE CHANGE
  COLOR 10,0
  KEY 1, "PRIM"
  KEY 2, "SEC"
  KEY 3, "L-VAL"
  KEY 4, "COAL"
  KEY 5, "LIME"
  KEY 6, ""
  KEY 7, ""
  KEY 8, ""
  KEY 9, ""
  KEY 10, "RETRN"
  KEY(1) ON: KEY(2) ON: KEY(3) ON: KEY(4) ON: KEY(5) ON
  KEY(6) ON: KEY(7) ON: KEY(8) ON: KEY(9) ON: KEY(10) ON
  ON KEY(1) GOSUB 7410
  ON KEY(2) GOSUB 7420
  ON KEY(3) GOSUB 7430
  ON KEY(4) GOSUB 7440
  ON KEY(5) GOSUB 7450
  ON KEY(6) GOSUB 9990
  ON KEY(7) GOSUB 9990
  ON KEY(8) GOSUB 9990
  ON KEY(9) GOSUB 9990
  ON KEY(10) GOSUB 7000
  KEY ON
  COLOR 7,0
  RETURN
REM*****
*****
REM*****
*****
7410
' SUBROUTINE TO CHANGE PRIMARY AIR FLOW
  CLS
  COLOR 10,0
  PRINT "ENTER NEW PRIMARY AIR FLOW (KG/HR)"
  INPUT sp(1)
  elm1p = 0
  elm2p = 0
  CLS
  WRITE #1, "CHANGE IN VALVE SETTING - PRIMARY AIR"
  WRITE #2, "CHANGE IN VALVE SETTING - PRIMARY AIR"
  WRITE #3, "CHANGE IN VALVE SETTING - PRIMARY AIR"
  WRITE #4, "CHANGE IN VALVE SETTING - PRIMARY AIR"
  WRITE #5, "CHANGE IN VALVE SETTING - PRIMARY AIR"
  COLOR 7,0
  CALL numdisp
  GOSUB 7400
  GOSUB 7300
  RETURN
REM*****
*****
REM*****
*****
7420

```

```

' SUBROUTINE TO CHANGE SECONDARY AIR FLOW

CLS
COLOR 10,0
PRINT "ENTER NEW SECONDARY AIR FLOW (KG/HR)"
INPUT sp(2)
elm1s = 0
elm2s = 0
CLS
' WRITE #1, "CHANGE IN VALVE SETTING - SECONDARY AIR"
' WRITE #2, "CHANGE IN VALVE SETTING - SECONDARY AIR"
' WRITE #3, "CHANGE IN VALVE SETTING - SECONDARY AIR"
' WRITE #4, "CHANGE IN VALVE SETTING - SECONDARY AIR"
' WRITE #5, "CHANGE IN VALVE SETTING - SECONDARY AIR"
COLOR 7,0
CALL numdisp
GOSUB 7400
GOSUB 7300
RETURN

' SUBROUTINE TO CHANGE COAL (LARGE) HOPPER FLOW
7440
CLS
COLOR 10,0
PRINT "ENTER NEW COAL SETTING (MA)"
INPUT sp(4)
CLS
' WRITE #1, "CHANGE IN VALVE SETTING - LARGE HOPPER"
' WRITE #2, "CHANGE IN VALVE SETTING - LARGE HOPPER"
' WRITE #3, "CHANGE IN VALVE SETTING - LARGE HOPPER"
' WRITE #4, "CHANGE IN VALVE SETTING - LARGE HOPPER"
' WRITE #5, "CHANGE IN VALVE SETTING - LARGE HOPPER"
COLOR 7,0
CALL numdisp
GOSUB 7400
command% = waout%
address% = 3
positions%(0) = 14
positions%(1) = -1
info%(0) = INT((sp(4) - 4) / 16 * 4095)
CALL opto
GOSUB 7300
RETURN

' SUBROUTINE TO CHANGE L-VALVE AIR FLOW
7430
CLS
COLOR 10,0
PRINT "ENTER NEW SOLIDS LOADING (KG/M3)"
INPUT sp(3)
elm1l = 0
elm2l = 0
CLS
' WRITE #1, "CHANGE IN VALVE SETTING - L-VALVE"
' WRITE #2, "CHANGE IN VALVE SETTING - L-VALVE"
' WRITE #3, "CHANGE IN VALVE SETTING - L-VALVE"
' WRITE #4, "CHANGE IN VALVE SETTING - L-VALVE"
' WRITE #5, "CHANGE IN VALVE SETTING - L-VALVE"

REM*****
*****
REM*****
*****

```



```

REM*****
*****
7450
' SUBROUTINE TO CHANGE LIMESTONE (SMALL) HOPPER FLOW

CLS
COLOR 10,0
PRINT "ENTER NEW LIMESTONE SETTING (MA)"
INPUT sp(5)
CLS
' WRITE #1, "CHANGE IN VALVE SETTING - SMALL HOPPER"
' WRITE #2, "CHANGE IN VALVE SETTING - SMALL HOPPER"
' WRITE #3, "CHANGE IN VALVE SETTING - SMALL HOPPER"
' WRITE #4, "CHANGE IN VALVE SETTING - SMALL HOPPER"
' WRITE #5, "CHANGE IN VALVE SETTING - SMALL HOPPER"
COLOR 7,0
CALL numdisp
GOSUB 7400
command% = waout%
address% = 3
positions%(0) = 15
positions%(1) = -1
info%(0) = INT((sp(5) - 4) / 16 * 4095)
CALL opto
GOSUB 7300
RETURN

REM*****
*****
7500
KEY 1, "Horiz"
KEY 2, "Lam"
KEY 3, "Adapt"
KEY 4, "P-Mat"
KEY 5, "Contrl"
KEY 6, ""
KEY 7, "Temp"

```

```

KEY 8, "O2"
KEY 9, "Coefs"
KEY 10, "Retm"

KEY(1) ON: KEY(2) ON: KEY(3) ON: KEY(4) ON: KEY(5) ON
KEY(6) ON: KEY(7) ON: KEY(8) ON: KEY(9) ON: KEY(10) ON

ON KEY(1) GOSUB 7510
ON KEY(2) GOSUB 7550
ON KEY(3) GOSUB 7520
ON KEY(4) GOSUB 7521
ON KEY(5) GOSUB 7530
ON KEY(6) GOSUB 9990
ON KEY(7) GOSUB 7540
ON KEY(8) GOSUB 7545
ON KEY(9) GOSUB 7547
ON KEY(10) GOSUB 7000

KEY ON
COLOR 7,0
RETURN

*****
*****
7510
CLS
INPUT "Input new control horizon "; nu%
CLS
CALL numdisp
GOSUB 7500
RETURN

*****
*****
7520
IF adapt% = 0 THEN
    adapt% = 1
ELSE

```

```

7540
CLS
LOCATE 1, 1: INPUT "Input new temperature setpoint"; setpnt(1)
setpnt(2) = setpnt(1)
CLS
CALL numdisp
GOSUB 7500
RETURN

7545
CLS
LOCATE 1, 1: INPUT "Input new oxygen setpoint"; setpnt(2)
CLS
CALL numdisp
GOSUB 7500
RETURN

7547
CLS
INPUT "Input 'a' for a, 'b' for b, or 'c' for c"; co$
IF co$ = "A" THEN
    INPUT "which a? (1 to 3)", cco%
    IF cco% > 3 OR cco% < 1 THEN GOTO 7547
    INPUT "a coefficient ?", a(cco%, 1)
    ELSEIF co$ = "a" THEN
        INPUT "which a? (1 to 3)", cco%
        IF cco% > 3 OR cco% < 1 THEN GOTO 7547
        INPUT "a coefficient ?", a(cco%, 1)
    ELSEIF co$ = "B" THEN
        INPUT "which b? (0 to 3)", cco%
        IF cco% > 3 OR cco% < 0 THEN GOTO 7547
        INPUT "b coefficient ?", b(cco%, 1, 1)
    ELSEIF co$ = "b" THEN
        INPUT "which b? (0 to 3)", cco%
        IF cco% > 3 OR cco% < 0 THEN GOTO 7547
        INPUT "b coefficient ?", b(cco%, 1, 1)
    ELSEIF co$ = "C" THEN
        INPUT "which c? (1 or 2)", cco%
        IF cco% > 2 OR cco% < 1 THEN GOTO 7547
        INPUT "c coefficient ?", cc(cco%)
    ELSEIF co$ = "c" THEN

```

```

        adapt% = 0
    END IF
    GOSUB 7500
    RETURN

7521
CLS
PRINT "input new diagonal for p-matrix"
PRINT "-999 to not reset and return"
INPUT initial

IF initial = -999 THEN
    ' don't do anything
ELSE
    FOR lo% = 1 TO 2
        FOR i% = 1 TO 10
            FOR j% = 1 TO 10
                IF i% = j% THEN
                    p(i%, j%, lo%) = initial
                ELSE
                    p(i%, j%, lo%) = 0
                END IF
            NEXT j%
        NEXT i%
    NEXT lo%
END IF
CLS
CALL numdisp
GOSUB 7500
RETURN

7530
IF warmup% = 0 THEN
    warmup% = 1
ELSE
    warmup% = 0
END IF
GOSUB 7500
RETURN

```

```

INPUT "which c? (1 or 2)", cco%
IF cco% > 2 OR cco% < 1 THEN GOTO 7547
INPUT "c coefficient ?", cc(cco%)
ELSE
    GOTO 7547
END IF
CLS
CALL numdisp
GOSUB 7500
RETURN

7550
CLS
LOCATE 1, 1: INPUT "Which input", z%
LOCATE 2, 1: INPUT "new constraint ", conny
FOR i% = 1 TO 10
    lam(i%, z%) = conny
NEXT i%
CLS
CALL numdisp
GOSUB 7500
RETURN

*****
***
9990
RETURN
*****
*****
*****
8088
CLS
IF prbst% = 0 THEN
    PRINT "you have entered the setup routine for a PRBS test"
    PRINT "you can abort at any time by answering with -999"
    PRINT
8089
    PRINT "which hopper is being used for the fuel feed (4 or 5)"
    INPUT hopper%
    IF hopper% = -999 THEN RETURN

IF hopper% < 4 OR hopper% > 5 THEN GOTO 8089
PRINT
PRINT "input high ma signal of PRBS signal"
INPUT sphigh
IF sphigh = -999 THEN RETURN
PRINT
PRINT "input low ma signal of PRBS signal"
INPUT splow
IF splow = -999 THEN RETURN
PRINT
PRINT "what is the desired switch interval to sample interval ratio"
INPUT prbsint%
IF prbsint% = -999 THEN RETURN
PRINT
INPUT "press ENTER to confirm above information is correct (-999 to abort)";
xxxxx
IF xxxxx = -999 THEN RETURN
prbs% = 0
prbst% = 1
CLS
ELSE
    CLS
INPUT "press ENTER to confirm end of PRBS test (-999 to continue test)"; xxxxx
IF xxxxx = -999 THEN RETURN
prbst% = 0
CLS
END IF
RETURN
*****
*****
8500
' KEY TEMPLATE for pid coal controller
IF pidt% = 0 THEN
    KEY 1, "CONTRL"
ELSE
    KEY 1, "NO CON"
END IF

```

```

KEY 2, ""
KEY 3, "K"
KEY 4, "TAUT"
KEY 5, "TAUD"
KEY 6, ""
KEY 7, "LIMIT"
KEY 8, ""
KEY 9, ""
KEY 10, "RETRN"

KEY(1) ON: KEY(2) ON: KEY(3) ON: KEY(4) ON: KEY(5) ON
KEY(6) ON: KEY(7) ON: KEY(8) ON: KEY(9) ON: KEY(10) ON

ON KEY(1) GOSUB 8510
ON KEY(2) GOSUB 9990
ON KEY(3) GOSUB 8520
ON KEY(4) GOSUB 8530
ON KEY(5) GOSUB 8540
ON KEY(6) GOSUB 9990
ON KEY(7) GOSUB 8550
ON KEY(8) GOSUB 9990
ON KEY(9) GOSUB 9990
ON KEY(10) GOSUB 7010

KEY ON
COLOR 7, 0

RETURN

8510
IF pidt% = 1 THEN
    pidt% = 0
ELSE
    pidt% = 1
END IF

RETURN

8520
CLS

```

```

INPUT "input new coal gain", k4
CALL numdisp
GOSUB 7300
RETURN

8530
CLS
INPUT "input new coal taut", taut4
CALL numdisp
GOSUB 7300
RETURN

8540
CLS
INPUT "input new coal taud", taud4
CALL numdisp
GOSUB 7300
RETURN

8550
IF limite% = 1 THEN
    limite% = 0
ELSE
    limite% = 1
CLS
INPUT "high ma setting", ma4high
INPUT "low ma setting", ma4low
CALL numdisp
GOSUB 7300
END IF
RETURN

!*****
*****
9998
KEY 1, "Retrn"
KEY 2, ""
KEY 3, ""
KEY 4, ""

```

```

KEY 5, ""
KEY 6, ""
KEY 7, ""
KEY 8, ""
KEY 9, ""
KEY 10, "Quit"

KEY(1) ON: KEY(2) ON: KEY(3) ON: KEY(4) ON: KEY(5) ON
KEY(6) ON: KEY(7) ON: KEY(8) ON: KEY(9) ON: KEY(10) ON

ON KEY(1) GOSUB 7000
ON KEY(2) GOSUB 9990
ON KEY(3) GOSUB 9990
ON KEY(4) GOSUB 9990
ON KEY(5) GOSUB 9990
ON KEY(6) GOSUB 9990
ON KEY(7) GOSUB 9990
ON KEY(8) GOSUB 9990
ON KEY(9) GOSUB 9990
ON KEY(10) GOSUB 9999

KEY ON
COLOR 7, 0
RETURN
*****
*****

*****
*****
9999
STOP

END
*****
*****

SUB aircraft STATIC
  SHARED sp(), stavin%, racb%, ravin%, value(), offer(), ranout%, waout%
  SHARED errors%, address%, command%, positions%(), modifiers%(), info%()
  SHARED dorprim, dpiprim, dorsec, dpipec, pair, sair
  SHARED ma(), tauip, tauis, tauil, taudp, tauds, tauil, kp, ks, kl, gaml

  SHARED kw, kox, tauiv, tauio, taudw, tauco, rango, ramin%, veloc, tairi
  SHARED arat, oxy%, water%, oxysp, twosp, two, airset, rtemp%, twof
  SHARED alphw, ma3high, ma3low, limit%
  SHARED pidt%, k4, taui4, taud4, e4, elm14, elm24, limite%, ma4high, ma4low

  command% = stavin%
  address% = 2
  positions%(0) = 10
  positions%(1) = -1
  info%(0) = 10
  CALL opto

  command% = stavin%
  address% = 2
  positions%(0) = 13
  positions%(1) = 14
  positions%(2) = -1
  info%(0) = 10
  CALL opto

  address% = 3
  positions%(0) = 6
  positions%(1) = 7
  positions%(2) = -1
  info%(0) = 10
  CALL opto

  ' Read current ma signals to control valves
  command% = ranout%
  address% = 3
  FOR i% = 0 TO 4
    positions%(i%) = i% + 11
  NEXT i%
  positions%(5) = -1
  CALL opto
  FOR i% = 11 TO 15
    ma(i% - 10) = (info%(i%) / 4095 * 16) + 4
  NEXT i%

  ' Check if board 2 is done

```

```

command% = racb%
address% = 2
11
CALL opto

IF info%(13) = 0 THEN GOTO 11
IF info%(14) = 0 THEN GOTO 11

' Read board 2
command% = ravin%
address% = 2
positions%(0) = 13
positions%(1) = 14
positions%(2) = -1
CALL opto
FOR i% = 13 TO 14
value(address%, i%) = (info%(i%) / 4095 * 10)
NEXT i%

' Check if board 3 is done
command% = racb%
address% = 3
12
CALL opto

IF info%(6) = 0 THEN GOTO 12
IF info%(7) = 0 THEN GOTO 12

' Read board 3
command% = ravin%
address% = 3
positions%(0) = 6
positions%(1) = 7
positions%(2) = -1
CALL opto
FOR i% = 6 TO 7
value(address%, i%) = (info%(i%) / 4095 * 10)
NEXT i%

' Offsets for primary air flow
value(2, 13) = value(2, 13) + offer(2, 13)
value(3, 6) = value(3, 6) + offer(3, 6)
value(2, 14) = value(2, 14) + offer(2, 14)
value(3, 7) = value(3, 7) + offer(3, 7)

' Ensure it is o.k. to calculate without non-negative flows
IF value(3, 6) < 1 THEN value(3, 6) = 1.00001
IF value(2, 13) < 1 THEN value(2, 13) = 1
IF value(3, 7) < 1 THEN value(3, 7) = 1.00001
IF value(2, 14) < 1 THEN value(2, 14) = 1

' Calculate primary air flow
p1 = (value(3, 6) - 1) * 6894.733 + 101325
p2 = p1 - (value(2, 13) - 1) * 498.348
dor = dorprim
dpip = dpipprim
pair = orifice(dor, dpip, p1, p2) * 3600

' Calculate secondary air flow
p1 = (value(3, 7) - 1) * 6894.733 + 101325
p2 = p1 - (value(2, 14) - 1) * 498.348
dor = dorsec
dpip = dpipsec
sair = orifice(dor, dpip, p1, p2) * 3600

' Check if done with pressure drop measurement
' Check if board 2 is done

command% = racb%
address% = 2
18
CALL opto

IF info%(10) = 0 THEN GOTO 18

' Read board 2
command% = ravin%
address% = 2
positions%(0) = 10

```

```

address% = 3
positions%(0) = 0
positions%(1) = -1
CALL opto
' the -.025 is an offset for the portable o2 analyser
value(address%, 0) = (info%(0) / 4095 - .025) * rango

IF water% = 1 THEN
  IF value(2, 10) >= 200 THEN
    el = 170 - value(2, 10)
    ma(3) = ma(3) + kl * (1 + 1 / tauil + tauil) * el - kl * (1 + 2 * tauil) * elm11
    + kl * tauil * elm21
  ELSEIF value(2, 10) > 170 AND value(2, 10) < 200 THEN
    IF two > twosp THEN
      ew = twosp - two
      ma(3) = ma(3) + kw * (1 + 1 / tauiw + tauiw) * ew - kw * (1 + 2 * tauiw)
      * ewm1w + kw * tauiw * ewm2w
    ELSE
      el = 170 - value(2, 10)
      ma(3) = ma(3) + kl * (1 + 1 / tauil + tauil) * el - kl * (1 + 2 * tauil) *
      elm11 + kl * tauil * elm21
    END IF
  ELSEIF value(2, 10) < 50 THEN
    IF two < twosp THEN
      ew = twosp - two
      ma(3) = ma(3) + kw * (1 + 1 / tauiw + tauiw) * ew - kw * (1 + 2 * tauiw)
      * ewm1w + kw * tauiw * ewm2w
    ELSE
      el = 50 - value(2, 10)
      ma(3) = ma(3) + kl * (1 + 1 / tauil + tauil) * el - kl * (1 + 2 * tauil) *
      elm11 + kl * tauil * elm21
    END IF
  ELSE
    ew = twosp - two
    ma(3) = ma(3) + kw * (1 + 1 / tauiw + tauiw) * ew - kw * (1 + 2 * tauiw) *
    ewm1w + kw * tauiw * ewm2w
  IF limit% = 1 THEN
    IF two < twosp THEN
      ma(3) = ma3high
    ELSE
      ma(3) = ma3low
    END IF
  END IF
END IF

```

```

positions%(1) = -1
CALL opto
value(address%, 10) = (info%(10) / 4095 * 10)

' Calculate average solids loading (kg/m3)
value(2, 10) = (value(2, 10) - 1) * 101.325
value(2, 10) = gam1 * value(2, 10) + (1 - gam1) * loado
loado = value(2, 10)

' read water temperatures
' Read thermocouples
command% = rtemp%
address% = 1
FOR i% = 0 TO 14
  positions%(i%) = i%
NEXT i%
positions%(15) = -1
CALL opto
FOR i% = 0 TO 14
  value(address%, i%) = info%(i%) / 16
NEXT i%

' calculate average outlet water temperature
two = 0
xx% = 0
FOR i% = 1 TO 3
  IF tube%(i%) = 1 THEN
    two = two + value(1, 10 + i%)
    xx% = xx% + 1
  END IF
NEXT i%
IF xx% = 0 THEN
  two = 0
ELSE
  two = two / xx%
END IF
two = alphw * twof + (1 - alphw) * two
twof = two

' read O2 analyser signal
command% = ranin%

```

```

    END IF
  END IF
  END IF
  ELSE
    el = sp(3) - value(2, 10)
    ma(3) = ma(3) + k1 * (1 + 1 / tauil + tauidl) * el - k1 * (1 + 2 * tauidl) * elm1l +
    k1 * tauidl * elm2l
    IF limit% = 1 THEN
      IF value(2, 10) < sp(3) THEN
        ma(3) = ma3high
      ELSE
        ma(3) = ma3low
      END IF
    END IF
  END IF
  END IF

  airset = sp(1) + sp(2) + 37
  IF fly% = 1 THEN
    IF oxy% = 1 THEN
      IF veloc < 6.6 THEN
        IF value(3, 0) < oxysp THEN
          eo = oxysp - value(3, 0)
          tairi = airset + kox * (1 + 1 / tauio + taudo) * eo - kox * (1 + 2 * taudo)
          * eom1o + kox * taudo * eom2o
          sp(1) = arat * tairi / (arat + 1) - 37
          sp(2) = tairi - sp(1) - 37
        ELSE
          airtot = 6.6 * 100 * 3600 * .1524 ^ 2 / 8.31434 / (value(1, 1) + 273)
          airtot = airtot - .15
          airtot = airtot * 28.84
          sp(1) = arat / (arat + 1) * airtot - 37
          sp(2) = 1 / (arat + 1) * airtot
        END IF
      ELSEIF veloc > 9.8 THEN
        IF value(3, 0) > oxysp THEN
          eo = oxysp - value(3, 0)
          tairi = airset + kox * (1 + 1 / tauio + taudo) * eo - kox * (1 + 2 * taudo)
          * eom1o + kox * taudo * eom2o
          sp(1) = arat * tairi / (arat + 1) - 37
          sp(2) = tairi - sp(1) - 37
        ELSE
          airtot = 9.8 * 100 * 3600 * .1524 ^ 2 / 8.31434 / (value(1, 1) + 273)
          airtot = airtot - .15
          airtot = airtot * 28.84
          sp(1) = arat / (arat + 1) * airtot - 37
          sp(2) = 1 / (arat + 1) * airtot
        END IF
      ELSE
        ma(1) = ma(1) + kp * (1 + 1 / tauip + taudp) * ep -
        kp * (1 + taudp) * elm1p + kp * taudp * elm2p
        ma(2) = ma(2) + ks * (1 + 1 / tauis + tausds) * es - ks * (1 + tausds) * elm1s + ks * tausds * elm2s
      END IF
    ELSE
      elm2p = elm1p
      elm1p = ep
      elm2s = elm1s
      elm1s = es
      elm2l = elm1l
    END IF
  END IF
  ELSE
    airtot = 9.8 * 100 * 3600 * .1524 ^ 2 / 8.31434 / (value(1, 1) + 273)
    airtot = airtot - .15
    airtot = airtot * 28.84
    sp(1) = arat / (arat + 1) * airtot - 37
    sp(2) = 1 / (arat + 1) * airtot
  END IF
  ELSE
    eo = oxysp - value(3, 0)
    tairi = airset + kox * (1 + 1 / tauio + taudo) * eo - kox * (1 + 2 * taudo) *
    eom1o + kox * taudo * eom2o
    sp(1) = arat * tairi / (arat + 1) - 37
    sp(2) = tairi - sp(1) - 37
  END IF
  ELSE
    airtot = risvel * 100 * 3600 * .1524 ^ 2 / 8.31434 / (value(1, 1) + 273)
    airtot = airtot - .15
    airtot = airtot * 28.84
    sp(1) = arat / (arat + 1) * airtot - 37
    sp(2) = 1 / (arat + 1) * airtot
  END IF
  END IF

  ep = sp(1) - pair
  es = sp(2) - sair

  ma(1) = ma(1) + (ep / kp)
  ma(2) = ma(2) + (es / ks)

  ' ma(1) = ma(1) + kp * (1 + 1 / tauip + taudp) * ep -
  ' kp * (1 + taudp) * elm1p + kp * taudp * elm2p
  ' ma(2) = ma(2) + ks * (1 + 1 / tauis + tausds) * es - ks * (1 + tausds) * elm1s + ks * tausds * elm2s

  elm2p = elm1p
  elm1p = ep
  elm2s = elm1s
  elm1s = es
  elm2l = elm1l

```



```

elm11 = el
ewm2w = ewm1w
ewm1w = ew
eom2o = eom1o
eom1o = eo

command% = waout%
address% = 3
positions%(1) = -1
FOR i% = 1 TO 3
    IF ma(i%) < 4 THEN ma(i%) = 4
    IF ma(i%) > 16 THEN ma(i%) = 16
    positions%(0) = i% + 10
    info%(0) = INT((ma(i%) - 4) / 16 * 4095)
    CALL opto
NEXT i%

IF pidt% = 1 THEN
    e4 = setpnt(1) - value(1, 1)
    sp(4) = sp(4) + k4 * (1 + 1 / tau14 + tau44) * e4 - k4 * (1 + 2 * tau44) * elm14 +
    k4 * tau44 * elm24
    elm24 = elm14
    elm14 = e4
    IF limitc% = 1 THEN
        IF value(1, 1) < setpnt(1) THEN
            sp(4) = ma4high
        ELSE
            sp(4) = ma4low
        END IF
    END IF

    command% = waout%
    address% = 3
    positions%(1) = -1
    i% = 4
    IF sp(i%) < 4 THEN sp(i%) = 4
    IF sp(i%) > 16 THEN sp(i%) = 16

positions%(0) = i% + 10
info%(0) = INT((sp(i%) - 4) / 16 * 4095)
CALL opto
END IF

END SUB

SUB errmsg
    ' subroutine to display error message if a problem with optomux
    SHARED errors%, command%
    IF errors% = -1 THEN GOTO 345 'indicates only power up clear is required (non-
    fatal)
    CLS
    PRINT "ERROR ", errors%, "FOR COMMAND ", command%
    STOP
345
END SUB

SUB getdata STATIC
    SHARED stavin%, rtemp%, rainin%, ranout%, racb%, ravin%, offer()
    SHARED dorprim, dpipprim, dorsec, dpipsec, sair, pair, lair, saver%
    SHARED elm1p, elm2p, elm1s, elm2s, elm1l, elm2l, value()
    SHARED sp(), waout%, arat, airtot, gout%, gam1, loado
    SHARED errors%, address%, command%, positions%(), modifiers%(), info%()
    SHARED coalrat, limerat, coalo, limeo, coal, lime, tsamp%, two, twof

    ' Start averaging pressure transducers
    address% = 2
    command% = stavin%
    FOR i% = 0 TO 15
        positions%(i%) = i%
    NEXT i%
    info%(0) = 10
    CALL opto

    ' Start averagin pressure transducers (10 samples)
    command% = stavin%
    address% = 1

```

```

positions%(0) = 15
positions%(1) = -1
info%(0) = 10
CALL opto

command% = stavin%
address% = 3
FOR i% = 0 TO 2
    positions%(i%) = i% + 6
NEXT i%
positions%(3) = -1
info%(0) = 10
CALL opto

' Read load cells
CALL loadcell

' Read thermocouples
command% = rtemp%
address% = 1
FOR i% = 0 TO 14
    positions%(i%) = i%
NEXT i%
positions%(15) = -1
CALL opto
FOR i% = 0 TO 14
    value(address%, i%) = info%(i%) / 16
NEXT i%

' Read gas analyser signals
command% = ranin%
address% = 3
FOR i% = 0 TO 5
    positions%(i%) = i%
NEXT i%
positions%(6) = -1
CALL opto
' the -.025 is an offset for the portable o2 analyser
value(address%, 0) = (info%(0) / 4095 - .025) * rango
value(address%, 1) = info%(1) / 4095 * 5 * rangco

value(address%, 2) = info%(2) / 4095 * 5 * range
value(address%, 3) = info%(3) / 4095 * 5 * rangeh
value(address%, 4) = info%(4) / 4095 * rangno
value(address%, 5) = info%(5) / 4095 * rangso

' Read the output signal to the control valves
command% = ranout%
address% = 3
FOR i% = 0 TO 4
    positions%(i%) = i% + 11
NEXT i%
positions%(5) = -1
CALL opto
FOR i% = 11 TO 15
    value(address%, i%) = (info%(i%) / 4095 * 16) + 4
NEXT i%

' Check if averaging is finished on board #1

command% = racb%
address% = 1
36
CALL opto
IF info%(15) = 0 THEN GOTO 36

' Read averaged input from board #1
command% = ravin%
address% = 1
positions%(0) = 15
positions%(1) = -1
CALL opto
value(address%, 15) = (info%(15) / 4095 * 10)

' Check if averaging is finished on board #2

command% = racb%
address% = 2
37
CALL opto

```

```

FOR i% = 0 TO 15
  IF info%(i%) = 0 THEN GOTO 37
NEXT i%

' Read averaged inputs from board #2
command% = ravin%
address% = 2
FOR i% = 0 TO 15
  positions%(i%) = i%
NEXT i%
CALL opto
FOR i% = 0 TO 15
  value(address%, i%) = (info%(i%) / 4095 * 10)
NEXT i%

' Check if averaging is finished on board #3
command% = racb%
address% = 3
38
CALL opto

FOR i% = 6 TO 8
  IF info%(i%) = 0 THEN GOTO 38
NEXT i%

' Read averaged inputs from board #3
command% = ravin%
address% = 3
FOR i% = 0 TO 2
  positions%(i%) = i% + 6
NEXT i%
positions%(3) = -1
CALL opto
FOR i% = 6 TO 8
  value(address%, i%) = (info%(i%) / 4095 * 10)
NEXT i%

' Offsets for primary air flow
value(2, 13) = value(2, 13) + offer(2, 13)
value(3, 6) = value(3, 6) + offer(3, 6)

value(2, 14) = value(2, 14) + offer(2, 14)
value(3, 7) = value(3, 7) + offer(3, 7)

LOCATE 21, 10: PRINT value(3, 6)
LOCATE 21, 20: PRINT value(2, 13)

' Calculate standpipe pressure (in kpa)
value(1, 15) = ((value(1, 15) - 1) - 2.5) * 101.325 / 14.696
FOR i% = 0 TO 5
  value(2, i%) = ((value(2, i%) - 1) - 2.5) * 101.325 / 14.696
NEXT i%

' Calculate riser solids loading
value(2, 6) = (value(2, 6) - 1) * 70.35 * 393.7 / 9 / 5
value(2, 7) = (value(2, 7) - 1) * 70.35 * 393.7 / 30 / 5
value(2, 8) = (value(2, 8) - 1) * 70.35 * 393.7 / 42 / 5
value(2, 9) = (value(2, 9) - 1) * 2 - 5 * 2.54 * 393.7 / 48
value(2, 10) = (value(2, 10) - 1) * 101.325
value(2, 11) = ((value(2, 11) - 1) * 2 - 5) * 2.54 * 393.7 / 48
value(2, 12) = ((value(2, 12) - 1) * 2 - 5) * 2.54 * 393.7 / 48
value(2, 10) = gam1 * value(2, 10) + (1 - gam1) * loado
loado = value(2, 10)

' Calculate mass flowrates of air
IF value(3, 6) <= 1 THEN value(3, 6) = 1.00001
IF value(2, 13) < 1 THEN value(2, 13) = 1
IF value(3, 7) <= 1 THEN value(3, 7) = 1.00001
IF value(2, 14) < 1 THEN value(2, 14) = 1
p1 = ((value(3, 6) - 1) * 101.325 / 14.696 * 1000) + 101325
p2 = p1 - (value(2, 13) - 1) * 2 * 2.54 / 100 * 9.81 * 998
dor = dorprim
dpip = dpipprim
pair = orifice(dor, dpip, p1, p2) * 3600

p1 = ((value(3, 7) - 1) * 101.325 / 14.696 * 1000) + 101325
p2 = p1 - (value(2, 14) - 1) * 2 * 2.54 / 100 * 9.81 * 998
dor = dorsec
dpip = dpipsec
sair = orifice(dor, dpip, p1, p2) * 3600

```

```

' calculate average outlet water temperature
two = 0
xx% = 0
FOR i% = 1 TO 3
  IF tube%(i%) = 1 THEN
    two = two + value(1, 10 + i%)
    xx% = xx% + 1
  END IF
NEXT i%
IF xx% = 0 THEN
  two = 0
ELSE
  two = two / xx%
END IF
two = twof * .9 + .1 * two
twof = two

'calculate riser velocity
veloc = 8.31434 * (value(1, 1) + 273) * (pair + sair + 37) / 28.82 / 3600 / (.1524) ^
2 / 100

' Check if should save data to disk
IF saver% = 1 THEN CALL saveit

' Check if controlling riser velocity

' Check how to display results to screen
IF gout% = 1 THEN CALL numdisp

templ = value(1, 1)
oxyg = value(3, 5)

' with the comments below the gpc controller is shut off permanently
IF itsrne% = 1 THEN

ccount% = ccount% + 1
IF ccount% = 5 THEN
  ccount% = 0
  PRINT "help"

```

```

coalrat = (coalo - coal) * 3600 / (tsamp% * 5)
PRINT "help"
limerat = (limeo - lime) * 3600 / tsamp%
PRINT "help"
coalo = coal
limeo = lime
END IF

change% = 1
itsrne% = 0
ppmco = value(3, 1)
CHAIN "c:\opto\qb\progs\2"
END IF

END SUB

SUB hello
  SHARED rtn$, cont%

' Print program header
CLS
COLOR 14, 0
PRINT "      Welcome to the CONTROL program"
PRINT "      for the CFB COMBUSTOR"
PRINT
PRINT
PRINT
PRINT
PRINT
PRINT
PRINT
PRINT "The optomux boards are all set to go"
PRINT
PRINT
PRINT
PRINT
PRINT
PRINT
PRINT "ARE YOU SURE YOU HAVE STORED THE PREVIOUSLY LOGGED
FILES???"
PRINT " "

```

```

PRINT " "
INPUT "type 'YES' to continue ", done$
IF done$ <> "YES" THEN STOP

' Open output files
OPEN "o", #1, "c:\opto\qb\progs\data\temp.out"
OPEN "o", #2, "c:\opto\qb\progs\data\flow.out"
OPEN "o", #3, "c:\opto\qb\progs\data\comp.out"
OPEN "o", #4, "c:\opto\qb\progs\data\load.out"
OPEN "o", #5, "c:\opto\qb\progs\data\stan.out"
OPEN "o", #6, "c:\opto\qb\progs\data\setp.out"
OPEN "o", #7, "c:\opto\qb\progs\data\vars.out"
OPEN "o", #8, "c:\opto\qb\progs\data\pid.out"
OPEN "o", #10, "c:\opto\qb\progs\data\other.out"
OPEN "o", #11, "c:\opto\qb\progs\data\o2con.out"
OPEN "o", #12, "c:\opto\qb\progs\data\twocon.out"
OPEN "o", #13, "c:\opto\qb\progs\data\tempcon.out"
OPEN "o", #14, "c:\opto\qb\progs\data\adapt.out"

'COLOR 15, 0
PRINT
PRINT
PRINT "Output files are now open"

' Open rs232 com port for communication with load cells
OPEN "com1:9600,n,8,1,cs,ds" FOR RANDOM AS 9

' Controlled or uncontrolled logging
PRINT
PRINT
PRINT
PRINT
PRINT
PRINT
13
INPUT "Input (1) for air control or (0) for no control ", cont%
IF cont% < 0 AND cont% > 1 THEN GOTO 13

' Write starting time to files
WRITE #1, TIME$
WRITE #1, "samtime", "VALUE(1, 0)", "VALUE(1, 1)", "VALUE(1, 2)",
"VALUE(1, 3)", "VALUE(1, 4)", "VALUE(1, 5)", "VALUE(1, 6)", "VALUE(1,
7)", "VALUE(1, 8)", "VALUE(1, 9)", "VALUE(1, 10)", "VALUE(1, 11)",
"VALUE(1, 12)", "VALUE(1, 13)", "VALUE(1, 14)"
WRITE #2, TIME$
WRITE #2, "samtime", "pair", "sair", "lair", "coal", "lime", "VALUE(3, 11)",
"VALUE(3, 12)", "VALUE(3, 13)", "VALUE(3, 14)", "VALUE(3, 15)"
WRITE #3, TIME$
WRITE #3, "samtime", "VALUE(3, 0)", "VALUE(3, 1)", "VALUE(3, 2)",
"VALUE(3, 3)", "VALUE(3, 4)", "VALUE(3, 5)"
WRITE #4, TIME$
WRITE #4, "samtime", "VALUE(2, 6)", "VALUE(2, 7)", "VALUE(2, 8)",
"VALUE(2, 9)", "VALUE(2, 10)", "VALUE(2, 11)", "VALUE(2, 12)"
WRITE #5, TIME$
WRITE #5, "samtime", "VALUE(1, 15)", "VALUE(2, 0)", "VALUE(2, 1)",
"VALUE(2, 2)", "VALUE(2, 3)", "VALUE(2, 4)", "VALUE(2, 5)"
WRITE #6, TIME$
WRITE #6, "samtime", "setpnt(1)", "setpnt(2)", "arat", "risvel", "sp(1)", "sp(2)",
"sp(3)", "sp(4)", "sp(5)", "adapt%", "warmup%", "nu%", "lam(1, 1)"
WRITE #7, TIME$
WRITE #7, "samtime", "a(1,1)", "a(2,1)", "a(3,1)", "b(0,1,1)", "b(1,1,1)",
"b(2,1,1)", "b(3,1,1)"
WRITE #8, TIME$
WRITE #8, "samtime", "kl", "taul", "taudl", "kw", "tauiw", "taudw", "kox",
"taui0", "taudo"
WRITE #10, TIME$
WRITE #10, "samtime", "tube1", "tube2", "tube3", "two", "veloc", "oxy%",
"water%", "oxy%", "twosp"
WRITE #11, TIME$
WRITE #11, "samtime", "%o2", "o2sp", "o2con", "tairi"
WRITE #12, TIME$
WRITE #12, "samtime", "two", "twosp", "twocon", "rhoave", "rhoht"
WRITE #13, TIME$
WRITE #13, "samtime", "L-12", "tsp", "gpccon", "coal"
WRITE #14, TIME$
WRITE #14, "samtime", "adapt", "a1", "a2", "a3", "b0", "b1", "b2", "b3", "c1",
"c2", "c3", "c4", "trace", "lamrls"

SCREEN 0
CLS
END SUB

```

```

SUB initialsp
  SHARED sp(), stavin%, racb%, ravin%, value(), offer()
  SHARED errors%, address%, command%, positions%(), modifiers%(), info%()
  SHARED dorprim, dpipprim, dorsec, dppsec, ffr

CLS
INPUT "primary air setpoint (kg/hr)"; sp(1)
pair = sp(1)
INPUT "secondary air setpoint (kg/hr)"; sp(2)
sair = sp(2)
INPUT "riser solids loading (kg/m3)"; sp(3)
load = sp(3)
INPUT "fuel feedrate (ma)"; sp(4)
ffr = sp(4)

END SUB

SUB loadcell
  SHARED coal, lime

  PRINT #9, "2",
  INPUT #9, mass$
  weight = VAL(MID$(mass$, 4, 6))
  lime = weight
  PRINT #9, "1",
  INPUT #9, mass$
  weight = VAL(MID$(mass$, 5, 6))
  coal = weight

END SUB

SUB numdisp
  SHARED thingy, value(), coal, lime, pair, sair, lair, coalrat
  SHARED limerat, saver%, TM$, two, oxysp, twosp

CLS
COLOR 7, 0
LOCATE 1, 17: PRINT "COEFFICIENTS"
LOCATE 2, 15: PRINT USING "#.##"; a(1, 1)

```

```

LOCATE 3, 15: PRINT USING "#.##"; a(2, 1)
LOCATE 4, 15: PRINT USING "#.##"; a(1, 2)
LOCATE 5, 15: PRINT USING "#.##"; a(2, 2)
LOCATE 2, 20: PRINT USING "#.##"; b(0, 1, 1)
LOCATE 3, 20: PRINT USING "#.##"; b(1, 1, 1)
LOCATE 4, 20: PRINT USING "#.##"; b(2, 1, 1)
LOCATE 5, 20: PRINT USING "#.##"; b(3, 1, 1)
LOCATE 2, 25: PRINT USING "#.##"; b(0, 2, 2)
LOCATE 3, 25: PRINT USING "#.##"; b(1, 2, 2)
LOCATE 4, 25: PRINT USING "#.##"; b(2, 2, 2)
LOCATE 5, 25: PRINT USING "#.##"; b(3, 2, 2)
LOCATE 6, 20: PRINT USING "#.##"; b(0, 1, 2)
LOCATE 7, 20: PRINT USING "#.##"; b(1, 1, 2)
LOCATE 8, 20: PRINT USING "#.##"; b(2, 1, 2)
LOCATE 9, 20: PRINT USING "#.##"; b(3, 1, 2)
LOCATE 6, 25: PRINT USING "#.##"; b(0, 2, 1)
LOCATE 7, 25: PRINT USING "#.##"; b(1, 2, 1)
LOCATE 8, 25: PRINT USING "#.##"; b(2, 2, 1)
LOCATE 9, 25: PRINT USING "#.##"; b(3, 2, 1)
LOCATE 1, 22: PRINT "water out temp"
LOCATE 2, 22: PRINT "tube 1"
LOCATE 3, 22: PRINT "tube 2"
LOCATE 4, 22: PRINT "tube 3"
LOCATE 2, 29: PRINT USING "###.#", value(1, 11)
LOCATE 3, 29: PRINT USING "###.#", value(1, 12)
LOCATE 4, 29: PRINT USING "###.#", value(1, 13)
LOCATE 5, 22: PRINT "trace"
LOCATE 6, 22: PRINT "lamrls"
LOCATE 5, 29: PRINT USING "###.#", trace
LOCATE 6, 29: PRINT USING "#.###", lamrls
LOCATE 2, 35
IF tube%(1) = 1 THEN
  PRINT "ON"
ELSE
  PRINT "OFF"
END IF
LOCATE 3, 35
IF tube%(2) = 1 THEN
  PRINT "ON"
ELSE
  PRINT "OFF"

```

```

END IF
LOCATE 4, 35
IF tube%(3) = 1 THEN
    PRINT "ON"
ELSE
    PRINT "OFF"
END IF

LOCATE 1, 2: PRINT "O2"
LOCATE 2, 2: PRINT "PAIR"
LOCATE 3, 2: PRINT "SAIR"
LOCATE 4, 2: PRINT "LOAD"
LOCATE 5, 2: PRINT "COAL"
LOCATE 6, 2: PRINT "TEMP"
LOCATE 7, 2: PRINT "P-S"
LOCATE 8, 2: PRINT "VEL"
LOCATE 1, 7: PRINT USING "###.#"; setpnt(2)
LOCATE 2, 7: PRINT USING "###.#", sp(1)
LOCATE 3, 7: PRINT USING "###.#", sp(2)
LOCATE 4, 7: PRINT USING "###.#", sp(3)
LOCATE 5, 7: PRINT USING "###.#", sp(4)
LOCATE 6, 7: PRINT USING "###.#"; setpnt(1)
LOCATE 7, 7: PRINT USING "###.#", arat
LOCATE 8, 7: PRINT USING "###.#", risvel
LOCATE 9, 2: PRINT "LAM1": LOCATE 9, 7: PRINT USING "###.#", lam(1, 1)
LOCATE 10, 2: PRINT "LAM2": LOCATE 10, 7: PRINT USING "###.#"; lam(1,
2)
LOCATE 11, 2: PRINT "HORIZ": LOCATE 11, 9: PRINT USING "###.#", nu%
LOCATE 12, 2: PRINT "AIR"
LOCATE 12, 8
IF cont% = 0 THEN
    PRINT "OFF"
ELSE
    PRINT "ON"
END IF
LOCATE 13, 2: PRINT "VEL"
LOCATE 13, 8
IF fly% = 0 THEN
    PRINT "OFF"
ELSE
    PRINT "ON"

END IF
LOCATE 14, 2: PRINT "ADAPT"
LOCATE 14, 8
IF adapt% = 0 THEN
    PRINT "OFF"
ELSE
    PRINT "ON"
END IF
LOCATE 15, 2: PRINT "TEMP"
LOCATE 15, 8
IF warmup% = 0 THEN
    PRINT "OFF"
ELSE
    PRINT "ON"
END IF
LOCATE 11, 1: PRINT "GPC"
LOCATE 11, 6
IF warmup% = 0 THEN
    PRINT "OFF"
ELSE
    PRINT "ON"
END IF
LOCATE 12, 1: PRINT "TEMP"
LOCATE 12, 10: PRINT USING "###.#", setpnt(1)

LOCATE 13, 1: PRINT "ADAPT"
LOCATE 13, 6
IF adapt% = 0 THEN
    PRINT "OFF"
ELSE
    PRINT "ON"
END IF
LOCATE 14, 1: PRINT "HORIZ"
LOCATE 14, 10: PRINT USING "###.#", nu%
LOCATE 15, 1: PRINT "LAMDA"
LOCATE 15, 10: PRINT USING "###.#", lam(1, 1)

LOCATE 1, 1: PRINT "VAR"
LOCATE 1, 6: PRINT "CONT"

```

```

LOCATE 1, 12: PRINT "S.P."
LOCATE 1, 18: PRINT "VAL"
LOCATE 2, 1: PRINT "O2"
LOCATE 2, 6
IF oxy% = 1 THEN
    PRINT "ON"
ELSE
    PRINT "OFF"
END IF
LOCATE 2, 10: PRINT USING "###.#", oxy%
LOCATE 3, 1: PRINT "AIR"
LOCATE 3, 6
IF cont% = 1 THEN
    PRINT "ON"
ELSE
    PRINT "OFF"
END IF
LOCATE 4, 1: PRINT "PAIR"
LOCATE 4, 10: PRINT USING "###.#", sp(1)
LOCATE 5, 1: PRINT "SAIR"
LOCATE 5, 10: PRINT USING "###.#", sp(2)
LOCATE 6, 1: PRINT "P.S"
LOCATE 6, 10: PRINT USING "###.#", arat
LOCATE 7, 1: PRINT "LOAD"
LOCATE 7, 6
IF cont% = 1 THEN
    PRINT "ON"
ELSE
    PRINT "OFF"
END IF
LOCATE 7, 10: PRINT USING "###.#", sp(3)
LOCATE 8, 1: PRINT "TWO"
LOCATE 8, 6
IF water% = 1 THEN
    PRINT "ON"
ELSE
    PRINT "OFF"
END IF
LOCATE 8, 10: PRINT USING "###.#", twosp
LOCATE 9, 1: PRINT "VEL"
LOCATE 9, 6

IF fly% = 1 THEN
    PRINT "ON"
ELSE
    PRINT "OFF"
END IF
LOCATE 9, 10: PRINT USING "###.#", risvel
LOCATE 10, 1: PRINT "COAL"
LOCATE 10, 10: PRINT USING "###.#", sp(4)

COLOR 7, 0
LOCATE 16, 5: PRINT "TEMPERATURES"
thingy = value(1, 1)
CALL tcolor
LOCATE 17, 5: PRINT "L3"
LOCATE 17, 10: PRINT USING "###.#", value(1, 1)
LOCATE 17, 15: PRINT "(C)"
thingy = value(1, 3)
CALL tcolor
LOCATE 18, 5: PRINT "L6"
LOCATE 18, 10: PRINT USING "###.#", value(1, 3)
LOCATE 18, 15: PRINT "(C)"
thingy = value(1, 7)
CALL tcolor
LOCATE 19, 5: PRINT "L12"
LOCATE 19, 10: PRINT USING "###.#", value(1, 7)
LOCATE 19, 15: PRINT "(C)"
thingy = value(1, 14)
CALL tcolor
LOCATE 20, 5: PRINT "R5"
LOCATE 20, 10: PRINT USING "###.#", value(1, 14)
LOCATE 20, 15: PRINT "(C)"
COLOR 7, 0
LOCATE 1, 40: PRINT "L-VALVE DELTA P"
COLOR 9, 0
LOCATE 2, 40: PRINT "DPL1"
LOCATE 2, 45: PRINT USING "###.#", value(1, 15)
LOCATE 2, 50: PRINT "(KPA)"
LOCATE 3, 40: PRINT "DPL2"
LOCATE 3, 45: PRINT USING "###.#", value(2, 0)
LOCATE 3, 50: PRINT "(KPA)"

```



```

LOCATE 4, 40: PRINT "DPL3"
LOCATE 4, 45: PRINT USING "####", value(2, 1)
LOCATE 4, 50: PRINT "(KPA)"
LOCATE 5, 40: PRINT "DPL4"
LOCATE 5, 45: PRINT USING "####", value(2, 2)
LOCATE 5, 50: PRINT "(KPA)"
LOCATE 6, 40: PRINT "DPL5"
LOCATE 6, 45: PRINT USING "####", value(2, 3)
LOCATE 6, 50: PRINT "(KPA)"
LOCATE 7, 40: PRINT "DPL6"
LOCATE 7, 45: PRINT USING "####", value(2, 4)
LOCATE 7, 50: PRINT "(KPA)"
LOCATE 8, 40: PRINT "DPL7"
LOCATE 8, 45: PRINT USING "####", value(2, 5)
LOCATE 8, 50: PRINT "(KPA)"
LOCATE 1, 40: PRINT "COEFFICIENTS"
LOCATE 2, 40: PRINT "A1"
LOCATE 3, 40: PRINT "A2"
LOCATE 4, 40: PRINT "A3"
LOCATE 5, 40: PRINT "B0"
LOCATE 6, 40: PRINT "B1"
LOCATE 7, 40: PRINT "B2"
LOCATE 8, 40: PRINT "B3"
LOCATE 2, 45: PRINT USING "##.##", a(1, 1)
LOCATE 3, 45: PRINT USING "##.##", a(2, 1)
LOCATE 4, 45: PRINT USING "##.##", a(3, 1)
LOCATE 5, 45: PRINT USING "##.##", b(0, 1, 1)
LOCATE 6, 45: PRINT USING "##.##", b(1, 1, 1)
LOCATE 7, 45: PRINT USING "##.##", b(2, 1, 1)
LOCATE 8, 45: PRINT USING "##.##", b(3, 1, 1)

COLOR 7, 0
LOCATE 1, 62: PRINT "RISER LOADING"
COLOR 12, 0
LOCATE 2, 60: PRINT "dP1"
LOCATE 2, 65: PRINT USING "####", value(2, 6)
LOCATE 2, 70: PRINT "(KG/M3)"
LOCATE 3, 60: PRINT "dP2"
LOCATE 3, 65: PRINT USING "####", value(2, 7)
LOCATE 3, 70: PRINT "(KG/M3)"
LOCATE 4, 60: PRINT "dP3"

```

```

LOCATE 4, 65: PRINT USING "####", value(2, 8)
LOCATE 4, 70: PRINT "(KG/M3)"
LOCATE 5, 60: PRINT "dP4"
LOCATE 5, 65: PRINT USING "####", value(2, 9)
LOCATE 5, 70: PRINT "(KG/M3)"
LOCATE 6, 60: PRINT "dP5"
LOCATE 6, 65: PRINT USING "####", value(2, 10)
LOCATE 6, 70: PRINT "(KG/M3)"
LOCATE 7, 60: PRINT "dP6"
LOCATE 7, 65: PRINT USING "####", value(2, 11)
LOCATE 7, 70: PRINT "(KG/M3)"
LOCATE 8, 60: PRINT "dP7"
LOCATE 8, 65: PRINT USING "####", value(2, 12)
LOCATE 8, 70: PRINT "(KG/M3)"
COLOR 7, 0
LOCATE 10, 25: PRINT "FEEDRATES"
COLOR 13, 0
LOCATE 11, 20: PRINT "COAL"
LOCATE 11, 30: PRINT USING "##.##", coal
LOCATE 11, 36: PRINT "(KG)"
LOCATE 12, 20: PRINT "LIME"
LOCATE 12, 30: PRINT USING "##.##", lime
LOCATE 12, 36: PRINT "(KG)"
LOCATE 13, 20: PRINT "PRIM.AIR"
LOCATE 13, 30: PRINT USING "##.##", pair
LOCATE 13, 36: PRINT "(KG/HR)"
LOCATE 14, 20: PRINT "SEC.AIR"
LOCATE 14, 30: PRINT USING "##.##", sair
LOCATE 14, 36: PRINT "(KG/HR)"
LOCATE 15, 20: PRINT "L-V.AIR"
LOCATE 15, 30: PRINT USING "##.##", lair
LOCATE 15, 36: PRINT "(L/MIN)"
LOCATE 17, 20: PRINT "COAL"
LOCATE 17, 30: PRINT USING "##.##", coalrat
LOCATE 17, 36: PRINT "(KG/HR)"
LOCATE 18, 20: PRINT "k4"
LOCATE 19, 20: PRINT "tau14"
LOCATE 20, 20: PRINT "tau44"

LOCATE 18, 27: PRINT USING "##.##", k4
LOCATE 19, 27: PRINT USING "##.##", tau14

```

```

LOCATE 20, 27: PRINT USING "###.##", taud4
LOCATE 21, 27: PRINT pidt%
COLOR 7, 0
LOCATE 10, 44: PRINT "VALVES (MA)"
COLOR 10, 0
LOCATE 11, 45: PRINT "COAL"
LOCATE 11, 50: PRINT USING "###.##", value(3, 14)
LOCATE 12, 45: PRINT "LIME"
LOCATE 12, 50: PRINT USING "###.##", value(3, 15)
LOCATE 13, 45: PRINT "PAIR"
LOCATE 13, 50: PRINT USING "###.##", value(3, 11)
LOCATE 14, 45: PRINT "SAIR"
LOCATE 14, 50: PRINT USING "###.##", value(3, 12)
LOCATE 15, 45: PRINT "LAIR"
LOCATE 15, 50: PRINT USING "###.##", value(3, 13)
LOCATE 16, 45: PRINT "RI": LOCATE 16, 50: PRINT USING "###.##", kd
LOCATE 17, 45: PRINT "RI": LOCATE 17, 50: PRINT USING "###.##", tauil
LOCATE 18, 45: PRINT "tdl": LOCATE 18, 50: PRINT USING "###.##", tauil
LOCATE 19, 45: PRINT "kw": LOCATE 19, 50: PRINT USING "###.##", kw
LOCATE 20, 45: PRINT "tiw": LOCATE 20, 50: PRINT USING "###.##", tauiw
LOCATE 21, 45: PRINT "tdw": LOCATE 21, 50: PRINT USING "###.##", tauiw
LOCATE 19, 60: PRINT "kox": LOCATE 19, 65: PRINT USING "###.##", kox
LOCATE 20, 60: PRINT "tio": LOCATE 20, 65: PRINT USING "###.##", tauio
LOCATE 21, 60: PRINT "ido": LOCATE 21, 65: PRINT USING "###.##", tauido

COLOR 7, 0
LOCATE 10, 63: PRINT "FLUE GAS"
COLOR 2, 0
LOCATE 11, 60: PRINT "O2"
LOCATE 11, 65: PRINT USING "###.##", value(3, 0)
LOCATE 11, 71: PRINT "%"
LOCATE 12, 60: PRINT "CO"
LOCATE 12, 65: PRINT USING "###.##", value(3, 1)
LOCATE 12, 71: PRINT "PPM"
LOCATE 13, 60: PRINT "CO2"
LOCATE 13, 65: PRINT USING "###.##", value(3, 2)
LOCATE 13, 71: PRINT "%"
LOCATE 14, 60: PRINT "CH4"
LOCATE 14, 65: PRINT USING "###.##", value(3, 3)
LOCATE 14, 71: PRINT "%"
LOCATE 15, 60: PRINT "NOX"

LOCATE 15, 65: PRINT USING "#####", value(3, 4)
LOCATE 15, 71: PRINT "PPM"
LOCATE 16, 60: PRINT "SO2"
LOCATE 16, 65: PRINT USING "###.##", value(3, 5)
LOCATE 16, 71: PRINT "PPM"
COLOR 7, 0
LOCATE 22, 38: PRINT "SAMPLE TIME....", TMS$
LOCATE 23, 38
IF saver% = 0 THEN PRINT "FILE STATUS = WAITING"
IF saver% = 1 THEN
    COLOR 4, 0: PRINT "FILE STATUS = SAVING "
END IF
COLOR 7, 0
END SUB

SUB opto
' subroutine to call optomux driver (optoware)
SHARED errors%, address%, command%, positions%(), modifiers%(), info%()
CALL OPTOWARE(errors%, address%, command%, VARPTR(positions%(0)),
VARPTR(modifiers%(0)), VARPTR(info%(0)))
' check for errors

IF errors% < 0 THEN CALL errmsg
END SUB

FUNCTION orifice (dor, dpip, p1, p2)
SHARED cp, cv
N% = 0
BETA = dor / dpip
IF BETA < .5 THEN ko = .6004 + .35 * BETA ^ 4 - .052 * (.5 - BETA) ^ (3 / 2)
IF BETA > .7 THEN ko = .6004 + .35 * BETA ^ 4 + .62 * (BETA - .7) ^ (5 / 3)
IF (BETA >= .5 AND BETA <= .7) THEN ko = .6004 + .35 * BETA ^ 4
b = (.002 + .026 * BETA ^ 4) * ko
XE = (p1 - p2) / p1
y = cp / cv
SMALL = 0
BIG = 1

```

```

M = .5
Y1 = 1 - (.41 + .35 * BETA ^ 4) * XE / y
RO1 = p1 * 291 / 8314.34 / 293.15
F = 3.14159 / 4 * dor * Y1 * (2 * RO1 * (p1 - p2)) ^ (1 / 2)
23
value = M ^ (3 / 2) - F * ko * M ^ (1 / 2) - 1000 * F * b * ko * (3.14159 / 4 * MU
* dpip / 10000) ^ (1 / 2)
N% = N% + 1
IF value < 0 THEN SMALL = M
IF value >= 0 THEN BIG = M
IF (BIG - SMALL) < .000001 THEN GOTO 55
M = (BIG + SMALL) / 2
GOTO 23
55
orifice = M

END FUNCTION

SUB powerup
SHARED pupclr%, reset%, conout%, probtp%, probtp%, setoff%
SHARED errors%, address%, command%, positions%(), modifiers%(), info%()
SHARED offset%()

' Power up clear optomux
command% = pupclr%
FOR address% = 1 TO 3
CALL opto
NEXT address%

' Reset modules
command% = reset%
FOR address% = 1 TO 3
CALL opto
NEXT address%

' Configure output modules
command% = conout%
address% = 3
FOR i% = 0 TO 4
positions%(i%) = i% + 11
NEXT i%

```

```

positions%(5) = -1
CALL opto

' Set temperature probe type (k-type)
command% = probtp%
address% = 1
FOR i% = 0 TO 14
positions%(i%) = i%
NEXT i%
positions%(15) = -1
info%(0) = 5
CALL opto

' Set offsets for input modules
command% = setoff%
address% = 1
FOR i% = 0 TO 15
positions%(i%) = i%
info%(i%) = offset%(address%, i%)
NEXT i%
CALL opto

address% = 2
FOR i% = 0 TO 15
positions%(i%) = i%
info%(i%) = offset%(address%, i%)
NEXT i%
CALL opto

address% = 3
FOR i% = 0 TO 8
positions%(i%) = i%
info%(i%) = offset%(address%, i%)
NEXT i%
positions%(9) = -1
CALL opto

END SUB

SUB saveit
SHARED value(), samtime, pair, sair, lair, coal, lime, two, veloc

```

```

' SUBROUTINE TO WRITE DATA TO FILES
'
' WRITE TEMPERATURES
WRITE #1, samtime, value(1, 0), value(1, 1), value(1, 2), value(1, 3), value(1, 4),
value(1, 5), value(1, 6), value(1, 7), value(1, 8), value(1, 9), value(1, 10), value(1,
11), value(1, 12), value(1, 13), value(1, 14)

' WRITE AIR FLOWS AND CONTROL VALVE POSITIONS
WRITE #2, samtime, pair, sair, lair, coal, lime, value(3, 11), value(3, 12), value(3,
13), value(3, 14), value(3, 15)

' WRITE GAS ANALYSIS
WRITE #3, samtime, value(3, 0), value(3, 1), value(3, 2), value(3, 3), value(3, 4),
value(3, 5)

' WRITE RISER SOLIDS LOADINGS
WRITE #4, samtime, value(2, 6), value(2, 7), value(2, 8), value(2, 9), value(2, 10),
value(2, 11), value(2, 12)

' WRITE STANDPIPE PRESSURE DROPS
WRITE #5, samtime, value(1, 15), value(2, 0), value(2, 1), value(2, 2), value(2, 3),
value(2, 4), value(2, 5)

' write set-points
WRITE #6, samtime, setpnt(1), setpnt(2), arat, risvel, sp(1), sp(2), sp(3), sp(4),
sp(5), adapt%, warmup%, nut%, lam(1, 1)

' write coefficients
WRITE #7, samtime, a(1, 1), a(2, 1), a(3, 1), b(0, 1, 1), b(1, 1, 1), b(2, 1, 1), b(3,
1, 1)

' write pid controll configurations
WRITE #8, samtime, kl, tauil, taucl, kw, tauiw, tauclw, kox, tauio, tauco

' write extra stuff for two and oxygen control
WRITE #10, samtime, tube%(1), tube%(2), tube%(3), two, veloc, oxy%, water%,
oxy%, twosp

WRITE #11, samtime / 60, value(3, 0), oxy%, oxy%, sp(1) + sp(2) + 35
WRITE #12, samtime / 60, two, twosp, water%, value(2, 10), value(2, 11)

WRITE #13, samtime / 60, value(1, 1), setpnt(1), warmup%, ma(4)
WRITE #14, samtime / 60, adapt%, a(1, 1), a(2, 1), a(3, 1), b(0, 1, 1), b(1, 1, 1),
b(2, 1, 1), b(3, 1, 1), cc(1), cc(2), cc(3), cc(4), trace, lamrls

END SUB

FUNCTION seconds
SHARED TMS$

TMS$ = TIME$
seconds = VAL(MID$(TMS$, 1, 2)) * 3600 + VAL(MID$(TMS$, 4, 2)) * 60 +
VAL(MID$(TMS$, 7, 2))

END FUNCTION

SUB tcolor
'SHARED thingy

' SUBROUTINE TO DETERMINE DISPLAY COLOR FOR TEMPERATURE
IF thingy < 800 THEN
' COLOR 1, 0
ELSEIF thingy > 950 THEN
' COLOR 4, 0
ELSE
COLOR 7, 0
END IF

END SUB

```

330

```

b(2, 1, 1) = 0
b(3, 1, 1) = 0
b(0, 1, 2) = 0
b(1, 1, 2) = 0
b(2, 1, 2) = 0
b(3, 1, 2) = 0
b(0, 2, 1) = 0
b(1, 2, 1) = 0
b(2, 2, 1) = 0
b(3, 2, 1) = 0
b(0, 2, 2) = 0
b(1, 2, 2) = 0
b(2, 2, 2) = 0
b(3, 2, 2) = 0

```

```

FOR lo% = 1 TO 2
FOR i% = 1 TO 10
FOR j% = 1 TO 10
IF i% = j% THEN
p(i%, j%, lo%) = initial
ELSE
p(i%, j%, lo%) = 0
END IF
NEXT j%
NEXT i%
NEXT lo%

```

```

times% = 1

```

```

675

```

```

y(0, 1) = temp1

```

```

yff(0, 1) = y(0, 1) - cc(1) * yff(-1, 1) - cc(2) * yff(-2, 1) - cc(3) * yff(-3, 1) - cc(4) * yff(-
4, 1)

```

```

uff(-1, 1) = u(-1, 1) - cc(1) * uff(-2, 1) - cc(2) * uff(-3, 1) - cc(3) * uff(-4, 1) * cc(4) *
uff(-5, 1)
duff(-1, 1) = u(-1, 1) - u(-2, 1) - cc(1) * duff(-2, 1) - cc(2) * duff(-3, 1) - cc(3) * duff(-
4, 1) - cc(4) * duff(-5, 1)
dyff(0, 1) = y(0, 1) - y(-1, 1) - cc(1) * dyff(-1, 1) - cc(2) * dyff(-2, 1) - cc(3) * dyff(-3,
1) - cc(4) * dyff(-4, 1)

```

```

spcc(1) = setpnt(1)

```

```

IF adapt% = 0 THEN GOTO 678

```

```

PERFORM RLS WITH VARIABLE FORGETTING FACTOR TO FIT
CONSTANTS
GOSUB 8100

```

```

678

```

```

IF warmup% = 0 THEN GOTO 689

```

```

IF nu% > 5 THEN nu% = 5

```

```

li% = 1

```

```

lo% = 1

```

```

e(0, 1, lo%) = 1

```

```

F(0, 1, lo%) = 1 - a(1, lo%) + cc(1)

```

```

F(1, 1, lo%) = a(1, lo%) - a(2, lo%) + cc(2)

```

```

F(2, 1, lo%) = a(2, lo%) - a(3, lo%) + cc(3)

```

```

F(3, 1, lo%) = a(3, lo%) + cc(4)

```

```

FOR j% = 1 TO nc%

```

```

F(0, j% + 1, lo%) = F(1, j%, lo%) - (a(1, lo%) - 1) * F(0, j%, lo%)

```

```

F(1, j% + 1, lo%) = F(2, j%, lo%) - (a(2, lo%) - a(1, lo%)) * F(0, j%, lo%)

```

```

F(2, j% + 1, lo%) = F(3, j%, lo%) - (a(3, lo%) - a(2, lo%)) * F(0, j%, lo%)

```

```

F(3, j% + 1, lo%) = a(3, lo%) * F(0, j%, lo%)

```

```

FOR i% = 0 TO j% - 1

```

```

e(i%, j% + 1, lo%) = e(i%, j%, lo%)

```

```

NEXT i%

```

```

e(j%, j% + 1, lo%) = F(0, j%, lo%)

```

```

NEXT j%

```

```

NEXT lo%

```

```

FOR lo% = 1 TO numout%

```

```

' FOR li% = 1 TO numin%
  FOR i% = 0 TO 3
    g(i%, 1, lo%, li%) = b(i%, lo%, li%)
  NEXT i%
NEXT li%
'
FOR j% = 1 TO nc%
  FOR li% = 1 TO numin%
    FOR k% = 0 TO j% + 3
      g(k%, j% + 1, lo%, li%) = 0
    NEXT k%
    FOR i% = 0 TO j%
      FOR k% = 0 TO 3
        g(i% + k%, j% + 1, lo%, li%) = g(i% + k%, j% + 1, lo%, li%) +
          e(i%, j% + 1, lo%) * b(k%, lo%, li%)
      NEXT k%
    NEXT i%
  NEXT li%
'
NEXT j%
NEXT lo%
FOR i% = 0 TO 3
  gamma(i%, 0) = 0
  gp(0) = 0
NEXT i%
'
FOR j% = 0 TO nc%
  gp(j% + 1) = (e(j%, j% + 1) * b(0, 1, 1) + gamma(0, j%))
  gamma(0, j% + 1) = gamma(1, j%) + e(j%, j% + 1, 1) * b(1, 1, 1) - cc(1) *
    gp(j% + 1)
  gamma(1, j% + 1) = gamma(2, j%) + e(j%, j% + 1, 1) * b(2, 1, 1) - cc(2) *
    gp(j% + 1)
  gamma(2, j% + 1) = gamma(3, j%) + e(j%, j% + 1, 1) * b(3, 1, 1) - cc(3) *
    gp(j% + 1)
  gamma(3, j% + 1) = -cc(4) * gp(j% + 1)
NEXT j%
'
FOR i% = 1 TO nc%
  FOR j% = 1 TO nc%
    IF j% > i% THEN
      gm(i%, j%) = 0
    ELSE
      gm(i%, j%) = gp(i% - j%)
    END IF
  NEXT j%
NEXT i%

```

```

      END IF
    NEXT j%
  NEXT i%
'
FOR i% = 1 TO nc%
  id(i%, i%) = id(i%, i%) * lam(i%, 1)
NEXT i%
'
CALL mattran(nc%, nu%, gm(), holda())
CALL matmult(nu%, nc%, nu%, holda(), gm(), holdb())
CALL MATADD(nu%, nu%, holdb(), id(), holde())
CALL MATINVS(nu%, holde(), holdc())
CALL matmult(nu%, nu%, nc%, holdc(), holda(), gt())
'
FOR i% = 1 TO 20
  FOR j% = 1 TO 20
    IF i% = j% THEN
      id(i%, j%) = 1
    ELSE
      id(i%, j%) = 0
    END IF
  NEXT j%
NEXT i%
'
w(0, 1) = y(0, 1)
FOR i% = 1 TO nc%
  w(i%, 1) = alph(1) * w(i% - 1, 1) + (1 - alph(1)) * spcc(1)
NEXT i%
'
FOR i% = 1 TO nc%
  lf(i%, 1) = F(0, i%, 1) * yf(0, 1) + F(1, i%, 1) * yf(-1, 1) + F(2, i%, 1) * yf(-2,
    1) + F(3, i%, 1) * yf(-3, 1)
  lf(i%, 1) = lf(i%, 1) + gamma(0, i%) * (uf(-1, 1) - uf(-2, 1)) + gamma(1, i%) *
    (uf(-2, 1) - uf(-3, 1)) + gamma(2, i%) * (uf(-3, 1) - uf(-4, 1)) + gamma(3, i%) *
    (uf(-4, 1) - uf(-5, 1))
NEXT i%
'
CALL matsub(nc%, 1, w(), lf(), holda())
CALL matmult(nu%, nc%, 1, gt(), holda(), holdb())

```

```

u(0, 1) = u(-1, 1) + holdb(1, 1)
'u(0, 2) = u(-1, 2) + holdb(nu% + 1, 1)

TF u(0, 1) > 16 THEN
' u(0, 1) = 16
END IF
TF u(0, 1) < 4 THEN
' u(0, 1) = 4
END IF
IF u(0, 1) > 6.8 THEN
u(0, 1) = 6.8
END IF
IF u(0, 1) < -5.2 THEN
u(0, 1) = -5.2
END IF

TF u(0, 2) > 250 THEN
' u(0, 2) = 250
END IF
TF u(0, 2) < 150 THEN
' u(0, 2) = 150
END IF

689
IF warmup% = 0 THEN
' u(0, 1) = sp(4)
u(0, 1) = sp(4) - 9.2
' u(0, 2) = tairi
ELSE
' sp(4) = u(0, 1)
sp(4) = u(0, 1) + 9.2
' tairi = u(0, 2)
END IF

i% = 1
FOR j% = -5 TO -1
y(j%, i%) = y(j% + 1, i%)
u(j%, i%) = u(j% + 1, i%)
yf(j%, i%) = yf(j% + 1, i%)

```

```

uf(j%, i%) = uf(j% + 1, i%)
dyf(j%, i%) = dyf(j% + 1, i%)
duf(j%, i%) = duf(j% + 1, i%)
NEXT j%

```

```

change% = 1

```

```

CHAIN "c:\opto\qb\progst1"

```

```

RETURN

```

```

8100

```

```

' SUBROUTINE TO PERFORM VARIABLE FORGETTING FACTOR RLS

```

```

lo% = 1

```

```

fi(1) = -dyf(-1, 1)
fi(2) = -dyf(-2, 1)
fi(3) = -dyf(-3, 1)
fi(4) = duf(-1, 1)
fi(5) = duf(-2, 1)
fi(6) = duf(-3, 1)
fi(7) = duf(-4, 1)

theta(1) = a(1, lo%)
theta(2) = a(2, lo%)
theta(3) = a(3, lo%)
theta(4) = b(0, lo%, 1)
theta(5) = b(1, lo%, 1)
theta(6) = b(2, lo%, 1)
theta(7) = b(3, lo%, 1)

```

```

error = 0
FOR i% = 1 TO 10
error = error + fi(i%) * theta(i%)
NEXT i%

```



```

error = dyf(0, 1) - error
FOR i% = 1 TO 10
FOR j% = 1 TO 10
    p(i%, j%) = p(i%, j%, lo%)
NEXT j%
NEXT i%

FOR i% = 1 TO 10
FOR j% = 1 TO 10
    holda(i%, j%) = 0
NEXT j%
NEXT i%

FOR i% = 1 TO 10
    holda(i%, 1) = fi(i%)
NEXT i%

CALL mattran(10, 10, holda(), holdb())
CALL matmult(10, 10, 10, holdb(), pt(), holdc())

value1 = 0
FOR i% = 1 TO 10
    value1 = value1 + holdc(1, i%) * fi(i%)
NEXT i%

lamrls = 1 - error ^ 2 / (1000 * (1 + value1))

IF lamrls < .9 THEN lamrls = .9

CALL matmult(10, 10, 10, pt(), holda(), holddd())

FOR i% = 1 TO 10
    k(i%) = holddd(i%, 1) / (1 + value1)
NEXT i%

FOR i% = 1 TO 10
    theta(i%) = theta(i%) + k(i%) * error
NEXT i%

```

```

FOR i% = 1 TO 10
FOR j% = 1 TO 10
    holde(i%, j%) = 0
NEXT j%
NEXT i%

FOR i% = 1 TO 10
    holde(i%, 1) = k(i%)
NEXT i%

CALL matmult(10, 10, 10, holde(), holdc(), holdff())
CALL matsub(10, 10, pt(), holdff(), wrls())

IF wrls(1, 1) / lamrls <= 1 THEN
    FOR i% = 1 TO 10
    FOR j% = 1 TO 10
        pt(i%, j%) = wrls(i%, j%) / lamrls
    NEXT j%
    NEXT i%
ELSE
    FOR i% = 1 TO 10
    FOR j% = 1 TO 10
        pt(i%, j%) = wrls(i%, j%)
    NEXT j%
    NEXT i%
END IF

trace = 0

FOR i% = 1 TO 10
FOR j% = 1 TO 10
    p(i%, j%, lo%) = pt(i%, j%)
IF i% = j% THEN
    trace = trace + p(i%, j%, lo%)
END IF
NEXT j%
NEXT i%

a(1, lo%) = theta(1)
a(2, lo%) = theta(2)
a(3, lo%) = theta(3)

```

```

b(0, lo%, 1) = theta(4)
b(1, lo%, 1) = theta(5)
b(2, lo%, 1) = theta(6)
b(3, lo%, 1) = theta(7)

```

```

NEXT lo%

```

```

RETURN

```

```

REM $STATIC

```

```

SUB MATADD (n%, m%, aay(), bbx(), c())

```

```

' SUBROUTINE TO ADD TWO MATRICES

```

```

' N=# ROWS

```

```

' M=# COLUMNS

```

```

' A=FIRST MATRIX

```

```

' B=SECOND MATRIX

```

```

' C=RESULTANT MATRIX

```

```

FOR i%= 1 TO n%

```

```

  FOR j%= 1 TO m%

```

```

    c(i%, j%) = aay(i%, j%) + bbx(i%, j%)

```

```

  NEXT j%

```

```

NEXT i%

```

```

END SUB

```

```

SUB MATINVR (n%, aay(), bbx())

```

```

' SUBROUTINE TO TAKE THE INVERSE OF A MATRIX (MUST BE
SQUARE)

```

```

' N=# ROWS = # COLUMNS

```

```

' A = INPUT MATRIX

```

```

' B=INVERSE OF MATRIX A

```

```

DIM id(1 TO 16, 1 TO 16)

```

```

' SET UP IDENTITY MATRIX

```

```

  FOR i% = 1 TO n%

```

```

    FOR j% = 1 TO n%

```

```

      IF i% = j% THEN

```

```

        bbx(i%, j%) = 1

```

```

      ELSE

```

```

        bbx(i%, j%) = 0

```

```

      END IF

```

```

    NEXT j%

```

```

  NEXT i%

```

```

' PUT A INTO ANOTHER MATRIX TO SAVE A AS IT IS

```

```

  FOR i% = 1 TO n%

```

```

    FOR j% = 1 TO n%

```

```

      id(i%, j%) = aay(i%, j%)

```

```

    NEXT j%

```

```

  NEXT i%

```

```

' GAUSS-JORDAN ELIMINATION TO TURN ID INTO IDENTITY MATRIX

```

```

' RESULT IN B IS THE INVERSE

```

```

  FOR i% = 1 TO n%

```

```

    IF id(i%, i%) = 0 THEN

```

```

      FOR j% = i% + 1 TO n%

```

```

        IF id(j%, i%) < 0 THEN

```

```

          FOR k% = 1 TO n%

```

```

            HOLD = id(i%, k%)

```

```

            id(i%, k%) = id(j%, k%)

```

```

            id(j%, k%) = HOLD

```

```

          HOLD = bbx(i%, k%)

```

```

          bbx(i%, k%) = bbx(j%, k%)

```

```

          bbx(j%, k%) = HOLD

```

```

        NEXT k%

```

```

      GOTO 10

```

```

END IF
IF j% = n% THEN
  PRINT "CANNOT TAKE INVERSE OF MATRIX"
  STOP
END IF
END IF
NEXT j%
END IF

10  HOLD = id(i%, i%)

    FOR j% = 1 TO n%
      id(i%, j%) = id(i%, j%) / HOLD
      bbx(i%, j%) = bbx(i%, j%) / HOLD
    NEXT j%

    FOR j% = 1 TO n%
      IF j% = i% THEN GOTO 66
      HOLD = id(j%, i%) / id(i%, i%)
      FOR k% = 1 TO n%
        id(j%, k%) = id(j%, k%) - HOLD * id(i%, k%)
        bbx(j%, k%) = bbx(j%, k%) - HOLD * bbx(i%, k%)
      NEXT k%
    NEXT j%

66  NEXT j%

    NEXT i%

  END SUB

SUB matmult (n%, m%, o%, aay(), bbx(), c())

  SUBROUTINE TO MULTIPLY TWO MATRICES
  N = # ROWS OF MATRIX A
  M = # COLUMNS OF MATRIX A
  O = # ROWS OF MATRIX B (I THINK ITS COLUMNS)
  A=FIRST MATRIX
  B=SECOND MATRIX
  C=RESULTANT MATRIX

```

```

IF n% > m% THEN
  BIG% = n%
ELSE
  BIG% = m%
END IF

FOR i% = 1 TO 10
FOR j% = 1 TO 30
  c(i%, j%) = 0
NEXT j%
NEXT i%

FOR k% = 1 TO n%
FOR i% = 1 TO o%
FOR j% = 1 TO m%
  c(k%, i%) = c(k%, i%) + aay(k%, j%) * bbx(j%, i%)
NEXT j%
NEXT i%
NEXT k%

END SUB

SUB matsub (n%, m%, aay(), bbx(), c())

  SUBROUTINE TO SUBTRACT TWO MATRICES
  N=# ROWS
  M=# COLUMNS
  A=FIRST MATRIX
  B=SECOND MATRIX
  C=RESULTANT MATRIX

FOR i% = 1 TO n%
  FOR j% = 1 TO m%
    c(i%, j%) = aay(i%, j%) - bbx(i%, j%)
  NEXT j%
NEXT i%

END SUB

SUB mattran (n%, m%, aay(), bbx())

```

```

' SUBROUTINE TO TAKE THE TRANSPOSE OF A MATRIX
' N=# ROWS OF A
' M=# COLUMNS OF A
' A=INPUT MATRIX
' B=TRANSPOSE OF A

  FOR i% = 1 TO n%
    FOR j% = 1 TO m%
      bbx(j%, i%) = aay(i%, j%)
    NEXT j%
  NEXT i%

END SUB

```

## **APPENDIX D - PROGRAM LISTING - DYNAMIC SIMULATOR OF A CIRCULATING FLUIDIZED BED COMBUSTOR**

This appendix provides the listing of the QuickBasic program written for the dynamic simulation of a circulating fluidized bed combustor. It employs the model described in Chapter 4.

```

'S$STATIC

'S$INCLUDE: c:\model\common.bi'

DECLARE SUB mattran (n%, m%, aay(), bbx())
DECLARE SUB matmult (n%, m%, o%, aay(), bbx(), c())
DECLARE SUB MATADD (n%, m%, aay(), bbx(), c())
DECLARE SUB matinvs (n%, aay(), bbx())
DECLARE SUB matsub (n%, m%, aay(), bbx(), c())

REM $DYNAMIC

DIM t(0 TO 50, 13)
DIM nt(0 TO 50, 13)
DIM ot(0 TO 50, 13)
DIM eh(0 TO 30)
DIM ebsh(0 TO 30)
DIM nts(0 TO 13)
' DIM aaa(0 TO 50, 0 TO 50)
' DIM ff(0 TO 50)

OPEN "O", #1, "SIM.PRN"
OPEN "I", #2, "c:\model\temp.in"
OPEN "T", #3, "c:\model\coal.in"

' SCREEN 8
CLS
COLOR 5, 0
DIM kmo2a(1 TO 21)
DIM kmco2a(0 TO 21)
DIM ar(1 TO 3)
DIM load(1 TO 3)
DIM molfo2(0 TO 21)

```

```

DIM molfo2a(0 TO 21)
DIM kgco2a(0 TO 3)
DIM KGH2OO(0 TO 3)
DIM KGSO2O(0 TO 3)
DIM KGN2O(0 TO 3)
DIM KGCAO4O(0 TO 3)
DIM KGCAOO(0 TO 3)
DIM kgasho(0 TO 3)
DIM kgco(0 TO 3)
DIM KGO2O(0 TO 3)
DIM KGCL2O(0 TO 3)
DIM ts(0 TO 13)
DIM qout(1 TO 3)
DIM tg(0 TO 3)
DIM bedc(0 TO 13)
DIM bedash(0 TO 13)
DIM smass(0 TO 13)
DIM rin(1 TO 13)
DIM h(1 TO 3)
DIM gamma(0 TO 3, 0 TO 20)
DIM gp(0 TO 20)
DIM aay(1 TO 30, 1 TO 30)
DIM bbx(1 TO 30, 1 TO 30)
DIM c(1 TO 20, 1 TO 30)
DIM e(0 TO 16, 1 TO 16, 2)
DIM F(0 TO 3, 1 TO 16, 2)
DIM g(0 TO 18, 1 TO 16, 1 TO 2, 1 TO 2)
DIM gmi(1 TO 30, 1 TO 20)
DIM id(1 TO 20, 1 TO 20)
DIM w(0 TO 20, 1 TO 1)
DIM spec(1 TO 2)
DIM lf(1 TO 30, 1)
DIM holda(1 TO 30, 1 TO 30)
DIM holdb(1 TO 20, 1 TO 30)
DIM holdc(1 TO 20, 1 TO 30)
DIM holddd(1 TO 20, 1 TO 30)
DIM holde(1 TO 20, 1 TO 30)
DIM holdf(1 TO 20, 1 TO 30)
DIM gt(1 TO 20, 1 TO 30)

```

```

DIM dmpo2f(-20 TO 0)
DIM rad(0 TO 50)
DIM loadx(1 TO 21)

FOR i% = 0 TO 21
  molfo2(i%) = .21
  molfo2o(i%) = molfo2(i%)
NEXT i%
scm% = 0
limit% = 0
dmpo2f(0) = 0
swing% = 0
swingper = -3.14159 / 2
s% = 1

rannum = 1
RANDOMIZE (rannum)

FOR i% = 4 TO 13
  bedc(i%) = 0
  bedash(i%) = 0
  smass(i%) = 10
NEXT i%

aistan = .0182
saistan = .1197

two = 273 + 65
twell = 273 + 600
waterfl = .08
twi = 273 + 4
aom = .155965
ntubes% = 2

o2cont% = 0
setpo2 = 3.5

testcnta% = 0
countint% = 0
warmup% = 0
adapt% = 0

```

```

nc% = 15
numout% = 2
numin% = 2
FOR i% = 1 TO 20
  FOR j% = 1 TO 20
    IF i% = j% THEN
      id(i%, j%) = 1
    ELSE
      id(i%, j%) = 0
    END IF
  NEXT j%
NEXT i%

IF times% = 1 THEN GOTO 675

nu% = 1
alph(1) = 0
alph(2) = 0
FOR i% = 1 TO 20
  FOR j% = 1 TO 2
    lam(i%, j%) = 0
  NEXT j%
NEXT i%

' FOR i% = 1 TO 20
'   lam(i%, 2) = 9.999999999999999D+32
' NEXT i%

FOR i% = -5 TO 0
  FOR j% = 1 TO 2
    y(i%, j%) = 0
    u(i%, j%) = 0
  NEXT j%
NEXT i%

cc(1) = -1.6
cc(2) = .64
cc(3) = 0
cc(4) = 0
a(1, 1) = -.855
a(2, 1) = 0
a(3, 1) = 0
a(1, 2) = 0
a(2, 2) = 0

```

```

b(0, 1, 1) = 1.688
b(1, 1, 1) = .198
b(2, 1, 1) = .193
b(3, 1, 1) = .122
b(0, 1, 2) = 0
b(1, 1, 2) = 0
b(2, 1, 2) = 0
b(3, 1, 2) = 0
b(0, 2, 1) = 0
b(1, 2, 1) = 0
b(2, 2, 1) = 0
b(3, 2, 1) = 0
b(0, 2, 2) = 0
b(1, 2, 2) = 0
b(2, 2, 2) = 0
b(3, 2, 2) = 0

```

```

FOR i% = 1 TO 10
FOR j% = 1 TO 10
FOR lo% = 1 TO 2
  p(i%, j%, lo%) = id(i%, j%)
NEXT lo%
NEXT j%
NEXT i%

```

```
times% = 1
```

```
675
```

```

' Initializations from rhodes bas
gs = 100
acration = 15
setpload = 120
mug = .0000442
rhog = .324
rhos = 2650
ebsmf = .44
lp = 3.6

```

```

ls = 3.8
g = 9.81
dcol = .1524
dp = .0005
y = .7
afactor = 1
acol = 3.14159 / 4 * dcol ^ 2
msolids = 100
k1 = 110000
k2 = .00001#
ncstr = 5

```

```
tg = 850 + 273
```

```
REM STANDPIPE
```

```

rstan = .05113
astan = 3.14159 * rstan * rstan
ROSTAN = .2
hstan = 5.3
AOSTAN = 2 * 3.14159 * ROSTAN * hstan
hostan = 27
vcell = astan * hstan / 10
hecll = hstan / 10
saicell = 2 * 3.14159 * rstan * hstan / 10
hgs = 150

```

```
REM CRITICAL PROPERTIES T(K), V(CM3/GMOL)
```

```

TCRITO2 = 154.6
TCRITN2 = 126.2
VCRITO2 = 73.4
VCRITN2 = 89.5

```

```
WMPO2 = 21
```

```

qout(1) = 1
qout(2) = 1
qout(3) = 1
DELTA1 = -1
DELTA2 = -1
testcnt% = 0
test% = 0

```



change = -1

#### REM RISER DIMENSIONS

```
REM H = 8
th = 14
h(1) = lp
h(2) = ls
h(3) = th - lp - ls
FOR i% = 1 TO 3
  rin(i%) = .086
  ar(i%) = 2 * 3.14159 * rin(i%) * h(i%)
NEXT i%
FOR i% = 4 TO 13
  rin(i%) = .05113
NEXT i%
```

```
aris = 3.14159 * rin(1) * rin(1)
qhx = 0
ss% = 0
r% = 0
```

#### REM INITIAL TEMPERATURES

```
T0 = 20 + 273
ta = 20 + 273
delt = 1
```

```
tai = 20 + 273
tg(0) = tai
FOR i% = 1 TO 3
  tg(i%) = 850 + 273
NEXT i%
FOR i% = 0 TO 13
  ts(i%) = 850 + 273
NEXT i%
press = 101.325
rg = 8.31434
```

#### REM INCREMENT DEFINITIONS

```
delr = .003
DEL = delr
NX% = 49
FOR i% = 0 TO NX%
  rad(i%) = rin + i% * DEL
NEXT i%
NX% = 49
count% = 0
minit = 0
```

#### REM SAND

```
hs = 121
hs = 50
cps = 800
smass = 100
TV = smass / rhos
v1 = 4 / 3 * 3.14159 * (dp / 2) ^ 3
sas = 3.14159 * dp ^ 2
NUMS = TV / v1
TAS = NUMS * sas
```

#### REM GAS

```
tairi = 211.3
psratio = 2
pair = tairi * psratio / (psratio + 1)
sair = tairi - pair
LIMEI = 0
CPG = 1100
mwg = 28
FO2 = .21
mw02 = 32
CCO2O = 0
```

#### REM AIR

```
ha = 12
MWAIR = 28.84
mwc = 12
mwh = 1
mwn = 14
```

mws = 32  
 mwo = 16  
 mwcl = 35.5  
 MWCAO3 = 100  
 mwh2o = 18  
 mwco2 = 44  
 MWSO2 = 64  
 mwn2 = 28  
 MWCAO = 56  
 MWCAO4 = 136  
 mwo2 = 32  
 MWCL2 = 71

kref = .3  
 cpref = 920  
 rhoref = 961

'j-paper  
 kj = .112  
 cpj = 1130  
 rhoj = 160

'carbon steel  
 kcs = 34  
 cpcs = 473  
 rhocs = 7801

'durablanket  
 kdur = .09  
 cpdur = 1130  
 rhodur = 64

'paroc  
 kpar = .08  
 cppar = 920  
 rhopar = 128

'stainless steel  
 kss = 26

cpss = 477  
 rhoss = 7900

# REM FUEL COMPOSITIONS (ULTIMATE ANALYSIS) REM PET. COKE (CONACO)

' PFC = 87.3  
 ' PFH = 3.8  
 ' PFN = 1.9  
 ' PFS = 4.9  
 ' PFO = 1.1  
 ' PFCL = .1  
 ' PFASH = .9  
 ' PFH2O = .5  
 ' rhoap = 1459  
 ' surfar = 255  
 ' order = 1  
 ' freq = 44.3946  
 ' ea = 4.99E+07

## REM HIGHVALE COAL

PFC = 62.4  
 PFH = 3.6  
 PFN = .8  
 PFS = .2  
 PFO = 18.7  
 PFCL = 0  
 PFASH = 14.3  
 PFH2O = 15.2  
 rhoap = 1185  
 surfar = 108100  
 order = .5  
 freq = 3.2664  
 ea = 5.699E+07

## REM ANTHRACITE COAL

' PFC = 77.5  
 ' PFH = 21  
 ' PFN = .9  
 ' PFS = .5  
 ' PFO = 2.4  
 ' PFCL = 0

```

' PFASH = 16.7
' PFH2O = 2.1
' rhoap = 1815
' surfar = 4210
' order = 0
' freq = 2.6935E-04
' ea = 7059000!

REM COAL
dc = .0009

ffr = 0

RG1 = 82.06
RG2 = 8319.17
mwc = 12
RHOC = 2000
mpart = 4 / 3 * 3.14159 * (dc / 2) ^ 3 * rhoap

REM CALCULATE HIGHERING VALUE OF FUEL (KJ/KGFUEL)
HHV = 1.055 / .4536 * (14544 * PFC / 100 + 62028 * (PFH / 100 - PFO /
100 / 8) + 4050 * PFS / 100)

bedc = 0
bedash = 0
REM SULPHUR CAPTURE (%)
sc = 0
RHOASH = 2700

GOSUB 9998

'read initial temperature distribution
FOR i% = 0 TO 49
INPUT #2, nt(i%, 1)
NEXT i%
FOR i% = 0 TO 49
FOR j% = 2 TO 13
nt(i%, j%) = nt(i%, 1)

```

---

```

NEXT j%
NEXT i%

o2concs = .21 * .010389

REM BEGIN CALCULATIONS
11

TF minit + count% / 60 > 100 THEN INPUT z%

TF minit + count% / 60 > 110 THEN
' ffr = ffr - .1
' IF ffr < 13.65 THEN ffr = 13.65
' coaly = ffr
'END IF

twi = 273 + 9.5

TF minit + count% / 60 > 1000 THEN STOP

STOP

IF hlhx > 0 THEN
r% = 1
ntubes% = 1
waterfl = .2
END IF

IF minit + count% / 60 > .5 THEN ss% = 1

GOTO 610

rfr = 26.5
coaly = rfr

IF rfr - ffr < 0 THEN
ffr = ffr - .1
IF ffr < rfr THEN ffr = rfr

```

```

coaly = ffr
ELSE
  ffr = ffr + .1
  IF ffr > rfr THEN ffr = rfr
  coaly = ffr
END IF
610
'GOTO 612
ffrhi = 28
ffrlow = 25
IF ABS(ffr - coaly) < .1 THEN
  ffr = coaly
ELSE
  ffr = ffr - .1 * ABS(ffr - coaly) / (ffr - coaly)
END IF
612
TF minit + count% / 60 < 65 THEN
  ffr = ffr + .1
  IF ffr > 30 THEN ffr = 30
  coaly = ffr
END IF
'dc = .0001
GOTO 611
IF minit + count% / 60 > 247 THEN
  IF countint% = 0 THEN
    INPUT #3, coaly
    coaly = coaly * 1.2156
    IF minit + count% / 60 > 236 THEN
      coaly = coaly / 1.16 * 1.175
    END IF
  END IF
IF ABS(ffr - coaly) < .1 THEN
    ffr = coaly
  ELSE
    ffr = ffr - .1 * ABS(ffr - coaly) / (ffr - coaly)
  END IF
  IF minit + count% / 60 > 200 THEN
    ffr = ffr + .1
    IF ffr > 14.4 THEN ffr = 14.4
  ELSE
    ffr = ffr + .1
    IF ffr > 15.4 THEN ffr = 15.4
  END IF
END IF
611
TF tg(1) > 1125 AND r% = 1 THEN
  warmup% = 1
  cont% = 1
END IF
TF count% > 30 THEN SS% = 1
TF count% > 45 THEN
  ffr = 21.5
  tai = 20 + 273
  IF hlhx / 3600 > -10 THEN
    r% = 1
    waterfl = 0
    ntubes% = 1
    setpnt(1) = 860
    warmup% = 1
    cont% = 1
    adapt% = 1
  END IF
END IF

```

```

y = 10 ^ (-1.5446 + 2.9162 * w - 1.0432 * w ^ 2)
ELSE
y = 10 ^ (-1.64758 + 2.94786 * w - 1.09703 * w ^ 2 + .17129 * w ^ 3)
END IF
END IF
vt = y * b2

FOR i% = -15 TO -1
dmpo2f(i%) = dmpo2f(i% + 1)
NEXT i%
pair = tairi * psratio / (psratio + 1)
sair = tairi - pair

count% = count% + 1
IF count% >= 60 THEN
count% = 0
minit = minit + 1
IF swing% = 1 THEN
swingper = swingper + 3.14159 / 120
waterfl = (.035 + .005 * SIN(swingper)) / ntubes%
IF waterfl * ntubes% > .04 THEN swingper = (.03 - .04) * .01
IF waterfl * ntubes% < .03 THEN swingper = (.04 - .03) * .01
END IF
END IF
STORE% = STORE% + 1

' perform hydrodynamic calculations
IF o2cont% = 1 THEN
erroro2 = setpo2 - dmpo2f(-15)
tairi = tairi + .01 * erroro2
IF tairi > 225 THEN tairi = 225
IF tairi < 150 THEN tairi = 150
pair = tairi * psratio / (1 + psratio)
IF pair < 85 THEN pair = 85
sair = tairi - pair
END IF

h = lp + ls
GOSUB 4000
h = th

```

```

tpart1 = tg(1) + 100
'IF tg(1) < (849.38 + 273) THEN
' ffr = 26
' ELSE
' ffr = 25
' END IF
IF minit + count% / 60 > 1300 THEN
INPUT z%
END IF

' calculate gas density (kg/m3)
rhog = 1.29425 * 273 * press / 101.325 / tg(1)

' calculate gas viscosity (kg/m/s)
mug = 6.31E-06 + tg(1) * 4.18E-08 - tg(1) ^ 2 * 6.7E-12

' calculate gas thermal conductivity (W/m/K)
kg = .001 * (19.178 + .047567 * tg(1))

' estimate molecular diffusivity (m2/s)
dab = 9.693E-06 * (tg(1) + 100) ^ 1.75 / 100 ^ 2

' schmit number
sc = mug / rhog / dab

' calculate single particle terminal velocity
b1 = (rhog * (rhos - rhog) * 9.81 / mug ^ 2) ^ (1 / 3)
b2 = (mug * 9.81 * (rhos - rhog) / rhog ^ 2) ^ (1 / 3)
x = dp * b1
IF x <= 3.8 THEN
y = x ^ 2 / 18 - 3.1234E-04 * x ^ 5 + 1.6415E-06 * x ^ 8 - 7.278E-10 * x ^ 11
ELSE
w = LOG(x) / LOG(10)
IF x < 7.58 THEN

```

```

12
interval% = 120

REM MASS BALANCE AROUND REACTOR (COMBUSTION
CALCULATIONS)
GOSUB 6000

REM HEAT BALANCE
GOSUB 7000

s% = 3
IF scm% = 0 THEN
    LOCATE 1, 1: PRINT "LOAD", load(1), load(2), load(3)
    LOCATE 1, 62: PRINT "TIME = "
    LOCATE 1, 76: PRINT "MIN."
    LOCATE 1, 69: PRINT USING "####.#", minit + count% / 60
    LOCATE 2, 69: PRINT count%
    LOCATE 5, 41: PRINT "GAS"
    LOCATE 6, 40: PRINT USING "####.#", tg(1) - 273

    LOCATE 5, 47: PRINT "SAND"
    LOCATE 6, 47: PRINT USING "####.#", ts(1) - 273

    LOCATE 5, 60: PRINT "SUP. GAS VEL."
    LOCATE 6, 60: PRINT "(M/S)"
    LOCATE 7, 60: PRINT USING "##.##", rg * tg(1) / press / aris / mwg * tairi
/ 3600

REM GOTO 987
LOCATE 4, 30: PRINT waterfl * ntubes%
LOCATE 5, 30: PRINT two - 273
LOCATE 6, 30: PRINT loadmem
LOCATE 7, 30: PRINT twall - 273
LOCATE 8, 30: PRINT (1 - ebsdils) * 2650
LOCATE 9, 30: PRINT gs
LOCATE 10, 30: PRINT loadave

LOCATE 3, 30: PRINT "TEMPERATURE PROFILES (DEG. C)"
FOR i% = 0 TO NX% STEP 2
    LOCATE i% + 8 - i% / 2, 40: PRINT USING "####.", nt(3%, 1) - 273
    NEXT i%
    LOCATE 10, 40: PRINT USING "####.#", nt(0, 1) - 273
    LOCATE 11, 40: PRINT USING "####.#", nt(30, 1) - 273
    LOCATE 12, 40: PRINT USING "####.#", nt(31, 1) - 273
    LOCATE 13, 40: PRINT USING "####.#", nt(33, 1) - 273
    LOCATE 14, 40: PRINT USING "####.#", nt(41, 1) - 273
    LOCATE 15, 40: PRINT USING "####.#", nt(49, 1) - 273
    LOCATE 17, 40: PRINT USING "####.#", nt(0, 13) - 273
    LOCATE 18, 40: PRINT USING "####.#", nt(26, 13) - 273
987

LOCATE 3, 1: PRINT "FLUE GAS COMPOSITION"
LOCATE 4, 1: PRINT aeration, "aeration"
LOCATE 5, 1: PRINT dmpo2, "%O2"
LOCATE 6, 1: PRINT DMPCO2, "%CO2"
LOCATE 7, 1: PRINT dppmso2, "PPM SO2"
LOCATE 9, 1: PRINT "CARBON IN BED (%)"
LOCATE 10, 1: PRINT bedc(13) * 100
LOCATE 12, 1: PRINT "ASH IN BED (%)"
LOCATE 13, 1: PRINT bedash(13) * 100
LOCATE 8, 1: PRINT dmpo2ff(-15), "%o2"

LOCATE 15, 1: PRINT "COAL FEEDRATE (KG/HR)"
LOCATE 16, 1: PRINT ff
LOCATE 18, 1: PRINT "AIR FEEDRATE (KG/HR)"
LOCATE 19, 1: PRINT tairi
LOCATE 20, 1: PRINT tai - 273, "DEG. C"
LOCATE 9, 50: PRINT "HEAT REMOVED (KW)"
LOCATE 10, 60: PRINT hlhx / 3600
LOCATE 12, 50: PRINT "HEAT RELEASED (KW)"
LOCATE 13, 50: PRINT hgencom / 3600
LOCATE 13, 60: PRINT -hgen / 3600
LOCATE 15, 50: PRINT "HEAT TO REFRACTORY (KW)"
LOCATE 16, 60: PRINT hlris / 3600
LOCATE 18, 50: PRINT "HEAT TO SAND (KW)"
LOCATE 19, 60: PRINT HS * TAS * RSMASST / SMASST * tg(s%) - ts) * 3.6
LOCATE 19, 60: PRINT hlrsand / 3600
LOCATE 21, 50: PRINT "HEAT TO STANDPIPE (KW)"
LOCATE 22, 60: PRINT hlstan / 3600
REM PRINT TG, TS, NTS

```

```

REMARK INPUT X%
LOCATE 22, 30
IF ss% = 0 THEN
    PRINT "STEADY-STATE "
ELSE
    PRINT "UNSTEADY-STATE"
END IF
LOCATE 23, 30
IF r% = 0 THEN
    PRINT "RUN OFF"
ELSE
    PRINT "RUN ON "
END IF
' IF (KGCI - CARBO) <= 0 THEN GOTO 642
' LOCATE 22, 1: PRINT "COMBUSTION EFFICIENCY (%)"
' LOCATE 23, 1: PRINT (1 - (((load * aris * h) * bedc / (smass + bedc +
bedash) - CARBO) / (KGCI - CARBO))) * 100
642
ELSEIF scm% = 1 THEN
    LOCATE 1, 1: PRINT "LOAD", load(1), load(2), load(3)
    LOCATE 1, 58: PRINT "TIME ="
    LOCATE 1, 73: PRINT "MIN."
    LOCATE 1, 65: PRINT USING "###.##", minit + count% / 60
    LOCATE 11, 65: PRINT countint%
    LOCATE 12, 65: PRINT testcnt%
    LOCATE 13, 45: PRINT "trace =", trace
    LOCATE 14, 45: PRINT "lamrls=", lamrls
    LOCATE 2, 1: PRINT "TG"
    LOCATE 3, 1: PRINT a(1, 1)
    LOCATE 4, 1: PRINT a(2, 1)
    LOCATE 5, 1: PRINT a(3, 1)
    LOCATE 9, 1: PRINT "two"
    LOCATE 10, 1: PRINT a(1, 2)
    LOCATE 11, 1: PRINT a(2, 2)
    LOCATE 2, 20: PRINT "ffr->tg"
    LOCATE 3, 20: PRINT b(0, 1, 1)
    LOCATE 4, 20: PRINT b(1, 1, 1)
    LOCATE 5, 20: PRINT b(2, 1, 1)
    LOCATE 6, 20: PRINT b(3, 1, 1)
    LOCATE 2, 35: PRINT "load->tg"
    LOCATE 3, 35: PRINT b(0, 1, 2)
    LOCATE 4, 35: PRINT b(1, 1, 2)
    LOCATE 5, 35: PRINT b(2, 1, 2)
    LOCATE 6, 35: PRINT b(3, 1, 2)
    LOCATE 2, 50: PRINT "ffr->two"
    LOCATE 3, 50: PRINT b(0, 2, 1)
    LOCATE 4, 50: PRINT b(1, 2, 1)
    LOCATE 5, 50: PRINT b(2, 2, 1)
    LOCATE 6, 50: PRINT b(3, 2, 1)
    LOCATE 2, 65: PRINT "load->two"
    LOCATE 3, 65: PRINT b(0, 2, 2)
    LOCATE 4, 65: PRINT b(1, 2, 2)
    LOCATE 5, 65: PRINT b(2, 2, 2)
    LOCATE 6, 65: PRINT b(3, 2, 2)
    LOCATE 15, 1: PRINT "load =", loadave
    LOCATE 16, 1: PRINT "ffr =", ffr
    LOCATE 15, 25: PRINT "two =", two - 273
    LOCATE 16, 25: PRINT "tg =" LOCATE 16, 35: PRINT tg(1) - 273: LOCATE
16, 50: PRINT y(-1, 1) - y(-2, 1): LOCATE 16, 65: PRINT y(-3, 1)
    LOCATE 17, 25: PRINT "tgflit =" LOCATE 17, 35: PRINT yf(0, 1): LOCATE 17,
50: PRINT dyf(-1, 1): LOCATE 17, 65: PRINT dyf(-2, 1)
    LOCATE 18, 25: PRINT "dmpo2f =" dmpo2f(-15)
    LOCATE 19, 25: PRINT "dmpo2 =" dmpo2
    LOCATE 17, 1: PRINT "tairi =" tairi
    ELSEIF scm% = 2 THEN
        PRINT minit + count% / 60
    LOCATE 2, 1: PRINT minit + count% / 60
    GOSUB 5400
    INPUT z%
    LOCATE 1, 1: PRINT "LOAD", load
    LOCATE 1, 32: PRINT "TIME ="
    LOCATE 1, 46: PRINT "MIN."
    LOCATE 1, 39: PRINT USING "###.##", minit + count% / 60
    LOCATE 4, 30: PRINT dmpo2f(-15)
    PRINT minit + count% / 60

```

```

STOP
END IF
testcnta% = testcnta% + 1
IF testcnta% >= 8 THEN
    number = RND -.5
    IF number >= 0 THEN
        coaly = ffrhi
    ELSE
        coaly = ffrlow
    END IF
    testcnta% = 0
END IF

number = RND -.5
IF number >= 0 THEN
    tairi = 205
ELSE
    tairi = 195
END IF
pair = tairi * psratio / (psratio + 1)
sair = tairi - pair
END IF
END IF

WRITE #1, minit + count% / 60, ffr, tairi, loadave, dmpo2ff(-15), tg(1) - 273, two -
273, aeration, waterfl * ntubes%, a(1, 1), b(0, 1, 1), b(2, 1, 1), b(3, 1, 1)
' a(1, 1), a(2, 1), b(0, 1, 1), b(1, 1, 1), b(2, 1, 1), a(2, 1), a(1, 2), a(2, 2), b(0, 1, 1),
' DMPO2, BEDC, ffr, aeration, load, a(1, 1), a(2, 1), a(1, 2), b(2, 1, 2), b(3, 1, 2), b(0, 2,
b(1, 1, 1), b(2, 1, 1), b(3, 1, 1), b(0, 1, 2), b(1, 1, 2), b(2, 1, 2), b(3, 1, 2), b(0, 2,
1), b(1, 2, 1), b(2, 2, 1), b(3, 2, 1), b(0, 2, 2), b(1, 2, 2), b(2, 2, 2), b(3, 2, 2)

countint% = 0
IF swing% = 1 THEN
    setpmt(1) = (waterfl * ntubes% - .03) / (.04 - .03) * (205 - 125) + 125
END IF
GOSUB 9365
END IF

IF limit% = 1 THEN
    IF tg(1) < 1125 THEN
        ffr = 24
    
```

```

IF STORE% = INT(60 / DELT) THEN
END IF
IF minit > 1999 THEN
FOR s% = 4 TO 13
FOR i% = 0 TO 26
WRITE #1, i%, nt(i%, s%)
NEXT i%
NEXT s%
WRITE #1, "*****"
INPUT z%
INPUT z%
INPUT z%
INPUT z%
INPUT z%

```

349



```

ELSE
  ffr = 19
  END IF
  END IF
  GOTO 11

9950
' subroutine for main menu
KEY 1, "comb"
KEY 2, ""
KEY 3, "limit"
KEY 4, "cont"
KEY 5, ""
KEY 6, "ans"
KEY 7, "coefs"
KEY 8, "blank"
KEY 9, ""
KEY 10, "quit"

KEY(1) ON
KEY(2) ON
KEY(3) ON
KEY(4) ON
KEY(5) ON
KEY(6) ON
KEY(7) ON
KEY(8) ON
KEY(9) ON
KEY(10) ON

ON KEY(1) GOSUB 9998
ON KEY(2) GOSUB 9990
ON KEY(3) GOSUB 9700
ON KEY(4) GOSUB 9960
ON KEY(5) GOSUB 9990
ON KEY(6) GOSUB 9951
ON KEY(7) GOSUB 9952
ON KEY(8) GOSUB 9953
ON KEY(9) GOSUB 9990
ON KEY(10) GOSUB 9999

KEY ON
RETURN

9951
  scm% = 0
  SCREEN 0
  CLS
  RETURN

9952
  scm% = 1
  SCREEN 0
  CLS
  RETURN

9953
  scm% = 2
  CLS
  SCREEN 0
  SCREEN 8
  GOSUB 5300
  RETURN

9960
' subroutine for control
KEY 1, "sp"
KEY 2, "nu"
KEY 3, "lam"
KEY 4, "adapt"
KEY 5, "on/off"
KEY 6, "alpha"
KEY 7, "swing"
KEY 8, "o2"
KEY 9, "two"
KEY 10, "main"

KEY(1) ON
KEY(2) ON
KEY(3) ON
KEY(4) ON
KEY(5) ON
KEY(6) ON
KEY(7) ON
KEY(8) ON
KEY(9) ON
KEY(10) ON

```

```

KEY(5) ON
KEY(6) ON
KEY(7) ON
KEY(8) ON
KEY(9) ON
KEY(10) ON

ON KEY(1) GOSUB 9961
ON KEY(2) GOSUB 9962
ON KEY(3) GOSUB 9963
ON KEY(4) GOSUB 9964
ON KEY(5) GOSUB 9965
ON KEY(6) GOSUB 9967
ON KEY(7) GOSUB 9969
ON KEY(8) GOSUB 9966
ON KEY(9) GOSUB 9968
ON KEY(10) GOSUB 9950

KEY ON
RETURN

9961
CLS
INPUT "setpoint for which output (1)=fg, (2)=two", x%
IF x% < 1 OR x% > 2 GOTO 9961
INPUT "new setpoint", setpnt(x%)
CLS
RETURN

9962
CLS
INPUT "new horizon", nu%
CLS
RETURN

9963
CLS
INPUT "constraint for which input (1)=ffr, (2)=acration", x%
IF x% > 2 THEN GOTO 9963
INPUT "new constraint factor", lambda
FOR i% = 1 TO 20

    lam(i%, x%) = lambda
    NEXT i%
    CLS
    RETURN

9964
CLS
IF adapt% = 1 THEN
    adapt% = 0
ELSE
    adapt% = 1
END IF
CLS
RETURN

9965
CLS
IF cont% = 1 THEN
    warmup% = 0
    cont% = 0
ELSE
    warmup% = 1
    cont% = 1
END IF
CLS
RETURN

9966
CLS
IF o2cont% = 1 THEN
    o2cont% = 0
ELSE
    o2cont% = 1
END IF
CLS
RETURN

9967
CLS
INPUT "input which output to constrain", xxxxx%
INPUT "alpha =?", alph(xxxxx%)

```

```

CLS
RETURN

9968 IF test1% = 1 THEN
    test1% = 0
ELSE
    test1% = 1
END IF
RETURN

9969 IF swing% = 1 THEN
    swing% = 0
ELSE
    swing% = 1
END IF

9990 REM SUBROUTINE FOR USELESS FUNCTION KEYS
RETURN

9991 REM SUBROUTINE TO CHANGE AIR INLET TEMPERAUR
CLS
INPUT "INPUT NEW GAS INLET TEMPERATURE (DEG. C)"; tai
tai = tai + 273
CLS
GOSUB 9998
RETURN

9992 REM SUBROUTINE TO CHANGE PRIMARY AIR FLOW RATE
CLS
1234 INPUT "CHANGE which air flow - primary(1) or secondary(2)?"; xxxxx%
INPUT "input new total air flow"; taiRI
pair = taiRI * psratio / (psratio + 1)
sair = taiRI - pair
CLS

CLS
RETURN

9993 REM SUBROUTINE TO CHANGE COAL FEEDRATE
CLS
INPUT "INPUT COAL FEEDRATE (KG/HR)"; ff
CLS
RETURN

9994 REM SUBROUTINE TO CHANGE TIME STEP
CLS
INPUT "INPUT NEW TIME STEP (S)"; delt
CLS
RETURN

9995 REM SUBROUTINE TO CHANGE HEAT REMOVAL
CLS
INPUT "INPUT NEW water flowrate"; waterfl
INPUT "number of tubes"; ntubes%
waterfl = waterfl / ntubes%
CLS
RETURN

9996 REM SUBROUTINE TO TOGGLE STEADY AND UNSTEADY STATE
CALCULATIONS
IF ss% = 0 THEN
    ss% = 1
ELSE
    ss% = 0
END IF
RETURN

9997 REM SUBROUTINE TO TOGGLE CONST. TEMP AND CALCULATED GAS
TEMP

```

```

IF r% = 0 THEN
  r% = 1
ELSE
  r% = 0
END IF
RETURN

9988 IF test% = 0 THEN
  test% = 1
ELSE
  test% = 0
END IF
TEST1 = 1
RETURN

9989 CLS
INPUT "input new load", setpload
RETURN

9998 REM SUBROUTINE TO SET UP FUNCTION KEYS
REM FUNCTION KEY TEMPLATE

KEY 1, "GASIT"
KEY 2, "GASF"
KEY 3, "COAL"
KEY 4, "TIME"
KEY 5, "HEAT"
KEY 6, "STATE"
KEY 7, "RUN"
KEY 8, "TEST"
KEY 9, "circ"
KEY 10, "main"

KEY(1) ON
KEY(2) ON
KEY(3) ON

KEY(4) ON
KEY(5) ON
KEY(6) ON
KEY(7) ON
KEY(8) ON
KEY(9) ON
KEY(10) ON

ON KEY(1) GOSUB 9991
ON KEY(2) GOSUB 9992
ON KEY(3) GOSUB 9993
ON KEY(4) GOSUB 9994
ON KEY(5) GOSUB 9995
ON KEY(6) GOSUB 9996
ON KEY(7) GOSUB 9997
ON KEY(8) GOSUB 9988
ON KEY(9) GOSUB 9989
ON KEY(10) GOSUB 9950

KEY ON
RETURN

9999

STOP
END

6000 REM MASS BALANCE

FOR s% = 13 TO 4 STEP -1
  bedc(s%) = bedc(s%) + delt * gs * astan / vcell / rhos / ebsmf * (bedc(s% - 1) -
  bedc(s%))
  bedash(s%) = bedash(s%) + delt * gs * astan / vcell / rhos / ebsmf * (bedash(s%
  - 1) - bedash(s%))

```

```

NEXT s%
bedc(0) = bedc(13)
bedash(0) = bedash(13)

kmc2o(0) = 0
s% = 1
REM  CALCULATE INLET FLOWS (KG) OVER TIME DELT

water = (1 - PFH2O / 100)
knci = ffr / 3600 * water * PFC / 100 / mwc + gs * aris * bedc(13) / mwc
kgco(0) = knci * mwc
kmhi = ffr / 3600 * water * PFH / 100 / mwH
kmni = pair / 3600 / MWAIR * .79 * 2 + (ffr / 3600 * water * PFN / 100) /
mwN
KMSI = ffr / 3600 * water * PFS / 100 / mws
knnoi = pair / 3600 / MWAIR * .21 * 2 + (ffr / 3600 * water * PFO / 100) /
mwO
KMCLI = ffr / 3600 * water * PFCL / 100 / mwcl
KMASHI = ffr * water * PFASH / 100 * TIME + (load(s%) * aris * h(s%) -
gs * aris) * bedash(s%) + gs * aris * bedash(s% - 1)
KMCACO3I = LIMEI * TIME
KMH2OI = ffr / 3600 * PFH2O / 100 / mwH2O
kgashi = ffr / 3600 * water * PFASH / 100

REM  COMBUSTION KINETICS CALCULATIONS
REM  OUTLET KMOL

REM  SPECIFIC VOLUME OF GAS AT T AND P(KMOL/M3)
sv = press / tg(s%) / rg

kgciz = bedc(s%) * load(s%) * 3.14159 * .086 ^ 2 * lp + knci * mwc - gs *
3.14159 * .086 ^ 2 * bedc(s%)
nbedc = kgciz / (load(s%) * 3.14159 * .086 ^ 2 * lp)

FOR i% = 1 TO 10

kgciz = nbedc * loadx(i%) * 3.14159 * .086 ^ 2 * .36
nunc = kgciz / mpart

gasvel = rg * tg(1) / press / aris / mwg * pair / 3600 / (1 - loadx(i%) / 2650)

' particle reynolds number
repp = rhog * (gasvel - gs / loadx(i%)) * dc / mug

' sherwood number
sh = 2 * (1 - loadx(i%) / 2650) + .69 * (repp / (1 - loadx(i%) / 2650)) ^ (1 / 2)
* sc ^ (1 / 3)

' mass transfer coefficient (m/s)
kd = sh * dab / dc

' estimate kinetic rate constant (kmol/m2/s/(kmol/m3)^n)
kcr = freq * EXP(-ea / 8314 / tpart1)
LOCATE 1, 2: PRINT i%
6001  molfo2o(i%) = molfo2(i%)
o2concb = molfo2o(i%) * sv
o2concs = o2concb

' determine surface concentration of oxygen (kmol/m3)
891

termy = kcr * o2concs ^ order - kd * (o2concb - o2concs)
IF ABS(termy) < .000001# GOTO 890
o2concs = o2concs - .01 * termy

GOTO 891
890
rate as(kmol/s)
crate = kcr * o2concs ^ order * 4 * 3.14159 * (dc / 2) ^ 2 * nunc

IF i% = 1 THEN
kmh2oo(s%) = kmhi / 2 + KMH2OI
kmso2o(s%) = KMSI * (1 - sc / 100)
kmn2o(s%) = kmni / 2
KMCASO4O(s%) = KMSI * sc / 100
KMCAOO(s%) = KMCACO3I - KMCASO4O(s%)

```

```

kmcl2o(s%) = KMCLI / 2
END IF

IF i% = 1 THEN
  kmo2i = (kmoi - 2 * kmsso2o - kmhi / 2) / 2
  kmo2o(i%) = kmo2i - crate
  kmco2o(i%) = crate + KMCACO3I
ELSE
  kmo2i = kmo2o(i% - 1)
  kmo2o(i%) = kmo2i - crate
  kmco2o(i%) = kmco2o(i% - 1) + crate
END IF

molfo2(i%) = kmo2o(i%) / (kmo2o(i%) + kmco2o(i%) + kmcl2o(s%) +
kmn2o(s%) + kmh2oo(s%) + kmsso2o(s%))
IF molfo2(i%) < 0 THEN
  molfo2(i5) = 0
  kmo2o(i%) = 0
  GOTO 6022
END IF

IF ABS(1 - molfo2(i%) / molfo2o(i%)) > .000001 THEN
  molfo2(i%) = (molfo2(i%) + molfo2o(i%)) / 2
  GOTO 6001
END IF

REM FLUE GAS COMPOSITION (WET)(MOL%)
6022 TOTAL = kmn2o(s%) + kmo2o(i%) + kmco2o(i%) + kmsso2o(s%) +
kmh2oo(s%)
WMPN2 = kmn2o(s%) / TOTAL * 100
WMPO2 = kmo2o(i%) / TOTAL * 100
WMPCO2 = kmco2o(i%) / TOTAL * 100
IF WMPO2 = 21 THEN
  WPPMSO2 = 0
ELSE
  WPPMSO2 = kmsso2o(s%) / TOTAL * 1000000 * (21 - 3) / (21 - WMPO2)
END IF

WMPH2O = kmh2oo(s%) / TOTAL * 100

REM FLUE GAS COMPOSITION (DRY)(MOL%)
TOTAL = kmn2o(s%) + kmo2o(i%) + kmco2o(i%) + kmsso2o(s%)
DMPN2 = kmn2o(s%) / TOTAL * 100
dmpo2 = kmo2o(i%) / TOTAL * 100
DMPCO2 = kmco2o(i%) / TOTAL * 100
IF dmpo2 = 21 THEN
  dppmsso2 = 0
ELSE
  dppmsso2 = kmsso2o(s%) / TOTAL * 1000000 * (21 - 3) / (21 - dmpo2)
END IF

kgciz = kgciz - (kmco2o(i%) - kmco2o(i% - 1)) * mwc
nbedc = kgciz / (load(s%) * 3.14159 * .086 ^ 2 * lp)
PRINT i%, kgciz, molfo2o(i%)

NEXT i%

*****
*
kmco(s%) = kgciz / mwc

REM OUTLET MASS (KG)
kgasho = kgashi

BEDCO = bedc(s%)

bedc(s%) = kmco(s%) * mwc / (load(s%) * 3.14159 * .086 ^ 2 * lp)
bedash(s%) = kgasho / (load(s%) * 3.14159 * .086 ^ 2 * lp)

s% = 2

REM CALCULATE INLET FLOWS (KG) OVER TIME DELT

```

```

'determine surface concentration of oxygen (kmol/m3)
893
termy = kcr * o2concs ^ order - kd * (o2concb - o2concs)
IF ABS(termy) < .000001# GOTO 892
o2concs = o2concs - .01 * termy

GOTO 893

892
rate as(kmol/s)
crate = kcr * o2concs ^ order * 4 * 3.14159 * (dc / 2) ^ 2 * numc

kmo2i = kmo2o(i% - 1)
kmo2o(i%) = kmo2i - crate
kmco2o(i%) = kmco2o(i% - 1) + crate

molfo2(i%) = kmo2o(i%) / (kmo2o(i%) + kmco2o(i%) + kmcl2o(s% - 1) +
kmn2o(s%) + kmh2oo(s% - 1) + kmso2o(s% - 1))

IF ABS(1 - molfo2(i%) / molfo2o(i%)) > .000001 THEN
molfo2(i%) = (molfo2(i%) + molfo2o(i%)) / 2
GOTO 6002
END IF

REM FLUE GAS COMPOSITION (WET)(MOL%)
TOTAL = kmn2o(s%) + kmo2o(i%) + kmco2o(i%) + kmso2o(s% - 1) +
kmh2oo(s% - 1)
WMPN2 = kmn2o(s%) / TOTAL * 100
WMPO2 = kmo2o(i%) / TOTAL * 100
WMPCO2 = kmco2o(i%) / TOTAL * 100
IF WMPO2 = 21 THEN
WPPMSO2 = 0
ELSE
WPPMSO2 = kmso2o(s%) / TOTAL * 1000000 * (21 - 3) / (21 - WMPO2)
END IF
WMPH2O = kmh2oo(s% - 1) / TOTAL * 100

REM FLUE GAS COMPOSITION (DRY)(MOL%)
TOTAL = kmn2o(s%) + kmo2o(i%) + kmco2o(i%) + kmso2o(s% - 1)

```

```

kgci = gs * 3.14159 * .086 ^ 2 * bedc(s% - 1)
kmni = kmn2o(1) * 2 + sair / 3600 / MWAIR * .79 * 2
kmoi = kmo2o(10) * 2 + sair / 3600 / MWAIR * .21 * 2
kgasho = kgasho
kmo2o(10) = kmoi / 2
kmn2o(s%) = kmni / 2

REM COMBUSTION KINETICS CALCULATIONS
REM SPECIFIC VOLUME OF GAS AT T AND P(KMOL/M3)
sv = press / tg(s%) / rg

kgciz = bedc(s%) * load(s%) * 3.14159 * .086 ^ 2 * ls + kgci - gs * 3.14159 *
.086 ^ 2 * bedc(s%)
nbedc = kgciz / (load(s%) * 3.14159 * .086 ^ 2 * ls)

FOR i% = 11 TO 20
kgciz = nbedc * loadx(i%) * 3.14159 * .086 ^ 2 * .38
numc = kgciz / mpart

gasvel = rg * tg(1) / press / aris / mwg * tairi / 3600 / (1 - loadx(i%) / 2650)
' particle reynolds number
repp = rhog * (gasvel - gs / loadx(i%)) * dc / mug

' sherwood number
sh = 2 * (1 - loadx(i%) / 2650) + .69 * (repp / (1 - loadx(i%) / 2650)) ^ (1 / 2)
* sc ^ (1 / 3)

' mass transfer coefficient (m/s)
kd = sh * dab / dc

' estimate kinetic rate constant (kmol/m2/s/(kmol/m3)^n)
ker = freq * EXP(-ea / 8314 / tpart1)
' LOCATE 1, 2: PRINT i%
6002 molfo2o(i%) = molfo2(i%)
o2concb = molfo2o(i%) * sv

```

```

DMPN2 = kmn2o(s%) / TOTAL * 100
dmpo2 = kno2o(i%) / TOTAL * 100
DMPCO2 = kmco2o(i%) / TOTAL * 100
IF dmpo2 = 21 THEN
  dppmso2 = 0
ELSE
  dppmso2 = kmso2o(s% - 1) / TOTAL * 1000000 * (21 - 3) / (21 - dmpo2)
END IF

kgciz = kgciz - (kmco2o(i%) - kmco2o(i% - 1)) * mwc
nbedc = kgciz / (load(s%) * 3.14159 * .086 ^ 2 * ls)
PRINT i%, kgciz, molfo2o(i%)

NEXT i%

kmco(s%) = kgciz / mwc

kgasho = kgashi
BEDCO = bedc(s%)

bedc(s%) = kmco(s%) * mwc / (load(s%) * 3.14159 * .086 ^ 2 * ls)
bedash(s%) = kgasho / (load(s%) * 3.14159 * .086 ^ 2 * ls)

s% = 3

REM  CALCULATE INLET FLOWS (KG) OVER TIME DELT

kgci = gs * 3.14159 * .086 ^ 2 * bedc(s% - 1)
kmni = kmn2o(2) * 2
kmnoi = kno2o(20) * 2
kgashi = kgasho

kmo2o(20) = kmnoi / 2
kmn2o(s%) = kmni / 2

REM  COMBUSTION KINETICS CALCULATIONS
REM  SPECIFIC VOLUME OF GAS AT T AND P(KMOL/M3)
sv = press / tg(s%) / rg
kgciz = bedc(s%) * load(s%) * vnl + kgci - gs * 3.14159 * .086 ^ 2 *
bedc(s%)
nbedc = kgciz / (load(s%) * vnl)
i% = 21
loadx(i%) = load(s%)

kgciz = nbedc * loadx(i%) * vnl
numc = kgciz / mpart

gasvel = rg * tg(1) / press / aris / mwg * tairi / 3600 / (1 - loadx(i%) / 2650)
' particle reynolds number
repp = rhog * (gasvel - gs / loadx(i%)) * dc / mug

' sherwood number
sh = 2 * (1 - loadx(i%) / 2650) + .69 * (repp / (1 - loadx(i%) / 2650)) ^ (1 / 2)
* sc ^ (1 / 3)

' mass transfer coefficient (m/s)
kd = sh * dab / dc

' estimate kinetic rate constant (kmol/m2/s/(kmol/m3)^n)
kcr = freq * EXP(-ea / 8314 / tpart1)

6003  molfo2o(i%) = molfo2(i%)
o2concb = molfo2o(i%) * sv

```



```

' determine surface concentration of oxygen (kmol/m3)
895
termy = kcr * o2concs ^ order - kd * (o2concb - o2concs)
IF ABS(termy) < .000001# GOTO 894
o2concs = o2concs - .01 * termy
IF o2concs < 0 THEN
o2concs = 0
GOTO 894
END IF
GOTO 895
894
rate as(kmol/s)
crate = kcr * o2concs ^ order * 4 * 3.14159 * (dc / 2) ^ 2 * numc

kmo2i = kmo2o(i% - 1)
kmo2o(i%) = kmo2i - crate
kmco2o(i%) = kmco2o(i% - 1) + crate

molfo2(i%) = kmo2o(i%) / (kmo2o(i%) + kmco2o(i%) + kmcl2o(s% - 2) +
kmn2o(s%) + kmh2oo(s% - 2) + kmso2o(s% - 2))
sick = ABS(1 - molfo2(i%) / molfo2o(i%))
IF ABS(1 - molfo2(i%) / molfo2o(i%)) > .001 THEN
molfo2(i%) = (molfo2(i%) + molfo2o(i%)) / 2
IF molfo2(i%) < 0 THEN molfo2(i%) = molfo2o(i%) / 2
GOTO 6003
END IF
896
REM FLUE GAS COMPOSITION (WET)(MOL%)
TOTAL = kmn2o(s%) + kmo2o(i%) + kmco2o(i%) + kmso2o(s% - 2) +
kmh2oo(s% - 2)
WMPN2 = kmn2o(s%) / TOTAL * 100
WMPO2 = kmo2o(i%) / TOTAL * 100
WMPCO2 = kmco2o(i%) / TOTAL * 100
IF WMPO2 = 21 THEN
WPPMSO2 = 0
ELSE
WPPMSO2 = kmso2o(s%) / TOTAL * 1000000 * (21 - 3) / (21 - WMPO2)
END IF
WMPH2O = kmh2oo(s% - 2) / TOTAL * 100

REM FLUE GAS COMPOSITION (DRY)(MOL%)
TOTAL = kmn2o(s%) + kmo2o(i%) + kmco2o(i%) + kmso2o(s% - 2)
DMPN2 = kmn2o(s%) / TOTAL * 100
dmpo2 = kmo2o(i%) / TOTAL * 100
DMPCO2 = kmco2o(i%) / TOTAL * 100
IF dmpo2 = 21 THEN
dppmsO2 = 0
ELSE
dppmsO2 = kmso2o(s% - 2) / TOTAL * 1000000 * (21 - 3) / (21 - dmpo2)
END IF

dmpo2al = .9
dmpo2f(0) = dmpo2f(-1) * dmpo2al + dmpo2 * (1 - dmpo2al)

kmco(s%) = kgciz / mwc - (kmco2o(21) - kmco2o(20))
kgasho = kgashi
BEDCO = bedc(s%)
bedc(s%) = kmco(s%) * mwc / (load(s%) * vnl)
bedashi(s%) = kgasho / (load(s%) * vnl)

RETURN
7000
hlris = 0
hlhx = 0
hlsand = 0
hngen = 0
hlstan = 0
hgencorn = 0
bigt = 1000
smallt = 750

```

ngto = tg

REM HEAT BALANCE

IF ss% = 1 THEN

s% = 1  
hg = 13 + 1.27 \* loadave + 4.735E-08 \* (tg(s%) ^ 4 - t(0, s%) ^ 4) / (tg(s%) - t(0, s%))

i% = 0

a0 = 2 \* 3.14159 \* rin(s%)

a1 = 2 + 3.14159 \* (rin(s%) + delr)

v = 3.14159 \* (delr ^ 2 + 2 \* delr \* rin(s%))

x = hg \* a0

y = kref \* a1 / 2 / delr

z = rhoref \* cpref \* v / 2 / delr

x = x / z

y = y / z

nt(i%, s%) = (ot(i%, s%) \* (1 - y / 2) + y \* t(i% + 2, s%) - x / 2 \* t(i%, s%) + x \* tg(s%)) / (1 + x / 2 + y / 2)

FOR i% = 1 TO 29

alphan = kref / rhoref / cpref

mm = delr ^ 2 / alphan / delr

aa = t(i% + 1, s%) \* (1 + delr / (2 \* (rin(s%) + i% \* delr))) + t(i% - 1, s%) \* (1 - delr / (2 \* (rin(s%) + i% \* delr)))

nt(i%, s%) = (mm - 2) / (mm + 2) \* ot(i%, s%) + 2 / (mm + 2) \* aa

NEXT i%

i% = 30

anm1 = 2 \* 3.14159 \* (rin(s%) + (i% - 1) \* delr)

anp1 = 2 \* 3.14159 \* (rin(s%) + (i% + 1) \* delr)

va = 3.14159 \* (2 \* rin(s%) \* i% \* delr + (i% \* delr) ^ 2 - 2 \* rin(s%) \* (i% - 1) \* delr - ((i% - 1) \* delr) ^ 2)

vb = 3.14159 \* (2 \* rin(s%) \* (i% + 1) \* delr + ((i% + 1) \* delr) ^ 2 - 2 \* rin(s%) \* i% \* delr - (i% \* delr) ^ 2)

w = kref \* anm1 / 2 / delr

x = rhoref \* cpref \* va / delr

y = kj \* anp1 / 2 / delr

z = rhoj \* cpj \* vb / delr

nt(i%, s%) = (w \* t(i% - 2, s%) + y \* t(i% + 2, s%) + t(i%, s%) \* (x - w - y + z)) / (z + x)

i% = 31

anm1 = 2 \* 3.14159 \* (rin(s%) + (i% - 1) \* delr)

anp1 = 2 \* 3.14159 \* (rin(s%) + (i% + 1) \* delr)

va = 3.14159 \* (2 \* rin(s%) \* i% \* delr + (i% \* delr) ^ 2 - 2 \* rin(s%) \* (i% - 1) \* delr - ((i% - 1) \* delr) ^ 2)

vb = 3.14159 \* (2 \* rin(s%) \* (i% + 1) \* delr + ((i% + 1) \* delr) ^ 2 - 2 \* rin(s%) \* i% \* delr - (i% \* delr) ^ 2)

w = kj \* anm1 / 2 / delr

x = rhoj \* cpj \* va / 2 / delr

y = kcs \* anp1 / 2 / delr

z = rhoes \* cpes \* vb / delr

nt(i%, s%) = (w \* t(i% - 2, s%) + y \* t(i% + 2, s%) + t(i%, s%) \* (x - w - y + z)) / (z + x)

i% = 32

alphan = kcs / rhoes / cpes

mm = delr ^ 2 / alphan / delr

aa = t(i% + 1, s%) \* (1 + delr / (2 \* (rin(s%) + i% \* delr))) + t(i% - 1, s%) \* (1 - delr / (2 \* (rin(s%) + i% \* delr)))

nt(i%, s%) = (mm - 2) / (mm + 2) \* ot(i%, s%) + 2 / (mm + 2) \* aa

i% = 33

anm1 = 2 \* 3.14159 \* (rin(s%) + (i% - 1) \* delr)

anp1 = 2 \* 3.14159 \* (rin(s%) + (i% + 1) \* delr)

va = 3.14159 \* (2 \* rin(s%) \* i% \* delr + (i% \* delr) ^ 2 - 2 \* rin(s%) \* (i% - 1) \* delr - ((i% - 1) \* delr) ^ 2)

vb = 3.14159 \* (2 \* rin(s%) \* (i% + 1) \* delr + ((i% + 1) \* delr) ^ 2 - 2 \* rin(s%) \* i% \* delr - (i% \* delr) ^ 2)

w = kcs \* anm1 / 2 / delr

x = rhoes \* cpes \* va / 2 / delr

y = kdur \* anp1 / 2 / delr

z = rhodur \* cpdur \* vb / delr

nt(i%, s%) = (w \* t(i% - 2, s%) + y \* t(i% + 2, s%) + t(i%, s%) \* (x - w - y + z)) / (z + x)

FOR i% = 34 TO 40

alphan = kdur / rhodur / cpdur

```

mm = delr ^ 2 / alphar / delt
aa = t(i% + 1, s%) * (1 + delr / (2 * (rin(s%) + i% * delr))) + t(i% - 1, s%) * (1
- delr / (2 * (rin(s%) + i% * delr)))
nt(i%, s%) = (mm - 2) / (mm + 2) * ot(i%, s%) + 2 / (mm + 2) * aa
NEXT i%

i% = 41
ann1 = 2 * 3.14159 * (rin(s%) + (i% - 1) * delr)
anp1 = 2 * 3.14159 * (rin(s%) + (i% + 1) * delr)
va = 3.14159 * (2 * rin(s%) * i% * delr + (i% * delr) ^ 2 - 2 * rin(s%) * (i% - 1) *
delr - ((i% - 1) * delr) ^ 2)
vb = 3.14159 * (2 * rin(s%) * (i% + 1) * delr + ((i% + 1) * delr) ^ 2 - 2 * rin(s%)
* i% * delr - (i% * delr) ^ 2)
w = kdur * ann1 / 2 / delr
x = rhodur * cpdur * va / 2 / delt
y = kpar * anp1 / 2 / delr
z = rhopar * cppar * vb / delt
nt(i%, s%) = (w * t(i% - 2, s%) + y * t(i% + 2, s%) + t(i%, s%) * (x - w - y + z)) /
(z + x)

FOR i% = 42 TO 48
  alphar = kpar / rhopar / cpar
  mm = delr ^ 2 / alphar / delt
  aa = t(i% + 1, s%) * (1 + delr / (2 * (rin(s%) + i% * delr))) + t(i% - 1, s%) * (1
- delr / (2 * (rin(s%) + i% * delr)))
  nt(i%, s%) = (mm - 2) / (mm + 2) * ot(i%, s%) + 2 / (mm + 2) * aa
NEXT i%

i% = 49
ann1 = 2 * 3.14159 * (rin(s%) + (i% - 1) * delr)
an = 2 * 3.14159 * (rin(s%) + i% * delr)
v = 3.14159 * (2 * rin(s%) * i% * delr + (i% * delr) ^ 2 - 2 * rin(s%) * (i% - 1) *
delr - ((i% - 1) * delr) ^ 2)
y = kpar * ann1 / 2 / delr
x = ha * an
z = rhopar * cpar * v / 2 / delt
x = x / z
y = y / z
nt(i%, s%) = (ot(i%, s%) * (1 - y / 2) - x / 2 * t(i%, s%) + y * t(i% - 2, s%) + x
* ta) / (1 + x / 2 + y / 2)

```

```

FOR s% = 4 TO 13
  i% = 0
  a0 = 2 * 3.14159 * rin(s%)
  a1 = 2 + 3.14159 * (rin(s%) + delr)
  v = 3.14159 * (delr ^ 2 + 2 * delr * rin(s%))
  x = hgs * a0
  y = kss * a1 / 2 / delr
  z = rhos * cpss * v / 2 / delt
  x = x / z
  y = y / z
  nt(i%, s%) = (ot(i%, s%) * (1 - y / 2) + y * t(i% + 2, s%) - x / 2 * t(i%, s%) + x *
ts(s%)) / (1 + x / 2 + y / 2)

  i% = 1
  ann1 = 2 * 3.14159 * (rin(s%) + (i% - 1) * delr)
  anp1 = 2 * 3.14159 * (rin(s%) + (i% + 1) * delr)
  va = 3.14159 * (2 * rin(s%) * i% * delr + (i% * delr) ^ 2 - 2 * rin(s%) * (i% - 1) *
delr - ((i% - 1) * delr) ^ 2)
  vb = 3.14159 * (2 * rin(s%) * (i% + 1) * delr + ((i% + 1) * delr) ^ 2 - 2 * rin(s%)
* i% * delr - (i% * delr) ^ 2)
  w = kss * ann1 / 2 / delr
  x = rhoss * cpss * va / delt
  y = kdur * anp1 / 2 / delr
  z = rhodur * cpdur * vb / delt
  nt(i%, s%) = (w * ts(s%) + y * t(i% + 2, s%) + t(i%, s%) * (x - w - y + z)) / (z + x)

  FOR i% = 2 TO 17
    alphar = kdur / rhodur / cpdur
    mm = delr ^ 2 / alphar / delt
    aa = t(i% + 1, s%) * (1 + delr / (2 * (rin(s%) + i% * delr))) + t(i% - 1, s%) * (1
- delr / (2 * (rin(s%) + i% * delr)))
    nt(i%, s%) = (mm - 2) / (mm + 2) * ot(i%, s%) + 2 / (mm + 2) * aa
  NEXT i%

  i% = 18
  ann1 = 2 * 3.14159 * (rin(s%) + (i% - 1) * delr)
  anp1 = 2 * 3.14159 * (rin(s%) + (i% + 1) * delr)

```

```

va = 3.14159 * (2 * rin(s%) * i% * delr + (i% * delr) ^ 2 - 2 * rin(s%) * (i% - 1) *
delr - ((i% - 1) * delr) ^ 2)
vb = 3.14159 * (2 * rin(s%) * (i% + 1) * delr + ((i% + 1) * delr) ^ 2 - 2 * rin(s%)
* i% * delr - (i% * delr) ^ 2)
w = kdur * ann1 / 2 / delr
x = rhodur * cpdur * va / delr
y = kpar * amp1 / 2 / delr
z = rhopar * cpper * vb / delr
nt(i%, s%) = (w * t(i% - 2, s%) + y * t(i% + 2, s%) + t(i%, s%) * (x - w - y + z)) /
(z + x)

FOR i% = 19 TO 25
    alphas = kpar / rhopar / cpper
    mm = delr ^ 2 / alphas / delr
    aa = t(i% + 1, s%) * (1 + delr / (2 * (rin(s%) + i% * delr))) + t(i% - 1, s%) * (1
- delr / (2 * (rin(s%) + i% * delr)))
    nt(i%, s%) = (mm - 2) / (mm + 2) * ot(i%, s%) + 2 / (mm + 2) * aa
NEXT i%

i% = 26
ann1 = 2 * 3.14159 * (rin(s%) + (i% - 1) * delr)
an = 2 * 3.14159 * (rin(s%) + i% * delr)
v = 3.14159 * (2 * rin(s%) * i% * delr + (i% * delr) ^ 2 - 2 * rin(s%) * (i% - 1) *
delr - ((i% - 1) * delr) ^ 2)
y = kpar * ann1 / 2 / delr
x = ha * an
z = rhopar * cpper * v / 2 / delr
x = x / z
y = y / z
nt(i%, s%) = (ot(i%, s%) * (1 - y / 2) - x / 2 * t(i%, s%) + y * t(i% - 2, s%) + x *
ta) / (1 + x / 2 + y / 2)

NEXT s%

END IF

REM HEAT LOSS THROUGH STANDPIPE (KJ/HR)
ts(3) = ts(1)
FOR s% = 4 TO 13
    ass1 = gs * astan * cps * 3.6

```

---

```

ass2 = hgs * saicell * 3.6
ts(s%) = (ass1 * ts(s% - 1) + ass2 * nt(0, s%)) / (ass1 + ass2)
histan = hstan + ass2 * (ts(s%) - nt(0, s%))

NEXT s%

ts(0) = ts(13)

s% = 1

hg = 13 + 1.27 * loadave + 4.735E-08 * (tg(s%) ^ 4 - t(0, s%) ^ 4) / (tg(s%) -
t(0, s%))

HR = hg

REM MASS OF SAND IN ZONE
rsmass = loadave * 3.14159 * rin(s%) ^ 2 * (h(1) + h(2) + h(3))

REM MASS OF SAND IN RISER AT NTS
mismass = circrat * delr

REM MASS OF SAND IN RISER AT TS
rsmass = rsmass - mismass

REM NEW SAND TEMPERATURE IN RISER
nts(s%) = rsmass / rsmass * ts(s%) + mismass / rsmass * ts(s% - 1)

REM CALCULATE HEAT OF REACTION AT 25 DEG. C (KJ/KGFUEL)
LOCATE 21, 25: PRINT HHV
LHV = HHV - PFH / 2.02 * 18 * 2370 / 100
HCOM = LHV

REM ENTHALPY OF AIR ENTERING (KJ/KGAIR)
REM REF T = 298 K

temp = tai
AI = 28.94
BI = .004147
CI = 3.191E-06
DI = -1.965E-09
GOSUB 8000

```

```

    hairi = dhi / MWAIR

    REM  ENTHALPY OF WATER ENTERING IN COAL (KJ/KGH2O)
    REM  GAS AT 25 DEG C
    REM  ASSUME FUEL ENTERS AT 25 DEG C THEREFORE
    IF s% = 1 THEN
        HH2OII = -2440
    ELSE HH2OII = 0
    END IF

    7001
    REM  ASSUME NO HEAT ASSOCIATED WITH SOLID FUEL FEED

    REM  ENTHALPY OF EXIT GASES (KJ/KG)
    REM  N2
        temp = tg(s%)
        AI = 29
        BI = .002119
        CI = 5.723E-06
        DI = -2.871E-09
        GOSUB 8000

        HN2O = dhi / mwN2

    REM  O2
        temp = tg(s%)
        AI = 29.1
        BI = .01158
        CI = -6.076E-06
        DI = 1.311E-09
        GOSUB 8000

        HO2O = dhi / mwO2

    REM  CO2
        temp = tg(s%)
        AI = 36.11

    BI = .04233
    CI = -2.887E-05
    DI = 7.464E-09
    GOSUB 8000

    HCO2O = dhi / mwCO2

    REM  H2O
        temp = tg(s%)
        AI = 33.46
        BI = .00688
        CI = 7.604E-06
        DI = -3.593E-09
        GOSUB 8000

        hh2oo = dhi / mwH2o

    REM  HEAT OF REACTION AT TEMPERATUREtg(s%)(KJ/HR)
        KGN2O(0) = 0
        KGO2O(0) = 0
        kgo2o(0) = 0
        KGH2OO(0) = 0
        IF s% = 1 THEN
            feedwat = ffr
            airin = pair + sair
        ELSEIF s% = 2 THEN
            feedwat = 0
            airin = sair
        ELSE
            feedwat = 0
            airin = 0
        END IF

        hrxn = -HCOM * kmco2o(21) * mwc * 100 / PFC * 3600
        hrxn = hrxn + (HN2O * kmn2o(3) * mwN2) * 3600
        hrxn = hrxn + (HO2O * kmo2o(21) * mwO2) * 3600
        hrxn = hrxn + (HCO2O * kmco2o(21) * mwCO2) * 3600
        hrxn = hrxn + (hh2oo * kmh2oo(1) * mwH2o) * 3600
        hrxn = hrxn - HH2OII * feedwat * PFH2O / 100

```

```
hrrxn = hrrxn - hairi * airin
```

```
' perform membrane wall calculations
```

```
IF r% = 0 GOTO 23
```

```
IF waterfl <= 0 THEN
```

```
    heatflow = 0
```

```
    GOTO 22
```

```
END IF
```

```
GOSUB 9500
```

```
22
```

```
    qhx = ntubes% * heatflow * 3.6
```

```
23
```

```
' correction for sand heat transfer calculation (see p.96330 of note book)
```

```
IF s% = 1 THEN
```

```
    ffra = ffr
```

```
ELSE
```

```
    ffra = 0
```

```
END IF
```

```
TERM1 = -hrrxn - qhx + hg * (ar(1) + ar(2) + ar(3)) * nt(0, s%) * 3.6 + 3.6 *
```

```
circrat * cps * nts(s%) + ffra * .7 * (273 + 20)
```

```
TERM2 = hg * (ar(1) + ar(2) + ar(3)) * 3.6 + 3.6 * circrat * cps + ffra * .7
```

```
ntg = TERM1 / TERM2
```

```
tg(s%) = ntg
```

```
IF r% = 0 THEN
```

```
    tg(s%) = 850 + 273
```

```
    ntgo = tg(s%)
```

```
    ntg = tg(s%)
```

```
END IF
```

```
IF r% = 0 THEN
```

```
    qhx = -hrrxn - hg * (ar(1) + ar(2) + ar(3)) * (tg(s%) - nt(0, s%)) * 3.6 - hs *
TAS * rsmass / smass * (tg(s%) - ts(s%)) * 3.6
    waterfl = qhx / 3.6 / 4175 / (two - twi)
END IF
```

```
IF (ABS(ntg - ntgo)) > .01 THEN
```

```
    tg(s%) = (tg(s%) + ntgo) / 2
```

```
    ntgo = tg(s%)
```

```
GOTO 7001
```

```
END IF
```

```
' compute sand temperature
```

```
qs = circrat * cps * (tg(s%) - nts(s%))
```

```
ts(s%) = (qs * delt / rsmass / cps) / 2 + ts(s%)
```

```
ts(s%) = tg(s%)
```

```
hlhx = qhx
```

```
hlris = hg * (ar(1) + ar(2) + ar(3)) * (tg(s%) - nt(0, s%)) * 3.6
```

```
hgen = hrrxn
```

```
hlisand = qs * 3600
```

```
hgencom = HCOM * kmco2α(21) * mwc * 100 / PFC * 3600
```

```
RETURN
```

```
8000
```

```
REM SUBROUTINE TO CALCULATE INTEGRAL CP, DELTAT (KJ/KMOL)
```

```
    dhi = AI * (temp - 298) + BI / 2 * temp * temp - BI / 2 * 298 * 298 + CI / 3 *
temp * temp * temp - CI / 3 * 298 * 298 * 298
```

```
REM + DI / 4 * TEMP * TEMP * TEMP - DI / 4 * 298 * 298 * 298 * 298 *
```

```
298
```

```

RETURN
4000
' SOLIDS DISTRIBUTION
' two controller
IF test1% = 1 THEN
    twoerror = 62 - (two - 273)
    setpload = setpload + twoerror
END IF

'solids loading controller
IF test1% = 0 THEN
    loaderr = setpload - loadd
    aeration = aeration + .02 * loaderr
END IF

up = rg * tg(1) / press / aris / mwg * pair / 3600
us = rg * tg(1) / press / aris / mwg * fairi / 3600
db = dcol
umf = dp ^ 2 * (rhos - rhog) * g / 1650 / mug

gsj = 23.4 - .61833333# * setpload + .00461111# * setpload ^ 2

13

' calculate riser pressure drop
ustar = up * (rhog ^ 2 / mug / 9.81 / (rhos - rhog)) ^ (1 / 3)
ebst = 1 - 2.1062698# * ustar ^ .2920853# * (gsj / rhos / (up - vt)) ^ .3661963#
gami = rhos * (1 - ebst)
fr = us / (9.81 * dcol) ^ .5
ftr = vt / (9.81 * dcol) ^ .5
ciggy = 1 + 5.6 / fr + .47 * fr ^ .41
x = ciggy * gsi / us / rhos
gamsusp = (1 - 1 / (1 + x)) * rhos * 1
sum = 0
count = 0
FOR z = .5 TO lp STEP .1
    count = count + 1
    sum = sum + gamsusp + (gami - gamsusp) * EXP(-2 * (z - .05))
NEXT z

NEXT z
count = count + 4
sum = sum + 4 * gami
load(1) = sum / count

sum = 0
count = 0
FOR z = lp + .1 TO lp + ls STEP .1
    count = count + 1
    sum = sum + gamsusp + (gami - gamsusp) * EXP(-2 * (z - .05))
NEXT z

load(2) = sum / count
loadave = (lp * load(1) + ls * load(2)) / (lp + ls)

IF ABS(setpload - loadave) > .1 THEN
    gsi = gsi + .1 * (setpload - loadave)
    GOTO 13
END IF

dpr = loadave * 9.81 * (lp + ls)

sum = 0
counts% = 1
loadx(1) = gami
FOR z = .72 TO lp + .36 STEP .36
    counts% = counts% + 1
    loadx(counts%) = gamsusp + (gami - gamsusp) * EXP(-2 * (z - .18))
NEXT z

FOR z = lp + .38 TO lp + ls + .38 STEP .38
    counts% = counts% + 1
    loadx(counts%) = gamsusp + (gami - gamsusp) * EXP(-2 * (z - .19))
' PRINT loadx(counts%), z - .19
NEXT z

```

```

b00 = 27641.76546# - 21151.71569# * psratio + 3539.172362# * psratio * psratio
b01 = -7471.419234# + 5728.169233# * psratio - 958.419231154# * psratio *
psratio
b02 = 502.316849812# - 385.721611718# * psratio + 64.554945054# * psratio *
psratio
b10 = -503.415904146# + 390.804747554# * psratio - 66.169934981# * psratio *
psratio
b11 = 136.252307686# - 105.935064098# * psratio + 17.93576923# * psratio *
psratio
b12 = -9.171672772999999# + 7.142643468# * psratio - 1.209340659# * psratio *
psratio
b20 = 2.120435345# - 1.664821864# * psratio + 285242521# * psratio * psratio
b21 = -.573995728# + 451262821# * psratio - 7.73247860000001D-02 * psratio
* psratio
b22 = .038638584# - .030420228# * psratio + .005213675# * psratio * psratio

b0 = b00 + b01 * us + b02 * us * us
b1 = b10 + b11 * us + b12 * us * us
b2 = b20 + b21 * us + b22 * us * us

gs = b0 + b1 * loadave + b2 * loadave * loadave

' Calculate cyclone pressure drop
delh = 24 * .1524 * .0635 / .1524 ^ 2 * (.3048 / .55245 / (.131445 - .55245)) ^ (1 /
3)
ug = us * .1524 ^ 2 / .1524 / .0635
dpcyc = rhog * ug ^ 2 * delh / 2

' Calculate pressure drop in horizontal section of l-valve
vsolids = gs * aris / astan / rhos / (1 - ebsmf)
rhobulk = rhos * (1 - ebsmf)
dphor = 4.266 * vsolids ^ .455 * .635 * rhobulk * 9.81 / (.1016 / dp) ^ .25

' Calculate pressure drop in l-valve bend
dpbend = 13.8 * vsolids ^ .34 * rhobulk / (.1016 / dp) ^ .25

dptot = dpr + dpcyc + dphor + dpbend

'PRINT dptot, dpr, dpcyc, dphor, dpbend
'input z%

'cyclone hydrodynamics
vnl = .09527
vs = .0072975#
gasres = 1 / 9.677399999999999D-03 / ug * (vs + vnl / 2)
rep = rhog * ug * dp / mug
vterm = (4 / 225 * (rhos * rhog) ^ 2 * 9.81 ^ 2 / rhog / mug) ^ (1 / 3) * dp
partres = .032 * gasres * rep ^ .43 * (ug - vterm) / vterm) ^ .7 * ((rhos - rhog) /
rhog) ^ .42 * .433289465#
cycmass = gs * aris * partres
'volume excluding vortex finder (m^3)
cycvol = .09527
load(3) = cycmass / cycvol

'surface area of cyclone (m^2)
cycarea = 1.1933

hstan = (msolids - loadave * aris * (lp + ls) - cycmass) / rhos / (1 - ebsmf) / astan

gsnew = (k1 * aeration * hstan - dptot) * k2

'PRINT gsnew, form, dpr, hstar

'input z%

'GOTO 13

30

loadd = loadave
loadmem = loadx(18)
circrat = gs * aris

RETURN

```



```

'
*****
*
9365
REM   GPC CONTROLLER

y(0, 1) = tg(1) - 273
yfi(0, 1) = y(0, 1) - cc(1) * yfi(-1, 1) - cc(2) * yfi(-2, 1) - cc(3) * yfi(-3, 1) - cc(4) * yfi(-
4, 1)
ufi(-1, 1) = u(-1, 1) - cc(1) * ufi(-2, 1) - cc(2) * ufi(-3, 1) - cc(3) * ufi(-4, 1) * cc(4) *
ufi(-5, 1)
dufi(-1, 1) = u(-1, 1) - u(-2, 1) - cc(1) * dufi(-2, 1) - cc(2) * dufi(-3, 1) - cc(3) * dufi(-
4, 1) - cc(4) * dufi(-5, 1)
dyfi(0, 1) = y(0, 1) - y(-1, 1) - cc(1) * dyfi(-1, 1) - cc(2) * dyfi(-2, 1) - cc(3) * dyfi(-3,
1) - cc(4) * dyfi(-4, 1)

spcc(1) = setpnt(1)
IF adapt% = 0 THEN GOTO 678

PERFORM RLS WITH VARIABLE FORGETTING FACTOT TO FIT
CONSTANTS
GOSUB 8100

678
cc(1) = -1.6
cc(2) = .64
cc(3) = 0
cc(4) = 0

IF warmup% = 0 THEN GOTO 689
IF nu% > 5 THEN nu% = 5
li% = 1
lo% = 1
e(0, 1, lo%) = 1
F(0, 1, lo%) = 1 - a(1, lo%) + cc(1)
F(1, 1, lo%) = a(1, lo%) - a(2, lo%) + cc(2)

F(2, 1, lo%) = a(2, lo%) - a(3, lo%) + cc(3)
F(3, 1, lo%) = a(3, lo%) + cc(4)
FOR j% = 1 TO nc%
  F(0, j% + 1, lo%) = F(1, j%, lo%) - (a(1, lo%) - 1) * F(0, j%, lo%)
  F(1, j% + 1, lo%) = F(2, j%, lo%) - (a(2, lo%) - a(1, lo%)) * F(0, j%, lo%)
  F(2, j% + 1, lo%) = F(3, j%, lo%) - (a(3, lo%) - a(2, lo%)) * F(0, j%, lo%)
  F(3, j% + 1, lo%) = a(3, lo%) * F(0, j%, lo%)
  FOR i% = 0 TO j% - 1
    e(i%, j% + 1, lo%) = e(i%, j%, lo%)
    NEXT i%
    e(j%, j% + 1, lo%) = F(0, j%, lo%)
    NEXT j%
    NEXT lo%

  FOR lo% = 1 TO numout%
    ' FOR li% = 1 TO numin%
      FOR i% = 0 TO 3
        g(i%, 1, lo%, li%) = b(i%, lo%, li%)
        NEXT i%
        NEXT li%
      FOR j% = 1 TO nc%
        ' FOR li% = 1 TO numin%
          FOR k% = 0 TO j% + 3
            g(k%, j% + 1, lo%, li%) = 0
            NEXT k%
            FOR i% = 0 TO j%
              FOR k% = 0 TO 3
                g(i% + k%, j% + 1, lo%, li%) = g(i% + k%, j% + 1, lo%, li%) +
e(i%, j% + 1, lo%) * b(k%, lo%, li%)
                NEXT k%
                NEXT i%
              ' NEXT li%
              NEXT j%
            NEXT lo%
          FOR i% = 0 TO 3
            gamma(i%, 0) = 0
            gp(0) = 0
            NEXT i%

          FOR j% = 0 TO nc%
            gp(j% + 1) = (e(j%, j% + 1, 1) * b(0, 1, 1) + gamma(0, j%))

```

```

        gamma(0, j% + 1) = gamma(1, j%) + e(j%, j% + 1, 1) * b(1, 1, 1) - cc(1) *
        gp(j% + 1)
        gamma(1, j% + 1) = gamma(2, j%) + e(j%, j% + 1, 1) * b(2, 1, 1) - cc(2) *
        gp(j% + 1)
        gamma(2, j% + 1) = gamma(3, j%) + e(j%, j% + 1, 1) * b(3, 1, 1) - cc(3) *
        gp(j% + 1)
        gamma(3, j% + 1) = -cc(4) * gp(j% + 1)
    NEXT j%

    FOR i% = 1 TO nc%
        FOR j% = 1 TO nc%
            IF j% > i% THEN
                gm(i%, j%) = 0
            ELSE
                gm(i%, j%) = gp(i% - j%)
            END IF
        NEXT j%
    NEXT i%

    FOR i% = 1 TO nc%
        id(i%, i%) = id(i%, i%) * lam(i%, 1)
    NEXT i%

    CALL mattran(nc%, nu%, gm(), holda())
    CALL matmult(nu%, nc%, nu%, holda(), gm(), holdb())
    CALL MATADD(nu%, nu%, holdb(), id(), holde())
    CALL matinvs(nu%, holde(), holdc())
    CALL matmult(nu%, nu%, nc%, holde(), holda(), gt())

    FOR i% = 1 TO 10
        FOR j% = 1 TO 10
            IF i% = j% THEN
                id(i%, j%) = 1
            ELSE
                id(i%, j%) = 0
            END IF
        NEXT j%
    NEXT i%

    w(0, 1) = y(0, 1)
    FOR i% = 1 TO nc%

```

```

        w(i%, 1) = alph(1) * w(i% - 1, 1) + (1 - alph(1)) * spcc(1)
    NEXT i%

    FOR i% = 1 TO nc%
        If(i%, 1) = F(0, i%) * yf(0, 1) + F(1, i%) * yf(1, 1) + F(2, i%) * yf(2,
        1) + F(3, i%, 1) * yf(3, 1)
        If(i%, 1) = If(i%, 1) + gamma(0, i%) * (uf(-1, 1) - uf(-2, 1)) + gamma(1, i%) *
        (uf(-2, 1) - uf(-3, 1)) + gamma(2, i%) * (uf(-3, 1) - uf(-4, 1)) + gamma(3, i%) *
        (uf(-4, 1) - uf(-5, 1))
    NEXT i%

    CALL matsub(nc%, 1, w(), If(), holda())
    CALL matmult(nu%, nc%, 1, gt(), holda(), holdb())

    u(0, 1) = u(-1, 1) + holdb(1, 1)
    u(0, 2) = u(-1, 2) + holdb(nu% + 1, 1)

    IF u(0, 1) > 22 THEN
        u(0, 1) = 22
    END IF
    IF u(0, 1) < -18 THEN
        u(0, 1) = -18
    END IF

    TF u(0, 2) > 250 THEN
        u(0, 2) = 250
    END IF
    TF u(0, 2) < 150 THEN
        u(0, 2) = 150
    END IF

    689
    IF warmup% = 0 THEN
        u(0, 1) = ff - 18
        u(0, 2) = tairi
    ELSE
        ff = u(0, 1) + 18
        tairi = u(0, 2)
    END IF

    i% = 1

```

```

FOR j% = -5 TO -1
    y(j%, i%) = y(j% + 1, i%)
    u(j%, i%) = u(j% + 1, i%)
    yf(j%, i%) = yf(j% + 1, i%)
    uf(j%, i%) = uf(j% + 1, i%)
    dyf(j%, i%) = dyf(j% + 1, i%)
    duf(j%, i%) = duf(j% + 1, i%)
NEXT j%

change% = 1

RETURN

8100

' SUBROUTINE TO PERFORM VARIABLE FORGETTING FACTOR RLS

lo% = 1

fi(1) = -dyf(-1, 1)
fi(2) = -dyf(-2, 1)
fi(3) = -dyf(-3, 1)
fi(4) = duf(-1, 1)
fi(5) = duf(-2, 1)
fi(6) = duf(-3, 1)
fi(7) = duf(-4, 1)

theta(1) = a(1, lo%)
theta(2) = a(2, lo%)
theta(3) = a(3, lo%)
theta(4) = b(0, lo%, 1)
theta(5) = b(1, lo%, 1)
theta(6) = b(2, lo%, 1)
theta(7) = b(3, lo%, 1)

error = 0

FOR i% = 1 TO 10
    error = error + fi(i%) * theta(i%)
NEXT i%

error = dyf(0, 1) - error

FOR i% = 1 TO 10
    FOR j% = 1 TO 10
        pt(i%, j%) = p(i%, j%, lo%)
    NEXT j%
NEXT i%

FOR i% = 1 TO 10
    FOR j% = 1 TO 10
        holda(i%, j%) = 0
    NEXT j%
NEXT i%

FOR i% = 1 TO 10
    holda(i%, 1) = fi(i%)
NEXT i%

CALL mattran(10, 10, holda(), holdb())
CALL matmult(10, 10, 10, holdb(), pt(), holdd())

value1 = 0
FOR i% = 1 TO 10
    value1 = value1 + holdd(1, i%) * fi(i%)
NEXT i%

lamrls = 1 - error ^ 2 / (1000 * (1 + value1))

IF lamrls < .9 THEN lamrls = .9
lamrls = .9
CALL matmult(10, 10, 10, pt(), holda(), holddd())

FOR i% = 1 TO 10
    k(i%) = holddd(i%, 1) / (1 + value1)
NEXT i%

```

```

FOR i% = 1 TO 10
    theta(i%) = theta(i%) + k(i%) * error
NEXT i%

FOR i% = 1 TO 10
    FOR j% = 1 TO 10
        holde(i%, j%) = 0
    NEXT j%
NEXT i%

FOR i% = 1 TO 10
    holde(i%, 1) = k(i%)
NEXT i%

CALL matmult(10, 10, 10, holde(), holde(), holdf())
CALL matsub(10, 10, pt(), holdf(), wris())

IF wris(1, 1) / lamrls <= 1 THEN
    FOR i% = 1 TO 10
        FOR j% = 1 TO 10
            pt(i%, j%) = wris(i%, j%) / lamrls
        NEXT j%
    NEXT i%
ELSE
    FOR i% = 1 TO 10
        FOR j% = 1 TO 10
            pt(i%, j%) = wris(i%, j%)
        NEXT j%
    NEXT i%
END IF
trace = 0
FOR i% = 1 TO 10
    FOR j% = 1 TO 10
        p(i%, j%, lo%) = pt(i%, j%)
        IF i% = j% THEN
            trace = trace + p(i%, j%, lo%)
        END IF
    NEXT j%
NEXT i%

a(1, lo%) = theta(1)

```

```

a(2, lo%) = theta(2)
a(3, lo%) = theta(3)
b(0, lo%, 1) = theta(4)
b(1, lo%, 1) = theta(5)
b(2, lo%, 1) = theta(6)
b(3, lo%, 1) = theta(7)

```

RETURN

```

9500
' subroutine for membrane wall heat transfer

waterq = waterfl / 1000
him = 58.9166 * (5 + 84150.251# * waterq ^ .851)
7501
dtl = tg(s%) - twi
dt2 = tg(s%) - two
lmtd = (dt1 - dt2) / LOG(dt1 / dt2)
7502
loadmem = 30
hom = 13 + 2 * loadmem + 4.735E-08 * (tg(s%) ^ 4 - twall ^ 4) / (tg(s%) - twall)
dto = lmtd / (1.5958 * hom / him + .000278225# * hom + 1)
twalln = tg(s%) - dto
IF ABS(twalln - twall) > 1 THEN
    twall = (twall + twalln) / 2
    GOTO 7502
END IF
uom = 1 / (1.5958 / him + .000278225# + 1 / hom)

```

```

heatflow = uom * aom * limtd
twon = heatflow / waterfl / 41.75 + twi
IF twon > 373 THEN twon = 373
IF ABS(twon - two) > .0001 THEN
  two = (twon + two) / 2
  IF two > 373 THEN two = 373
  GOTO 7501
END IF

```

```

RETURN

```

```

9700

```

```

KEY 1, "on/off"
KEY 2, ""
KEY 3, ""
KEY 4, "u-ch"
KEY 5, "l-ch"
KEY 6, ""
KEY 7, "u-sp"
KEY 8, "l-sp"
KEY 9, ""
KEY 10, "main"

```

```

KEY(1) ON
KEY(2) ON
KEY(3) ON
KEY(4) ON
KEY(5) ON
KEY(6) ON
KEY(7) ON
KEY(8) ON
KEY(9) ON
KEY(10) ON

```

```

ON KEY(1) GOSUB 9710
ON KEY(2) GOSUB 9990
ON KEY(3) GOSUB 9990
ON KEY(4) GOSUB 9720

```

```

ON KEY(5) GOSUB 9730
ON KEY(6) GOSUB 9990
ON KEY(7) GOSUB 9740
ON KEY(8) GOSUB 9750
ON KEY(9) GOSUB 9990
ON KEY(10) GOSUB 9950

```

```

KEY ON
RETURN

```

```

9710
CLS
IF limit% = 0 THEN
  limit% = 1
ELSE
  limit% = 0
END IF
RETURN
9720
CLS
INPUT "upper limit of change = ?"; uch
RETURN

```

```

9730
CLS
INPUT "lower limit of change = ?"; lch
RETURN

```

```

9740
CLS
INPUT "upper set-point = ?"; usp
usp = usp + 273
RETURN

```

```

9750
CLS
INPUT "lower set-point = ?"; lsp
lsp = lsp + 273
RETURN

```

```

5300      REM      THIS SUBROUTINE SETS UP TEMPLATE FOR TEMPERATURE
      REM      PROFILE PLOTS
      COLOR 7,0
      LINE (39,4)-(451,4)
      LINE (451,4)-(451,156)
      LINE (451,156)-(39,156)
      LINE (39,156)-(39,4)
      LOCATE 2,58: PRINT "FILE STATUS....."
      LOCATE 6,58: PRINT "ELAPSED TIME....."
      LOCATE 10,58: COLOR 3,0: PRINT "L-3....."
      LOCATE 14,58: COLOR 2,0: PRINT "L-12....."
      LOCATE 18,58: COLOR 4,0: PRINT "R-5....."
      COLOR 14,0
      LOCATE 23,16: PRINT "TRANSIENT TEMPERATURE PROFILES"
      COLOR 7,0
      xc% = 39

      REM*****
      *****

      REM*****
      *****

5400      REM      SUBROUTINE TO PUT NUMBERS ON TEMPERATURE PROFILE
      REM      TEMPLATE
      IF countint% = 0 THEN
        maxt% = 950
        mint% = 800
        xc% = xc% + 1
        IF xc% > 450 THEN
          LINE (xc% - 1,162)-(xc% - 1,157),0
          LINE (xc% - 1,157)-(xc% - 1 - 2,159),0
          LINE (xc% - 1,157)-(xc% - 1 + 2,159),0
          LINE (xc% - 1 - 2,159)-(xc% - 1 + 2,159),0
          LINE (xc% - 162)-(xc%,157)
          LINE (xc%,157)-(xc% - 2,159)
          LINE (xc%,157)-(xc% + 2,159)
          LINE (xc% - 2,159)-(xc% + 2,159)
          xc% = 40
        END IF
        COLOR 8,0
      END IF

      Xxc% = xc% + 1

      LINE (xc%,5)-(xc%,155),0
      IF Xxc% <= 450 THEN
        LINE (Xxc%,5)-(Xxc%,155)
      END IF
      REM      DEFINE POINT FOR L-3
      PT1% = INT((((8 - dmpo2) / 8) * 150) + 5)

      REM      DEFINE POINT FOR L-12
      PT2% = INT((((maxt% - (tg(1) - 273)) / 150) * 150) + 5)

      REM      DEFINE POINT FOR R-5
      PT3% = INT((((maxt% - VALUE(1,14)) / 200) * 150) + 5)

      REM      DEFINE LIMITS OF VALUES FOR PLOT
      IF PT1% < 5 THEN PT1% = 5
      IF PT2% < 5 THEN PT2% = 5
      IF PT3% < 5 THEN PT3% = 5
      IF PT1% > 155 THEN PT1% = 155
      IF PT2% > 155 THEN PT2% = 155
      IF PT3% > 155 THEN PT3% = 155

      REM      PUT ON POINTS
      COLOR 3,0: PSET (xc%,PT1%)
      COLOR 2,0: PSET (xc%,PT2%)
      COLOR 4,0: PSET (xc%,PT3%)

      REM      DRAW CUTE LITTLE ARROW
      COLOR 5,0
      LINE (xc% - 1,162)-(xc% - 1,157),0
      LINE (xc% - 1,157)-(xc% - 1 - 2,159),0
      LINE (xc% - 1,157)-(xc% - 1 + 2,159),0
      LINE (xc% - 1 - 2,159)-(xc% - 1 + 2,159),0
      LINE (xc%,162)-(xc%,157)
      LINE (xc%,157)-(xc% - 2,159)
      LINE (xc%,157)-(xc% + 2,159)
      LINE (xc% - 2,159)-(xc% + 2,159)

      END IF

```

```

REM PUT ON THE NUMBERS
LOCATE 3, 65
IF saver% = 0 THEN
    COLOR 7, 0: PRINT "WAITING"
ELSEIF saver% = 1 THEN
    COLOR 4, 0: PRINT "SAVING "
END IF
COLOR 7, 0
LOCATE 7, 65: PRINT USING "###.##", minit + count% / 60
LOCATE 11, 65: COLOR 3, 0: PRINT USING "###.###"; dmpo2
LOCATE 15, 65: COLOR 2, 0: PRINT USING "###.###"; tg(1) - 273
LOCATE 19, 65: COLOR 4, 0: PRINT USING "###.###"; fr
COLOR 7, 0

' END IF

RETURN
REM*****g*****

REM $STATIC
SUB MATADD (n%, m%, aay(), bbx(), c())

' SUBROUTINE TO ADD TWO MATRICES
' N=# ROWS
' M=# COLUMNS
' A=FIRST MATRIX
' B=SECOND MATRIX
' C=RESULTANT MATRIX

FOR i% = 1 TO n%
    FOR j% = 1 TO m%
        c(i%, j%) = aay(i%, j%) + bbx(i%, j%)
    NEXT j%
NEXT i%

END SUB

SUB matinvs (n%, aay(), bbx())

```

---

```

' SUBROUTINE TO TAKE THE INVERSE OF A MATRIX (MUST BE
SQUARE)
' N=# ROWS = # COLUMNS
' A = INPUT MATRIX
' B=INVERSE OF MATRIX A

DIM id(1 TO 16, 1 TO 16)

' SET UP IDENTITY MATRIX
-FOR i% = 1 TO n%
FOR j% = 1 TO n%
    IF i% = j% THEN
        bbx(i%, j%) = 1
    ELSE
        bbx(i%, j%) = 0
    END IF
NEXT j%
NEXT i%

' PUT A INTO ANOTHER MATRIX TO SAVE A AS IT IS
FOR i% = 1 TO n%
FOR j% = 1 TO n%
    id(i%, j%) = aay(i%, j%)
NEXT j%
NEXT i%

' GAUSS-JORDAN ELIMINATION TO TURN ID INTO IDENTITY MATRIX
' RESULT IN B IS THE INVERSE
FOR i% = 1 TO n%
    IF id(i%, i%) = 0 THEN
        FOR j% = i% + 1 TO n%
            IF id(j%, i%) < 0 THEN
                FOR k% = 1 TO n%
                    HOLD = id(i%, k%)
                    id(i%, k%) = id(j%, k%)
                    id(j%, k%) = HOLD
                END FOR
                bbx(i%, k%) = bbx(j%, k%)
                bbx(j%, k%) = HOLD
            END IF
        NEXT k%
        GOTO 10
    END IF

```

```

END IF
IF j% = n% THEN
  PRINT "CANNOT TAKE INVERSE OF MATRIX"
  STOP
END IF
NEXT j%
END IF

10  HOLD = id(i%, i%)

FOR j% = 1 TO n%
  id(i%, j%) = id(i%, j%) / HOLD
  bbx(i%, j%) = bbx(i%, j%) / HOLD
NEXT j%

FOR j% = 1 TO n%
  IF j% = i% THEN GOTO 66
  HOLD = id(j%, i%) / id(i%, i%)
  FOR k% = 1 TO n%
    id(j%, k%) = id(j%, k%) - HOLD * id(i%, k%)
    bbx(j%, k%) = bbx(j%, k%) - HOLD * bbx(i%, k%)
  NEXT k%
NEXT j%

66  NEXT j%
NEXT i%

END SUB

SUB matmult (n%, m%, o%, aay(), bbx(), c())

' SUBROUTINE TO MULTIPLY TWO MATRICES
' N = # ROWS OF MATRIX A
' M = # COLUMNS OF MATRIX A
' O = # ROWS OF MATRIX B (I THINK ITS COLUMNS)
' A=FIRST MATRIX
' B=SECOND MATRIX
' C=RESULTANT MATRIX

SUB mattran (n%, m%, aay(), bbx())

```

10

66



```

' SUBROUTINE TO TAKE THE TRANSPOSE OF A MATRIX
'
' N=# ROWS OF A
' M=# COLUMNS OF A
' A=INPUT MATRIX
' B=TRANSPOSE OF A
'
  FOR i% = 1 TO n%
    FOR j% = 1 TO m%
      btx(j%, i%) = aay(i%, j%)
    NEXT j%
  NEXT i%

END SUB

```

## **APPENDIX E - PROGRAM LISTING - TIME SERIES ANALYSIS AND TRANSFER**

### **FUNCTION FITTING**

This appendix provides the listing of the QuickBasic program written for analysis of the data collected for process identification. It employs the methods outlined in Section 5.5.2 and includes auto-, cross-, and partial-correlation of the data, graphical output to the screen for model order identification, and parameter fitting of the chosen transfer function.

```

REM $DYNAMIC
DECLARE SUB VARI(N AS INTEGER, I AS INTEGER, DATAA() AS
SINGLE, MEAN AS SINGLE, VAR AS SINGLE)
DECLARE SUB COREL(AUTOA() AS SINGLE, AUTOB() AS SINGLE,
CHOICE AS INTEGER, N AS INTEGER, DATAA() AS SINGLE, DATAB() AS
SINGLE, MEANA AS SINGLE, MEANB AS SINGLE, VARA AS SINGLE,
VARB AS SINGLE, COR() AS SINGLE, ERM() AS SINGLE, ERP() AS _
SINGLE)
DECLARE SUB GAUS(C AS INTEGER, XX() AS SINGLE, AUTOA() AS
SINGLE, CROSST() AS SINGLE, V() AS SINGLE, FLAG AS INTEGER)

REM DIMENSION VARIABLES

DIM FIRST$(1 TO 11)
DIM REF%(1 TO 10)
DIM ORDERN%(1 TO 10)
DIM ORDERD%(1 TO 10)
DIM DEAD%(1 TO 11)
DIM VART(1 TO 11, 0 TO 2)
DIM MEANT(1 TO 11, 0 TO 2)
DIM DAT(-10 TO 1000, 1 TO 11)
DIM HOLDER(1 TO 3, 1 TO 11)
DIM TEMPD(1 TO 1000)
DIM PLOTT(1 TO 1000)
DIM INPCOL%(11)
DIM PLOTM(0 TO 1000)
DIM MEAN(1 TO 11)
DIM VAR(1 TO 11)
DIM COR(-50 TO 50)
DIM SAVE(-7 TO 5)
DIM TAG(-50 TO 50)
DIM ERM(-50 TO 50)
DIM ERP(-50 TO 50)
DIM IMPUL(-7 TO 50, 1 TO 100)
DIM AUTO(-50 TO 50, 1 TO 11)
DIM ERPP(-50 TO 50, 1 TO 11)
DIM ERM(-50 TO 50, 1 TO 11)
DIM CROSS(-50 TO 50, 1 TO 100)
DIM CERP(-50 TO 50, 1 TO 100)
DIM CERM(-50 TO 50, 1 TO 100)

DIM V(1 TO 51)
DIM WATE(0 TO 50)
DIM DATAA(1 TO 1000)
DIM DATAB(1 TO 1000)
DIM AUTOA(-50 TO 50)
DIM AUTOB(-50 TO 50)
DIM CROSST(-50 TO 51)
DIM ERR(1 TO 1000)
DIM STD(1 TO 2)
DIM COEF(1 TO 50)
DIM YY(-50 TO 50)
DIM XX(1 TO 51, 1 TO 51)
DIM ANS(1 TO 50)
DIM COEFO(1 TO 50)
DIM YCALC(-10 TO 1000, 1 TO 5)
DIM DERV(1 TO 1000, 1 TO 11)
DIM AO(-10 TO 1000)
DIM AOT(0 TO 1000)
DIM YC(0 TO 1000)
DIM NO(-10 TO 1000)
DIM PART(1 TO 50)
DIM FI(1 TO 50, 1 TO 50)

REM OPEN INPUT AND OUTPUT FILES

OPEN "C:\OPTO\QB\progs\DATA\COR.OUT" FOR OUTPUT ACCESS
WRITE AS #10
OPEN "C:\OPTO\QB\progs\DATA\COR.IN" FOR INPUT ACCESS READ
AS #11

REM INITIALIZE VARIABLES

FOR I% = -50 TO 50
  FOR J% = 1 TO 11
    AUTO(I%, J%) = 0!
  NEXT J%
NEXT I%

FOR I% = 1 TO 11

```

```

DEAD%(I%) = 0!
FOR J% = -10 TO 0
  DAT(J%, I%) = 0!
  NO(J%) = 0!
  NEXT I%
NEXT J%
DIF% = 0
AOT(0) = 0
YC(0) = 0
FLAG = 0
CHECK = 0
LAM = .001
STD(1) = 10000
FIRST$(11) = "RESIDUALS"

```

REM READ DATA

```

PRINT " "
PRINT " "
PRINT " "
PRINT " "
PRINT " "
INPUT #11, N%
PRINT " "
PRINT " "
INPUT #11, M%
PRINT " "
PRINT " "

```

```

FOR I% = 1 TO M%
  INPUT #11, FIRST$(I%)
NEXT I%

PRINT " "
PRINT " "
PRINT "COLUMN", "NAME"
PRINT "_____"
FOR I% = 1 TO M%
  PRINT I%, FIRST$(I%)
NEXT I%

```

```

FOR I% = 1 TO N%
  FOR J% = 1 TO M%
    INPUT #11, DAT(I%, J%)
    IF J% = 1 THEN PRINT DAT(I%, J%)
    INPUT Z%
    NEXT J%
  NEXT I%

  FOR I% = 1 TO 6
    PRINT #10, " "
  NEXT I%

  PRINT #10, CHR$(34); FIRST$(1); CHR$(34), CHR$(34); FIRST$(2);
  CHR$(34), CHR$(34); FIRST$(3); CHR$(34), CHR$(34); FIRST$(4); CHR$(34),
  CHR$(34); FIRST$(5); CHR$(34), CHR$(34); FIRST$(6); CHR$(34), CHR$(34);
  FIRST$(7); CHR$(34)

  PRINT #10, CHR$(34); "_____" ; CHR$(34)

```

```

FOR I% = 1 TO N%
  PRINT #10, DAT(I%, 1), DAT(I%, 2), DAT(I%, 3), DAT(I%, 4),
  DAT(I%, 5), DAT(I%, 6), DAT(I%, 7), DAT(I%, 8), DAT(I%, 9), DAT(I%, 10)
NEXT I%

```

```

FOR I% = 1 TO M%
  FOR J% = 1 TO N%
    DATAA(J%) = DAT(J%, I%)
  NEXT J%
  VARI N%, I%, DATAA(), MEAN(I%), VAR(I%)
  MEANT(I%, DIF%) = MEAN(I%)
  VART(I%, DIF%) = VAR(I%)
NEXT I%

```

```

PRINT #10, " "
PRINT #10, CHR$(34); "MEAN"; CHR$(34)
PRINT #10, MEAN(1), MEAN(2), MEAN(3), MEAN(4), MEAN(5),
  MEAN(6), MEAN(7), MEAN(8), MEAN(9), MEAN(10)
PRINT #10, " "

```



```

PRINT " "
PRINT "(1) AUTOCORRELATION"
PRINT "(2) CROSSCORRELATION"
PRINT "(3) RESPONSE"
PRINT "(4) FIT MODEL"
PRINT "(5) DIFFERENCE DATA"
PRINT "(6) UNDIFFERENCE DATA"
PRINT "(7) PLOTS"
PRINT "(8) MODEL SUMMARY"
PRINT "(9) QUIT"
INPUT CHOICE%
IF CHOICE% = 0 GOTO 999
IF CHOICE% < 0 OR CHOICE% > 8 GOTO 70

71
IF CHOICE% = 1 THEN
  IF FLAGGER% = 1 THEN
    INPUT "WHICH COLUMN WOULD YOU LIKE TO
    AUTOCORRELATE? ---- ", A%
    B% = A%
    END IF
    GOTO 620
  ELSEIF CHOICE% = 2 THEN
    IF FLAGGER% = 1 THEN
      INPUT "CROSSCORRELATION INPUT? ---- ", A%
      INPUT "CROSSCORRELATION OUTPUT? ---- ", B%
      END IF
      GOTO 620
    ELSEIF CHOICE% = 3 THEN
      IF FLAGGER% = 1 THEN
        INPUT "RESPONSE INPUT? ---- ", A%
        INPUT "RESPONSE OUTPUT? ---- ", B%
        INPUT "NUMBER OF LAGS? ---- ", C%
        C% = 20
        END IF
        GOTO 621
      ELSEIF CHOICE% = 4 THEN
        GOTO 987
      ELSEIF CHOICE% = 5 THEN
        DIF% = DIF% + 1
        FOR J% = 1 TO M%
          FOR I% = DIF% + 1 TO N%
            TEMPD(I%) = DAT(I%, J%) - DAT(I% - 1, J%)
            DATAA(I% - DIF%) = TEMPD(I%)
          NEXT I%
          FOR I% = DIF% + 1 TO N%
            DAT(I%, J%) = TEMPD(I%)
          NEXT I%
          VARI N% - DIF%, J%, DATAA(), MEANT(J%, DIF%),
          VART(J%, DIF%)
          FOR I% = DIF% + 1 TO N%
            DAT(I%, J%) = DAT(I%, J%) - MEANT(J%, DIF%)
            DATAA(I% - DIF%) = DAT(I%, J%)
          NEXT I%
          VARI N% - DIF%, J%, DATAA(), MEAN(J%), VAR(J%)
          NEXT J%
          FLAGGER% = 0
          DOIT% = 1
          CHOICE% = 1
          HALT% = 0
          GOTO 70
        ELSEIF CHOICE% = 6 THEN
          DIF% = DIF% - 1
          FOR J% = 1 TO M%
            FOR I% = DIF% + 2 TO N%
              DAT(I%, J%) = DAT(I%, J%) + MEANT(J%, DIF% + 1) +
              DAT(I% - 1, J%)
            NEXT I%
            DATAA(I% - DIF%) = DAT(I%, J%)
            NEXT J%
            DATAA(1) = DAT(1 + DIF%, J%)
            VARI N% - DIF%, J%, DATAA(), MEAN(J%), VAR(J%)
            MEANT(J%, DIF% + 1) = 0
            NEXT J%
            FLAGGER% = 0
            DOIT% = 1
            CHOICE% = 1
            HALT% = 0
            GOTO 70
          ELSEIF CHOICE% = 7 THEN

```

```

GOTO 7000
ELSEIF CHOICE% = 8 THEN
  PRINT " "
  PRINT " "
  PRINT "MODEL ACCOUNTS FOR"
  REM   PRINT 100 - 100 * STAN / VAR(C%)
  PRINT 100 - 100 * STAN / VART(C%, 0)
  PRINT "PERCENT OF THE OUTPUT VARIANCE"
  PRINT " "
  PRINT "RESIDUAL VARIANCE = ", STAN
  PRINT " "
  FOR I% = 1 TO H%
    PRINT "FOR TRANSFER FUNCTION", FIRST$(OUTCOL%),
    "TO", FIRST$(REF%(I%))
    PRINT "   ORDER OF NUMERATOR = ", ORDERN%(I%)
    PRINT "   ORDER OF DENOMINATOR = ", ORDERD%(I%)
    PRINT " "
  NEXT I%
  PRINT "NOISE MODEL"
  PRINT "   ORDER OF NUMERATOR = ", ONN%
  PRINT "   ORDER OF DENOMINATOR = ", OND%
  PRINT " "
  FOR I% = 1 TO NUMCO%
    PRINT "VARIABLE #", I%, "=", COEFO(I%)
  NEXT I%
  PRINT " "
  INPUT "ANY KEY TO CONTINUE", CRAP
  PRINT #10, " "
END IF

620 FOR I% = -50 TO 50
  AUTOA(I%) = AUTO(I%, A%)
  AUTOB(I%) = AUTO(I%, B%)
NEXT I%

FOR I% = DIF% + 1 TO N%
  DATAA(I% - DIF%) = DAT(I%, A%)
  DATAB(I% - DIF%) = DAT(I%, B%)
NEXT I%

COREL AUTOA(), AUTOB(), CHOICE%, N% - DIF%, DATAA(),
DATAB(), MEAN(A%), MEAN(B%), VAR(A%), VAR(B%), COR(), ERM(),
ERP()

GOTO 670

621 C% = C% + 1
FOR I% = -50 TO 50
  AUTOA(I%) = AUTO(I%, A%)
  CROSST(I%) = CROSS(I%, 10 * A% + B%)
NEXT I%

FLAG% = 1

GAUS C%, XX(), AUTOA(), CROSST(), V(), FLAG%

FLAG% = 0
GOTO 670

670 FOR I% = 1 TO 5
  PRINT #10, " "
NEXT I%

IF CHOICE% = 1 THEN
  FOR I% = -50 TO 50
    AUTO(I%, A%) = COR(I%)
    ERM(I%, A%) = ERM(I%)
    ERP(I%, A%) = ERP(I%)
  NEXT I%
  PRINT #10, CHR$(34); "AUTOCORRELATION OF COLUMN ";
  CHR$(34), A%
  PRINT #10, " "
  PRINT #10, CHR$(34); "K"; CHR$(34), CHR$(34); "CORREL ";
  CHR$(34), CHR$(34); "CONFID-"; CHR$(34), CHR$(34); "CONFID+";
  CHR$(34)
  PRINT #10, CHR$(34); "-----";
  CHR$(34)
  FOR I% = 1 TO 50

```

```

PRINT #10, I%, COR(I%), ERM(I%), ERP(I%)
NEXT I%

ELSEIF CHOICE% = 2 THEN
PRINT #10, CHR$(34); "CROSSCORRELATION OF COLUMN ";
CHR$(34), B%, CHR$(34); " TO COLUMN "; CHR$(34), A%
PRINT #10, " "
PRINT #10, CHR$(34); "K"; CHR$(34), CHR$(34); "CORREL ";
CHR$(34), CHR$(34); "CONFID-"; CHR$(34), CHR$(34); "CONFID+";
CHR$(34)
PRINT #10, CHR$(34); "_____";
CHR$(34)
FOR I% = -50 TO 50
CROSS(I%, 10 * A% + B%) = COR(I%)
CERM(I%, 10 * A% + B%) = ERM(I%)
CERP(I%, 10 * A% + B%) = ERP(I%)
NEXT I%
ERRI(10 * A% + B%) = ERP(0) / 2

FOR I% = -50 TO 50
PRINT #10, I%, COR(I%), ERM(I%), ERP(I%)
NEXT I%

ELSEIF CHOICE% = 3 THEN

FOR I% = 1 TO C%
WATE(I% - 1) = V(I%) * SQR(VAR(B%) / VAR(A%))
NEXT I%
CP% = 10 * A% + B%

FOR I% = 0 TO C% - 1
IMPUL(I%, CP%) = WATE(I%)
NEXT I%

REM*****
REM IMPULSE RESPONSE FOR WHITE NOISE INPUT
FOR I% = 0 TO C%
IMPUL(I%, C%) = CROSS(I%, CP%) * SQR(VAR(B%) /
VAR(A%))
WATE(I%) = IMPUL(I%, C%)
NEXT I%

REM*****

```

987

```

VARI 20, 12, WATE(), MEANI, VARI
ERRI(CP%) = MEANI + 2.1 * (VARI / 20) ^ .5
ERRI(999) = -ERRI(CP%)

PRINT #10, CHR$(34); "IMPULSE RESPONSE OF COLUMN ";
CHR$(34), B%, CHR$(34); " TO COLUMN "; CHR$(34), A%
PRINT #10, " "
PRINT #10, CHR$(34); "K"; CHR$(34), CHR$(34); "RESPONSE";
CHR$(34)
PRINT #10, CHR$(34); "_____";
CHR$(34)
FOR I% = 1 TO 20
PRINT #10, I% - 1, WATE(I% - 1), ERM(I% - 1) * ERM(999)
NEXT I%
FOR I% = 1 TO 5
PRINT #10, " "
NEXT I%
PRINT #10, CHR$(34); "STEP RESPONSE OF COLUMN ";
CHR$(34), B%, CHR$(34); " TO COLUMN "; CHR$(34), A%
PRINT #10, " "
PRINT #10, CHR$(34); "K"; CHR$(34), CHR$(34); "RESPONSE";
CHR$(34)
PRINT #10, CHR$(34); "_____";
CHR$(34)
SUM = 0
SUM1 = 0
FOR I% = 1 TO 20
SUM = SUM + WATE(I% - 1)
PRINT #10, I% - 1, SUM
NEXT I%

END IF

GOTO 70

```



```

PRINT " "
PRINT " "
INPUT "ENTER OUTPUT COLUMN FOR MODEL RESPONSE -----"
" , C%
REM C% = OUTCOL%

FOR I% = 1 TO 10
ORDERN%(I%) = 0
ORDERD%(I%) = 0
NEXT I%

INPUT "HOW MANY INPUTS ARE THERE? -----" , H%

FOR I% = 1 TO H%
PRINT "INPUT COLUMN NUMBER FOR INPUT " , I%
INPUT G%
REF%(I%) = G%
INPUT "ORDER OF NUMERATOR (0-3) -----" , X%
ORDERN%(I%) = X%
INPUT "ORDER OF DENOMINATOR (0-3) -----" , X%
ORDERD%(I%) = X%
INPUT "INPUT DEAD TIME -----" , X%
DEAD%(I%) = X%
NEXT I%

INPUT "ORDER OF NUMERATOR OF NOISE -----" ,
ONN%
INPUT "ORDER OF DENOMINATOR OF NOISE -----" ,
OND%
COUNTS% = 0
DEADM% = 0
FOR I% = 1 TO 10
IF DEAD%(I%) > DEADM% THEN DEADM% = DEAD%(I%)
NEXT I%

DEADM% = DEADM% + 4

GOTO 2007
K% = 1
FOR I% = 1 TO H%
FOR J% = 0 TO ORDERN%(I%)

```

```

PRINT "INPUT INITIAL GUESS FOR COEF #", K%
INPUT COEF(K%)
K% = K% + 1
NEXT J%
FOR J% = 1 TO ORDERD%(I%)
IF ORDERD%(I%) = 0 THEN GOTO 2006
PRINT "INPUT INITIAL GUESS FOR COEF #", K%
INPUT COEF(K%)
K% = K% + 1
NEXT J%
NEXT I%
2006
NEXT I%
NEXT I%
2007
FOR I% = 1 TO H%
IF OND% > 0 THEN GOTO 2111
IF ONN% > 0 THEN GOTO 2111
NEXT I%
IF H% = 0 THEN GOTO 2112
K% = 1
FOR I% = 1 TO H%
S% = ORDERN%(I%)
B% = DEAD%(I%)
CP% = 10 * REF%(I%) + C%
K% = K% + 1 + ORDERN%(I%)
FOR L% = B% - 1 TO -5 STEP -1
SAVE(L%) = IMPUL(L%, CP%)
IMPUL(L%, CP%) = 0
NEXT L%

IF ORDERD%(I%) = 2 THEN
K% = K% + 1
T1 = IMPUL(B% + 2 + S%, CP%) * IMPUL(B% + S%, CP%)
T2 = (IMPUL(B% + 1 + S%, CP%)) ^ 2
T3 = (IMPUL(B% + S%, CP%)) ^ 2
T4 = IMPUL(B% + 1 + S%, CP%) * IMPUL(B% - 1 + S%, CP%)
COEF(K%) = (T1 - T2) / (T3 - T4)
DEL2 = COEF(K%)
K% = K% - 1
T1 = IMPUL(B% + 1 + S%, CP%) / IMPUL(B% + S%, CP%)
T2 = IMPUL(B% - 1 + S%, CP%) / IMPUL(B% + S%, CP%)
COEF(K%) = T1 - T2 * COEF(K% + 1)

```

```

DEL1 = COEF(K%)
K% = K% - 1 - ORDERN%(I%)
FOR F% = 0 TO ORDERN%(I%)
  COEF(K%) = DEL1 * IMPUL(B% - 1 + F%, CP%) + DEL2 *
  IMPUL(B% - 2 + F%, CP%) - IMPUL(B% + F%, CP%)
  COEF(K%) = -1 * COEF(K%)
  K% = K% + 1
NEXT F%
ELSEIF ORDERD%(I%) = 1 THEN
  COEF(K%) = IMPUL(B% + 1 + S%, CP%) / IMPUL(B% + S%,
  CP%)
DEL1 = COEF(K%)
K% = K% - 1 - ORDERN%(I%)
FOR F% = 0 TO ORDERN%(I%)
  COEF(K%) = DEL1 * IMPUL(B% - 1 + F%, CP%) - IMPUL(B%
  + F%, CP%)
  COEF(K%) = -1 * COEF(K%)
  K% = K% + 1
NEXT F%
ELSE
  K% = K% - 1 - ORDERN%(I%)
  FOR F% = 0 TO ORDERN%(I%)
    COEF(K%) = -1 * IMPUL(B% + F%, CP%)
    COEF(K%) = -1 * COEF(K%)
    K% = K% + 1
  NEXT F%
  END IF
  K% = K% + ORDERD%(I%)
  NEXT I%
2111
  K% = 0
  FOR I% = 1 TO H%
    K% = K% + 1 + ORDERN%(I%)
    K% = K% + ORDERD%(I%)
    K% = K% + 1
  NEXT I%
2112
  PRINT "1"
  B% = DEAD%(1)
  FOR L% = B% - 1 TO -5 STEP -1

```

```

  IMPUL(L%, CP%) = SAVE(L%)
  NEXT L%
  PRINT "2"
  IF OND% = 1 AND ONN% = 0 THEN
    IF K% = 0 THEN K% = 1
    COEF(K%) = -TAG(1)
    K% = K% + 1
  ELSEIF OND% = 2 AND ONN% = 0 THEN
    PRINT "3", K%
    COEF(K%) = -TAG(1) * (1 - TAG(2)) / (1 - TAG(2) ^ 2)
    K% = K% + 1
  PRINT "4", K%
    COEF(K%) = -(TAG(2) - TAG(1) ^ 2) / (1 - TAG(1) ^ 2)
    K% = K% + 1
  PRINT "5", K%
  ELSE
    FOR I% = 1 TO ONN%
      IF ONN% = 0 THEN GOTO 2012
      PRINT "INPUT INITIAL GUESS FOR COEF #", K%
      INPUT COEF(K%)
      K% = K% + 1
    NEXT I%
    FOR I% = 1 TO OND%
      IF OND% = 0 THEN GOTO 2016
      PRINT "INPUT INITIAL GUESS FOR COEF #", K%
      INPUT COEF(K%)
      K% = K% + 1
    NEXT I%
    END IF
    LAM = .001
    REM INPUT "LAM?", LAM
    REM INPUT "STD(1)?", STD(1)
    STD(1) = 1E+32
    NUMCO% = K% - 1
9000

```

```

REM      PRINT NUMCO%
REM      FOR I% = 1 TO NUMCO%
REM      PRINT COEF(I%)
REM      NEXT I%

IF H% = 0 THEN
  FOR I% = 1 TO N%
    YC(I%) = 0
  NEXT I%
  Y% = 1
  GOTO 9005
END IF

FOR I% = -10 TO 0
  FOR J% = 1 TO H%
    YCALC(I%, J%) = 0
    YC(I + 10) = 0
    AO(I + 10) = 0
  NEXT J%
NEXT I%

IF DIF% > 0 THEN
  FOR I% = 1 TO DIF%
    FOR J% = 1 TO 11
      HOLDER(I%, J%) = DAT(I%, J%)
      DAT(I%, J%) = 0
    NEXT J%
  NEXT I%
END IF

5000
FOR I% = 1 + DIF% TO N%
  Y% = 1
  FOR K% = 1 TO H%
    YCALC(I%, K%) = 0
    FOR J% = 0 TO ORDERN%(K%)
      YCALC(I%, K%) = YCALC(I%, K%) + COEF(Y%) *
      DAT(I% - DEAD%(K%) - J%, REF%(K%))
    NEXT J%
  NEXT K%
NEXT I%

Y% = Y% + 1
NEXT J%

FOR J% = 1 TO ORDERD%(K%)

  IF ORDERD%(K%) = 0 THEN GOTO 593
  IF ABS(YCALC(I% - J%, K%)) > 1E+25 THEN
    PRINT "YCALC = ", YCALC(I% - J%, K%)
    GOTO 74
  END IF

  YCALC(I%, K%) = YCALC(I%, K%) + COEF(Y%) *
  YCALC(I% - J%, K%)
  Y% = Y% + 1
593
  NEXT J%
NEXT K%

YC(I%) = 0
FOR U% = 1 TO H%
  YC(I%) = YC(I%) + YCALC(I%, U%)
NEXT U%
NEXT I%

NOST% = Y%

9005
SUM = 0
FOR I% = 1 + DIF% TO N%
  SUM = SUM + YC(I%)
NEXT I%
REM   FOR I% = 1 + DIF% TO N%
REM   YC(I%) = YC(I%) - SUM / (N% - DIF%)
REM   NEXT I%
FOR I% = 1 TO N%
  NO(I%) = DAT(I%, C%) - YC(I%)
NEXT I%
FOR I% = -10 TO 0
  AO(I%) = 0
  NO(I%) = 0
NEXT I%

```

```

COEF(Z%) = COEF(Z%) + COEF(Z%) * .001
IF H% = 0 THEN
  FOR I% = 1 TO N%
    YC(I%) = 0
  NEXT I%
  Y% = 1
  GOTO 9006
END IF
FOR I% = 1 + DIF% TO N%
  Y% = 1
  FOR K% = 1 TO H%
    YCALC(I%, K%) = 0
    FOR J% = 0 TO ORDERN%(K%)
      YCALC(I%, K%) = YCALC(I%, K%) +
        COEF(Y%) * DAT(I% - DEAD%(K%) - J%, REF%(K%))
      Y% = Y% + 1
    NEXT J%
    FOR J% = 1 TO ORDERD%(K%)
      IF ORDERD%(K%) = 0 THEN GOTO 1006
      YCALC(I%, K%) = YCALC(I%, K%) +
        COEF(Y%) * YCALC(I% - J%, K%)
      Y% = Y% + 1
    NEXT J%
  NEXT K%
  YC(I%) = 0
  FOR U% = 1 TO H%
    YC(I%) = YC(I%) + YCALC(I%, U%)
  NEXT U%
NEXT I%

SUM = 0
FOR I% = 1 + DIF% TO N%
  SUM = SUM + YC(I%)
NEXT I%
FOR I% = 1 + DIF% TO N%
  YC(I%) = YC(I%) - SUM / (N% - DIF%)
NEXT I%
FOR I% = 1 + DIF% TO N%
  NO(I%) = DAT(I%, C%) - YC(I%)
NEXT I%

```

```

FOR I% = 1 + DIF% TO N%
  Y% = NOST%
  AO(I%) = 0
  FOR J% = 0 TO ONN%
    IF J% = 0 THEN
      AO(I%) = AO(I%) + NO(I%)
    ELSE
      AO(I%) = AO(I%) + COEF(Y%) * AO(I% - J%)
      Y% = Y% + 1
    END IF
  NEXT J%
  FOR J% = 1 TO OND%
    IF OND% = 0 THEN GOTO 835
    AO(I%) = AO(I%) + COEF(Y%) * NO(I% - J%)
    Y% = Y% + 1
  NEXT J%
NEXT I%

FOR I% = 1 + DIF% TO N%
  AOT(I%) = AO(I%)
NEXT I%

FOR J% = 1 + DIF% TO N%
  DATAA(J% - DIF%) = AOT(J%)
NEXT J%
I% = 11
VARI N% - DIF%, I%, DATAA(), MEAN(I%), VAR(I%)

STAN = VAR(11)
IF STAN >= STD(1) THEN
  LAM = LAM / 1.5
  CHECK% = 1
  GOTO 1111
ELSE
  LAM = LAM * 1.5
  STD(1) = STAN
  FOR Z% = 1 TO NUMCO%
    COEFOL = COEF(Z%)

```

```

FOR I%= -10 TO 0 + DIF%
  AO(I%) = 0
  NO(I%) = 0
NEXT I%

FOR I% = 1 + DIF% TO N%
  Y% = NOST%
  AO(I%) = 0
  FOR J% = 0 TO ONN%
    IF J% = 0 THEN
      AO(I%) = AO(I%) + NO(I%)
    ELSE
      AO(I%) = AO(I%) + COEF(Y%) * AO(I% - J%)
      Y% = Y% + 1
    END IF
  NEXT J%
  FOR J% = 1 TO OND%
    IF OND% = 0 THEN GOTO 1011
    AO(I%) = AO(I%) + COEF(Y%) * NO(I% - J%)
    Y% = Y% + 1
  NEXT J%
NEXT I%

FOR I% = 1 + DIF% TO N%
  DERV(I%, z%) = 1000 * (AO(I%) - AOT(I%))
NEXT I%
COEF(z%) = COEFOL
NEXT z%

FOR I% = 1 TO NUMCO%
  YY(I%) = 0
  FOR J% = 20 TO N%
    YY(I%) = YY(I%) + AOT(J%) * DERV(J%, I%)
  NEXT J%
  FOR K% = 1 TO NUMCO%
    FOR J% = 20 TO N%
      YY(I%) = YY(I%) - COEF(K%) * DERV(J%, K%) *
DERV(J%, I%)
    NEXT J%
    NEXT K%

```

386

1011

```

NEXT I%

FOR I% = 1 TO NUMCO%
  YY(I%) = -1 * YY(I%)
NEXT I%

FOR I% = 1 TO NUMCO%
  FOR J% = 1 TO NUMCO%
    XX(I%, J%) = 0
    FOR K% = 20 TO N%
      XX(I%, J%) = XX(I%, J%) + DERV(K%, I%) *
DERV(K%, J%)
    NEXT K%
    NEXT J%
    NEXT I%

GAUS NUMCO%, XX(), AUTOA(), YY(), ANSO, FLAG%

REM      PRINT " "
FOR I% = 1 TO NUMCO%
  COEFO(I%) = COEF(I%)
NEXT I%

END IF

1111 SUMCOF = 0
FOR I% = 1 TO NUMCO%
  COEF(I%) = COEFO(I%) + (ANS(I%) - COEFO(I%)) * LAM
  SUMCOF = SUMCOF + ABS(COEF(I%) - COEFO(I%))
NEXT I%

PRINT " "
PRINT " "
PRINT "LAM = ", LAM
PRINT "RESIDUAL VARIANCE = ", STAN
FOR I% = 1 TO NUMCO%
  PRINT "VARIABLE #", I%, "=", COEFO(I%)
NEXT I%

REM      PRINT LAM, STAN, COEF(1), COEF(2)

```

```

REM      IF LAM < 1E-10 THEN GOTO 9994
      IF LAM < .0000001# THEN GOTO 9994
      IF CHECK% = 1 THEN GOTO 5713
REM      IF SUMCOF < 1E-10 THEN GOTO 9994
REM      IF SUMCOF < .001 THEN GOTO 9994
REM      PRINT "*****", SUMCOF
5713
      COUNTS% = COUNTS% + 1
      CHECK% = 0
      GOTO 5000

9994      IF COUNTS% < 3 THEN GOTO 5713

      FOR I% = 1 TO NUMCO%
        PRINT COEF(I%)
      NEXT I%

      IF DIF% > 0 THEN
        FOR I% = 1 TO DIF%
          FOR J% = 1 TO 11
            DAT(I%, J%) = HOLDER(I%, J%)
          NEXT J%
        NEXT I%
      END IF

      PRINT " "
      PRINT " "
      PRINT " "
      PRINT "MODEL ACCOUNTS FOR"
      REM      PRINT 100 - 100 * STAN / VAR(C%)
      PRINT 100 - 100 * STAN / VART(C%, 0)
      PRINT "PERCENT OF THE OUTPUT VARIANCE"
      PRINT " "
      PRINT " "
      INPUT "ANY KEY TO CONTINUE", CRAP
      PRINT #10, " "
      PRINT #10, " "
      PRINT #10, CHR$(34); "MODEL ACCOUNTS FOR"; CHR$(34)

      PRINT #10, 100 - 100 * STAN / VAR(C%)
      PRINT #10, CHR$(34); "PERCENT OF THE OUTPUT VARIANCE";
      CHR$(34)
      PRINT #10, " "
      FOR I% = 1 TO NUMCO%
        PRINT #10, CHR$(34); "COEFFICIENT #"; CHR$(34), I%, CHR$(34);
        " = ", CHR$(34), COEF(I%)
      NEXT I%
      PRINT #10, " "
      PRINT #10, " "
      PRINT #10, " "
      FOR I% = 0 TO N%
        DAT(I%, 11) = AOT(I%)
      NEXT I%

      FOR J% = 1 + DIF% TO N%
        DATAA(J% - DIF%) = DAT(J%, 11)
      NEXT J%
      I% = 11
      VARI N% - DIF%, I%, DATAA(), MEAN(I%), VAR(I%)

      PRINT #10, CHR$(34); "MEAN OF RESIDUALS = ", CHR$(34),
      MEAN(I%)
      PRINT #10, CHR$(34); "VARIANCE OF RESIDUALS = ", CHR$(34),
      VAR(I%)
      PRINT #10, " "
      PRINT #10, " "

      B% = 11
      FOR J% = 0 TO H%
        IF J% = 0 THEN
          A% = B%
          CHOICE% = 1
          PRINT "AUTOCORRELATING RESIDUALS"
          PRINT " "
          PRINT " "
        ELSE
          A% = REF%(J%)
          CHOICE% = 2
          PRINT "CROSSCORRELATING INPUTS WITH RESIDUALS"

```

```

PRINT #10, "
PRINT #10, CHR$(34); "K"; CHR$(34); CHR$(34); CHR$(34); "CORREL";
CHR$(34); CHR$(34); "CONFID-"; CHR$(34); CHR$(34); CHR$(34); "CONFID+";
CHR$(34)
PRINT #10, CHR$(34); "_____";
CHR$(34)
FOR I% = -50 TO 50
  CROSS(I%, 10 * A% + B%) = COR(I%)
  CERP(I%, 10 * A% + B%) = ERP(I%)
  CERM(I%, 10 * A% + B%) = ERM(I%)
NEXT I%
ERRI(10 * A% + B%) = ERP(0) / 2
FOR I% = -50 TO 50
  PRINT #10, I%, COR(I%), ERM(I%), ERP(I%)
NEXT I%
PRINT #10, "
PRINT #10, "
END IF
NEXT J%

PRINT "PARTIAL CORRELATING RESIDUALS"
PRINT "
PRINT "
FI(1, 1) = AUTO(1, 1)
PART(1) = FI(1, 1)
FOR P% = 1 TO 19
  TOPSUM = 0
  BOTSUM = 0
  FOR J% = 1 TO P%
    TOPSUM = TOPSUM + FI(P%, J%) * AUTO(P% + 1 - J%, 1)
    BOTSUM = BOTSUM + FI(P%, J%) * AUTO(J%, 1)
  NEXT J%
  FI(P% + 1, P% + 1) = (AUTO(P% + 1, 1) - TOPSUM) / (1 -
BOTSUM)
  PART(P% + 1) = FI(P% + 1, P% + 1)
  FOR J% = 1 TO P%
    FI(P% + 1, J%) = FI(P%, J%) - FI(P% + 1, P% + 1) * FI(P%, P% -
J% + 1)
  NEXT J%
NEXT P%

PRINT #10, CHR$(34); "K"; CHR$(34); CHR$(34); "CORREL";
CHR$(34); CHR$(34); "CONFID-"; CHR$(34); CHR$(34); "CONFID+";
CHR$(34)
PRINT #10, CHR$(34); "_____";
CHR$(34)
FOR I% = -50 TO 50
  AUTO(I%, A%) = COR(I%)
  IF ONN% = 0 AND OND% = 0 THEN TAG(I%) = COR(I%)
  ERM(I%, A%) = ERM(I%)
  ERPP(I%, A%) = ERP(I%)
NEXT I%
PRINT #10, CHR$(34); "AUTOCORRELATION OF THE
RESIDUALS"; CHR$(34)
PRINT #10, "
PRINT #10, CHR$(34); "K"; CHR$(34); CHR$(34); "CORREL";
CHR$(34); CHR$(34); "CONFID-"; CHR$(34); CHR$(34); "CONFID+";
CHR$(34)
PRINT #10, CHR$(34); "_____";
CHR$(34)
FOR I% = 1 TO 50
  PRINT #10, I%, COR(I%), ERM(I%), ERP(I%)
NEXT I%
PRINT #10, "
PRINT #10, "
ELSE
PRINT #10, CHR$(34); "CROSSCORRELATION OF COLUMN";
CHR$(34); A%, CHR$(34); " WITH THE RESIDUALS"; CHR$(34)

```

```

PRINT #10, CHR$(34); "PARTIAL CORRELATION OF RESIDUALS";
CHR$(34)
PRINT #10, " "
PRINT #10, CHR$(34); "K"; CHR$(34), CHR$(34); "CORREL"; CHR$(34)
FOR I% = 1 TO 20
  PRINT #10, I%, PART(I%)
NEXT I%

PRINT #10, " "
PRINT #10, " "
PRINT #10, CHR$(34); "RESIDUALS"; CHR$(34)
PRINT #10, " "
FOR I% = 1 TO N%
  PRINT #10, DAT(I%, 11)
NEXT I%
PRINT #10, " "
PRINT #10, " "

GOTO 70
7000

CLS
SCREEN 8
PRINT "PLOT OF:"
PRINT " (1) RAW DATA"
PRINT " (2) AUTOCORRELATION"
PRINT " (3) CROSSCORRELATION"
PRINT " (4) IMPULSE RESPONSE"
PRINT " (5) CROSSCORRELATION OF INPUT WITH RESIDUALS"
PRINT " (6) AUTOCORRELATION OF RESIDUALS"
PRINT " (7) PARTIAL CORRELATION OF RESIDUALS"
PRINT " (8) FULL MODEL RESPONSE"
PRINT " (9) RETURN TO MAIN MENU"
INPUT P%

IF P% = 1 THEN
  INPUT "WHICH COLUMN ?", A%
  CLS
  COLOR 7, 0
  LINE (39, 4)-(451, 4)
  LINE (451, 4)-(451, 156)
  LINE (451, 156)-(39, 156)

```

```

LINE (39, 156)-(39, 4)
MINX = 1 + DIF%
MAXX = N%
MINY = 0
MAXY = 0

FOR I% = 1 + DIF% TO N%
  IF DAT(I%, A%) < MINY THEN MINY = DAT(I%, A%)
  IF DAT(I%, A%) > MAXY THEN MAXY = DAT(I%, A%)
NEXT I%
Y% = INT((MAXY - 0) / (MAXY - MINY) * 150 + 5)
LINE (39, Y%)-(451, Y%)
COLOR 4, 0
I% = 1 + DIF%
X% = INT((I% - MINX) / (MAXX - MINX) * 400 + 40)
Y% = INT((MAXY - DAT(I%, A%)) / (MAXY - MINY) * 150 + 5)
PSET (X%, Y%)
XO% = X%
YO% = Y%
FOR I% = 2 + DIF% TO N%
  X% = INT((I% - MINX) / (MAXX - MINX) * 400 + 40)
  Y% = INT((MAXY - DAT(I%, A%)) / (MAXY - MINY) * 150 + 5)
  LINE (XO%, YO%)-(X%, Y%)
  XO% = X%
  YO% = Y%
NEXT I%
COLOR 2, 0
LOCATE 2, 58: PRINT "MEAN"

LOCATE 3, 58
PRINT MEANT(A%, DIF%)
LOCATE 6, 58: PRINT "VARI"
LOCATE 7, 58
PRINT VART(A%, DIF%)
LOCATE 10, 58: PRINT "DOFD"
LOCATE 11, 58: PRINT DIF%
COLOR 9, 0
LOCATE 23, 16: PRINT "RAW DATA PLOT OF", FIRST$(A%)
COLOR 9, 0
LOCATE 1, 1: PRINT MAXY

```



```

LOCATE 21, 1: PRINT MINY
LOCATE 21, 40: PRINT "NUMBER OF POINTS"
LOCATE 21, 57: PRINT MAXX
COLOR 7, 0
LOCATE 24, 16: INPUT "ANY KEY TO CONTINUE", Z%
GOTO 7000

ELSEIF P% = 2 THEN
  INPUT "WHICH COLUMN ?", A%
  CLS
  COLOR 7, 0
  LINE (39, 4)-(451, 4)
  LINE (451, 4)-(451, 156)
  LINE (451, 156)-(39, 156)
  LINE (39, 156)-(39, 4)
  MINX = 0
  MAXX = 20
  MINY = 0
  MAXY = 0
  FOR I% = 1 TO 20
    IF AUTO(I%, A%) < MINY THEN MINY = AUTO(I%, A%)
    IF AUTO(I%, A%) > MAXY THEN MAXY = AUTO(I%, A%)
  NEXT I%
  YC% = INT((MAXY - 0) / (MAXY - MINY) * 150 + 5)
  LINE (39, YC%)-(451, YC%)
  FOR I% = 1 TO 20
    X% = INT((I% - MINX) / (MAXX - MINX) * 400 + 40)
    Y% = INT((MAXY - AUTO(I%, A%)) / (MAXY - MINY) * 150 + 5)
  NEXT I%
  IF (AUTO(I%, A%) > ERPP(I%, A%)) OR (AUTO(I%, A%) < ERM(M(I%, A%))) THEN
    COLOR 4, 0
  ELSE
    COLOR 9, 0
  END IF
  LINE (X%, Y%)-(X%, YC%)
NEXT I%
COLOR 2, 0
LOCATE 10, 58: PRINT "DOFD"
LOCATE 11, 58: PRINT DIF%
COLOR 9, 0

```

```

LOCATE 23, 16: PRINT "AUTOCORRELATION OF", FIRST$(A%)
COLOR 7, 0
LOCATE 24, 16: INPUT "ANY KEY TO CONTINUE", Z%
GOTO 7000

ELSEIF P% = 3 THEN
  INPUT "INPUT COLUMN ?", A%
  INPUT "OUTPUT COLUMN ?", B%
  CLS
  COLOR 7, 0
  LINE (39, 4)-(451, 4)
  LINE (451, 4)-(451, 156)
  LINE (451, 156)-(39, 156)
  LINE (39, 156)-(39, 4)
  MINX = -20
  MAXX = 20
  MINY = 0
  MAXY = 0
  FOR I% = -20 TO 20
    IF CROSS(I%, 10 * A% + B%) < MINY THEN MINY = CROSS(I%, 10 * A% + B%)
    IF CROSS(I%, 10 * A% + B%) > MAXY THEN MAXY = CROSS(I%, 10 * A% + B%)
  NEXT I%
  YC% = INT((MAXY - 0) / (MAXY - MINY) * 150 + 5)
  LINE (39, YC%)-(451, YC%)
  X% = INT((0 - MINX) / (MAXX - MINX) * 400 + 40)
  LINE (X%, 4)-(X%, 156)
  FOR I% = -20 TO 20
    X% = INT((I% - MINX) / (MAXX - MINX) * 400 + 40)
    Y% = INT((MAXY - CROSS(I%, 10 * A% + B%)) / (MAXY - MINY) * 150 + 5)
    IF (CROSS(I%, 10 * A% + B%) > CERP(I%, 10 * A% + B%)) OR (CROSS(I%, 10 * A% + B%) < CERM(I%, 10 * A% + B%)) THEN
      COLOR 4, 0
    ELSE
      COLOR 9, 0
    END IF
    LINE (X%, Y%)-(X%, YC%)
  NEXT I%

```

```

COLOR 2, 0
LOCATE 10, 58: PRINT "DOFD"
LOCATE 11, 58: PRINT DIF%
COLOR 9, 0
LOCATE 23, 5: PRINT "CROSSCORRELATION OF", FIRST$(B%),
"TO", FIRST$(A%)
COLOR 7, 0
LOCATE 24, 16: INPUT "ANY KEY TO CONTINUE", z%
GOTO 7000

ELSEIF P% = 4 THEN
INPUT "INPUT COLUMN ?", A%
INPUT "OUTPUT COLUMN ?", B%
CLS
COLOR 7, 0
LINE (39, 4)-(451, 4)
LINE (451, 4)-(451, 156)
LINE (451, 156)-(39, 156)
LINE (39, 156)-(39, 4)
MINX = 0
MAXX = 20
MINY = 0
MAXY = 0

FOR I% = 0 TO 20
IF IMPUL(I%, 10 * A% + B%) < MINY THEN MINY =
IMPUL(I%, 10 * A% + B%)
IF IMPUL(I%, 10 * A% + B%) > MAXY THEN MAXY =
IMPUL(I%, 10 * A% + B%)
NEXT I%
YC% = INT((MAXY - 0) / (MAXY - MINY) * 150 + 5)
LINE (39, YC%)-(451, YC%)
FOR I% = 0 TO 20
X% = INT((I% - MINX) / (MAXX - MINX) * 400 + 40)
Y% = INT((MAXY - IMPUL(I%, 10 * A% + B%)) / (MAXY -
MINY) * 150 + 5)
IF (IMPUL(I%, 10 * A% + B%) > ERRJ(10 * A% + B%)) OR
(IMPUL(I%, 10 * A% + B%) < ERRJ(999)) THEN
COLOR 4, 0
ELSE

```

```

COLOR 9, 0
END IF
LINE (X%, Y%)-(X%, YC%)
NEXT I%

COLOR 2, 0
LOCATE 10, 58: PRINT "DOFD"
LOCATE 11, 58: PRINT DIF%
COLOR 9, 0
LOCATE 23, 5: PRINT "IMPULSE RESPONSE OF", FIRST$(B%),
"TO", FIRST$(A%)
COLOR 7, 0
LOCATE 24, 16: INPUT "ANY KEY TO CONTINUE", z%
GOTO 7000

ELSEIF P% = 5 THEN
INPUT "INPUT COLUMN ?", A%
B% = 11
CLS
COLOR 7, 0
LINE (39, 4)-(451, 4)
LINE (451, 4)-(451, 156)
LINE (451, 156)-(39, 156)
LINE (39, 156)-(39, 4)
MINX = -20
MAXX = 20
MINY = 0
MAXY = 0

FOR I% = -20 TO 20
IF CROSS(I%, 10 * A% + B%) < MINY THEN MINY =
CROSS(I%, 10 * A% + B%)
IF CROSS(I%, 10 * A% + B%) > MAXY THEN MAXY =
CROSS(I%, 10 * A% + B%)
NEXT I%
YC% = INT((MAXY - 0) / (MAXY - MINY) * 150 + 5)
LINE (39, YC%)-(451, YC%)
X% = INT((0 - MINX) / (MAXX - MINX) * 400 + 40)
LINE (X%, 4)-(X%, 156)
FOR I% = -20 TO 20
X% = INT((I% - MINX) / (MAXX - MINX) * 400 + 40)

```

```

Y% = INT((MAXY - CROSS(I%, 10 * A% + B%)) / (MAXY -
MINY) * 150 + 5)
IF (CROSS(I%, 10 * A% + B%) > CERP(I%, 10 * A% + B%)) OR
(CROSS(I%, 10 * A% + B%) < CERM(I%, 10 * A% + B%)) THEN
    COLOR 4, 0
ELSE
    COLOR 9, 0
END IF
LINE (X%, Y%)-(X%, YC%)
NEXT I%
COLOR 2, 0
LOCATE 10, 58: PRINT "DOFD"
LOCATE 11, 58: PRINT DIF%
COLOR 9, 0
LOCATE 23, 5: PRINT "CROSSCORRELATION OF RESIDUALS
TO", FIRST$(A%)
COLOR 7, 0
LOCATE 24, 16: INPUT "ANY KEY TO CONTINUE", Z%
GOTO 7000

```

```

ELSEIF P% = 6 THEN
    A% = 11
    CLS
    COLOR 7, 0
    LINE (39, 4)-(451, 4)
    LINE (451, 4)-(451, 156)
    LINE (451, 156)-(39, 156)
    LINE (39, 156)-(39, 4)
    MINX = 0
    MAXX = 20
    MINY = 0
    MAXY = 0

```

```

FOR I% = 1 TO 20
    IF AUTO(I%, A%) < MINY THEN MINY = AUTO(I%, A%)
    IF AUTO(I%, A%) > MAXY THEN MAXY = AUTO(I%, A%)
NEXT I%
YC% = INT((MAXY - 0) / (MAXY - MINY) * 150 + 5)
LINE (39, YC%)-(451, YC%)
FOR I% = 1 TO 20

```

```

X% = INT((I% - MINX) / (MAXX - MINX) * 400 + 40)
Y% = INT((MAXY - AUTO(I%, A%)) / (MAXY - MINY) * 150 +
5)
IF (AUTO(I%, A%) > ERPP(I%, A%)) OR (AUTO(I%, A%) <
ERMM(I%, A%)) THEN
    COLOR 4, 0
ELSE
    COLOR 9, 0
END IF
LINE (X%, Y%)-(X%, YC%)
NEXT I%
COLOR 2, 0
LOCATE 10, 58: PRINT "DOFD"
LOCATE 11, 58: PRINT DIF%
COLOR 9, 0
LOCATE 23, 16: PRINT "AUTOCORRELATION OF RESIDUALS"
COLOR 7, 0
LOCATE 24, 16: INPUT "ANY KEY TO CONTINUE", Z%
GOTO 7000

```

```

ELSEIF P% = 7 THEN
    CLS
    COLOR 7, 0
    LINE (39, 4)-(451, 4)
    LINE (451, 4)-(451, 156)
    LINE (451, 156)-(39, 156)
    LINE (39, 156)-(39, 4)
    MINX = 0
    MAXX = 20
    MINY = 0
    MAXY = 0

```

```

FOR I% = 1 TO 20
    IF PART(I%) < MINY THEN MINY = PART(I%)
    IF PART(I%) > MAXY THEN MAXY = PART(I%)
NEXT I%
YC% = INT((MAXY - 0) / (MAXY - MINY) * 150 + 5)
LINE (39, YC%)-(451, YC%)
COLOR 4, 0
FOR I% = 1 TO 20

```

```

X% = INT((I% - MINX) / (MAXX - MINX) * 400 + 40)
Y% = INT((MAXY - PART(I%)) / (MAXY - MINY) * 150 + 5)
LINE (X%, Y%)-(X%, YC%)
NEXT I%
COLOR 2, 0
LOCATE 10, 58: PRINT "DOFD"
LOCATE 11, 58: PRINT DIF%
COLOR 9, 0
LOCATE 23, 16: PRINT "PARTIAL CORRELATION OF RESIDUALS"
COLOR 7, 0
LOCATE 24, 16: INPUT "ANY KEY TO CONTINUE", Z%
GOTO 7000

ELSEIF P% = 8 THEN
CLS
  INPUT "OUTPUT COLUMN?", A%
  A% = OUTCOL%
CLS
COLOR 7, 0
LINE (39, 4)-(451, 4)
LINE (451, 4)-(451, 156)
LINE (451, 156)-(39, 156)
LINE (39, 156)-(39, 4)
MINX = 1
MAXX = N%
MINY = 0
MAXY = 0

FOR I% = 1 TO N%
  PLOTT(I%) = DAT(I%, A%)
NEXT I%
DIF% = DIF%

997 IF DIF% = 0 THEN GOTO 996
DIF% = DIF% - 1
FOR I% = DIF% + 2 TO N%
  PLOTT(I%) = PLOTT(I%) + MEANT(A%, DIF% + 1) +
PLOTT(I% - 1)
NEXT I%

```

```

996
GOTO 997

FOR I% = 1 TO N%
  PLOTT(I%) = PLOTT(I%) + MEANT(A%, DIF%)
  PLOTM(I%) = PLOTT(I%) - DAT(I%, 11)
NEXT I%

VARIN%, I%, PLOTT(), DMEAN, DVAR
VARIN%, I%, PLOTM(), MMEAN, MVAR

MINY = PLOTT(1)
MAXY = PLOTT(1)
FOR I% = 1 TO N%
  IF PLOTT(I%) < MINY THEN MINY = PLOTT(I%)
  IF PLOTT(I%) > MAXY THEN MAXY = PLOTT(I%)
  IF PLOTM(I%) < MINY THEN MINY = PLOTM(I%)
  IF PLOTM(I%) > MAXY THEN MAXY = PLOTM(I%)
NEXT I%

COLOR 9, 0
I% = 1 + DIF%
X% = INT((I% - MINX) / (MAXX - MINX) * 400 + 40)
Y% = INT((MAXY - PLOTT(I%)) / (MAXY - MINY) * 150 + 5)
PSET (X%, Y%)
XO% = X%
YO% = Y%
FOR I% = 2 + DIF% TO N%
  X% = INT((I% - MINX) / (MAXX - MINX) * 400 + 40)
  Y% = INT((MAXY - PLOTT(I%)) / (MAXY - MINY) * 150 + 5)
  LINE (XO%, YO%)-(X%, Y%)
  XO% = X%
  YO% = Y%
NEXT I%
COLOR 4, 0
I% = 1 + DIF%
X% = INT((I% - MINX) / (MAXX - MINX) * 400 + 40)
Y% = INT((MAXY - PLOTM(I%)) / (MAXY - MINY) * 150 + 5)
PSET (X%, Y%)
XO% = X%
YO% = Y%
FOR I% = 2 + DIF% TO N%
  X% = INT((I% - MINX) / (MAXX - MINX) * 400 + 40)
  Y% = INT((MAXY - PLOTM(I%)) / (MAXY - MINY) * 150 + 5)
  LINE (XO%, YO%)-(X%, Y%)
  XO% = X%
  YO% = Y%
NEXT I%
COLOR 4, 0
I% = 1 + DIF%
X% = INT((I% - MINX) / (MAXX - MINX) * 400 + 40)
Y% = INT((MAXY - PLOTM(I%)) / (MAXY - MINY) * 150 + 5)
PSET (X%, Y%)
XO% = X%
YO% = Y%
FOR I% = 2 + DIF% TO N%
  X% = INT((I% - MINX) / (MAXX - MINX) * 400 + 40)

```

```

Y% = INT((MAXY - PLOTM(I%)) / (MAXY - MINY) * 150 + 5)
LINE (XO%, YO%)-CX%, Y%)
XO% = X%
YO% = Y%
NEXT I%

COLOR 4, 0
LOCATE 2, 58: PRINT "MODEL"
LOCATE 3, 58: PRINT "MEAN"
LOCATE 3, 64: PRINT "MMEAN"
LOCATE 4, 58: PRINT "VAR"
LOCATE 4, 64: PRINT "MVAR"

COLOR 9, 0
LOCATE 6, 58: PRINT "DATA"
LOCATE 7, 58: PRINT "MEAN"
LOCATE 7, 64: PRINT "DMEAN"
LOCATE 8, 58: PRINT "VAR"
LOCATE 8, 64: PRINT "DVAR"

COLOR 2, 0
LOCATE 10, 58: PRINT "DOFD"
LOCATE 11, 58: PRINT "DIF%"

COLOR 9, 0
LOCATE 1, 1: PRINT MAXY
LOCATE 21, 1: PRINT MINY
LOCATE 21, 40: PRINT "NUMBER OF POINTS"
LOCATE 21, 57: PRINT MAXX

COLOR 9, 0
LOCATE 23, 16: PRINT "FULL MODEL PREDICTION (MODEL IN
RED)"

COLOR 7, 0
LOCATE 24, 16: INPUT "ANY KEY TO CONTINUE", Z%
GOTO 7000

ELSEIF P% = 0 THEN
  SCREEN 0

```

```

GOTO 70
END IF

999 STOP
END

REM $STATIC
SUB COREL (AUTOA(), AUTOB(), CHOICE%, N%, DATAA(), DATAO,
MEANA, MEANB, VARA, VARB, COR(), ERM(), ERP())
DIM TEMP(-50 TO 50)

FOR K% = 1 TO 50
  SUM = 0
  FOR I% = 1 TO (N% - K%)
    VALU = ((DATAA(I%) - MEANA) * (DATAO(I% + K%) -
MEANB))
    SUM = SUM + VALU
  NEXT I%
  COR(K%) = SUM / N%
  COR(K%) = COR(K%) / SQR(VARA * VARB)
NEXT K%

FOR K% = 0 TO -50 STEP -1
  SUM = 0
  FOR I% = 1 TO (N% + K%)
    VALU = ((DATAA(I%) - MEANB) * (DATAA(I% - K%) -
MEANA))
    SUM = SUM + VALU
  NEXT I%
  COR(K%) = SUM / N%
  COR(K%) = COR(K%) / SQR(VARA * VARB)
NEXT K%

IF CHOICE% = 1 THEN
  TEMP(1) = 1
  FOR K% = 2 TO 50

```

```

    TEMP(K%) = TEMP(K% - 1) + COR(K% - 1) ^ 2 * 2
    NEXT K%
    FOR K% = 1 TO 50
        TEMP(K%) = TEMP(K%) / N%
        ERM(K%) = -2 * SQR(TEMP(K%))
        ERP(K%) = 2 * SQR(TEMP(K%))
    NEXT K%
    ELSEIF CHOICE% = 2 THEN
        SUM = 0
        FOR K% = -50 TO 50
            SUM = SUM + AUTOA(K%) * AUTOB(K%)
        NEXT K%
        FOR K% = 0 TO 50
            TEMP(K%) = SUM / (N% - K%)
        NEXT K%
        FOR K% = -1 TO -50 STEP -1
            TEMP(K%) = SUM / (N% + K%)
        NEXT K%
        FOR K% = -50 TO 50
            ERM(K%) = -2 * SQR(TEMP(K%))
            ERP(K%) = 2 * SQR(TEMP(K%))
        NEXT K%
    END IF
    END SUB
    SUB GAUS (C%, MAT(), AUTO(), CROSS(), V(), FLAG%)
    FOR I% = 1 TO C%
        V(I%) = CROSS(I%)
    NEXT I%
    IF FLAG% = 1 THEN
        FOR I% = 1 TO C%
            FOR J% = 1 TO C%
                K% = ABS(I% - J%)
                MAT(I%, J%) = AUTO(K%)
            NEXT J%
            V(I%) = CROSS(I% - 1)
        NEXT I%
    END IF

```

```

    NEXT I%
    END IF
    L% = C%
    FOR I% = 1 TO L%
        M% = 1
        IF MAT(I%, I%) = 0 THEN
            FOR O% = 1 TO L%
                HOLD = MAT(I%, O%)
                MAT(I%, O%) = MAT(I% + M%, O%)
                MAT(I% + M%, O%) = HOLD
            NEXT O%
            HOLD = V(I%)
            V(I%) = V(I% + M%)
            V(I% + M%) = HOLD
            M% = M% + 1
            GOTO 90
        END IF
        DIV = MAT(I%, I%)
        FOR J% = 1 TO L%
            MAT(I%, J%) = MAT(I%, J%) / DIV
        NEXT J%
        V(I%) = V(I%) / DIV
    NEXT I%
    IF J% <> I% THEN
        XMULT = MAT(J%, I%)
        FOR K% = I% TO L%
            MAT(J%, K%) = MAT(J%, K%) - MAT(I%, K%) *
        NEXT K%
        V(J%) = V(J%) - V(I%) * XMULT
    END IF
    NEXT J%
    NEXT I%

```

```

END SUB

SUB VARI (N%, I%, DATAA(), MEAN, VAR)
SUM = 0
FOR J% = 1 TO N%
    SUM = SUM + DATAA(J%)
NEXT J%

MEAN = SUM / N%
SUM = 0
FOR J% = 1 TO N%
    SUM = SUM + (DATAA(J%) - MEAN) ^ 2
NEXT J%
VAR = SUM / N%

END SUB

```