2006

Development of an adaptive fuzzy logic controller for HVAC system

Rahul Laxman Navale
Iowa State University

Development of an adaptive fuzzy logic controller for HVAC system

by

Rahul Laxman Navale

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Mechanical Engineering

Program of Study Committee:
Ron M. Nelson, Major Professor
Michael Pate
Greg Luecke
Steve Hoff
Eric Bartlett

Iowa State University
Ames, Iowa
2006

Graduate College

Iowa State University

This is to certify that the doctoral dissertation of

Rahul Laxman Navale

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Major Professor

Signature was redacted for privacy.

For the Major Program

Table of Contents

List of Tables vii
List of Figures ix
Nomenclature xiii
List of Acronyms xvi
Acknowledgement xvii
Abstract xviii
Chapter 1 Introduction 1
1.1 Background/Literature Review 1
1.1.1 Fuzzy Logic Controllers 2
1.1.2 Summary 8
1.2 Project Objectives 8
1.3 The Adaptive FLC description 8
1.4 Proposed Adaptive Fuzzy Logic Controller - GA 9
1.5 Proposed Adaptive Fuzzy Logic Controller - ES 11
1.6 Plan of Work 12
1.7 Outline of Contents 13
Chapter 2 Experimental Facility Description 15
2.1 Introduction 15
2.2 Thermal Loads 16
2.3 Air Handling Unit 17
2.4 Cooling Plant 19
2.5 Data Acquisition 20
2.6 Bias Study for System A and System B 22
2.7 Chilled Water Valve Position Study 24
2.7.1 Introduction 24
2.7.2 Commanded and Real Chilled Water Valve Position 24
2.7.3 Chilled Water Flow Through the Cooling Coil 26
2.8 Summary 29
Chapter 3 Performance Indices 31
3.1 Introduction 31
3.1.1 Root Mean Square Error 31
3.1.2 Controller Performance Parameters 32
3.1.3 Hydronic Energy 33
3.1.4 Actuator Travel Distance 36
3.2 Summary 37
Chapter 4 Cooling Coil Models 38
4.1 Introduction 38
4.2 Model using Neural Networks 38
4.2.1 Introduction 38
4.2.2 Normalization 39
4.2.3 Training of Neural Networks Model 41
4.2.4 Summary 42
4.3 Model using General Regression Neural Networks 43

4.3.1 Introduction 43
4.3.2 GRNN Theory 43
4.3.3 GRNN Models 45
4.3.4 Summary 47
4.4 Model using Adaptive Neural Networks 47
4.4.1 Introduction 47
4.4.2 Summary 47
4.5 Model using Lumped Capacitance Method 47
4.5.1 Introduction 47
4.5.2 Lump Capacitance Model Development 48
4.5.3 Summary 57
4.6 Summary 57
Chapter 5 Development of a Fuzzy Logic Controller 58
5.1 Basics of Fuzzy Logic Controller 58
5.1.1 Fuzzification 58
5.1.2 Fuzzy Rule Matrix 60
5.1.3 Defuzzification Methods 60
5.2 Manual Tuning of FLC 61
5.2.1 Modifying FRM 61
5.2.2 Modifying Selective Rule in the FRM 63
5.2.3 Modifying Shape of Fuzzy Membership Functions: 64
5.2.4 Modifying Scaling Factor 65
5.3 Adaptive Fuzzy Logic Controller 66
5.3.1 Introduction 66
5.3.2 Adaptation Mechanism 67
5.3.3 Use of ES for Evolving Scaling Factor: 69
5.3.4 Use of ES for Evolving Mapping Factor 72
5.3.5 Use of GA for Evolving FRM 80
5.4 Summary 86
Chapter 6 Results for manually tuned FLC and SAT models 87
6.1 Results for Manual Tuned FLC 87
6.1.1 Real-time Results for 49 Rules FRM 87
6.1.2 Real-time test results for 121 rules FRM 90
6.1.3 Summary 94
6.2 Neural Network Model Results for predicting SAT 94
6.2.1 Offline Neural Network Models 94
6.2.2 Real-time Test Results for NN Models and Adaptive NN Model 104
6.2.3 Summary 110
6.3 GRNN Model Results for predicting SAT 110
6.3.1 Offline Test 111
6.3.2 Real-time Test 113
6.3.3 Summary 115
6.4 Lump Capacitance Models Results for predicting SAT 115
6.4.1 Offline Test 116
6.4.2 Real-time Test 117

6.4.3 Summary 119
6.5 Real-time Test Result Comparison for Predicting SAT using Different Models . 119
6.6 Summary 124
Chapter 7 Results for Adaptive Fuzzy Logic Controller (AFLC) 125
7.1 AFLC using Genetic Algorithms (AFLC-GA) 125
7.1.1 FRM obtained by Random Numbers Generated between 1 through 11 126
7.1.2 Changing Human FRM by Random Numbers Generated without changing Specific Elements 129
7.1.3 Changing Human FRM by Random Number Generated without changing Specific Elements in the Outer Loop 133
7.2 Summary 136
7.3 AFLC using Evolutionary Strategies (AFLC-ES) 137
7.3.1 Offline Test Results for AFLC-ES with evolving Scaling Factor 139
7.3.2 Real-time Test Results for AFLC-ES with evolving Scaling Factor 140
7.3.3 Offline Test Results for AFLC-ES with evolving Mapping Factor 141
7.3.4 Real-time Test Results for AFLC-ES with evolving Mapping Factor 142
7.4 Summary 143
Chapter 8 Comparison test results for AFLC-GA/ES and PIDL 145
8.1 Introduction 145
8.2 Comparison Test Results for Variation in SATSPT with AFLC-GA on AHUA . 148
8.2.1 RMS Error 150
8.2.2 Hydronic Energy 153
8.2.3 Actuator Travel Distance 155
8.2.4 Rise Time, Overshoot and Setting Time 156
8.3 Comparison Tests Results for Variations in SACFM with AFLC-ES on AHUA 161
8.3.1 RMS Error 166
8.3.2 Hydronic Energy 168
8.3.3 Actuator Travel Distance 169
8.3.4 Rise Time, Overshoot and Settling Time 173
8.4 Comparison test results for variation in SACFM using AFLC-ES on AHUB 175
8.4.1 RMS Error 176
8.4.2 Hydronic Energy 178
8.4.3 Actuator Travel Distance 179
8.4.4 Rise Time, Overshoot and Settling Time 184
8.5 Summary 185
Chapter 9 Conclusions 187
9.1 Details 187
9.2 Contributions 189
9.3 Recommendations 190

Appendix A ERS-Air Handling Unit and Cooling Coil Specifications 191
Appendix B DDE 195
Appendix C Fuzzy Logic 197
Appendix D Neural Networks 207
Appendix E Sensitivity Analysis 213

Appendix F Genetic Algorithms 218
Appendix G Evolutionary Strategies 223
Appendix H ERS System Test Setup 225
References 229

List of Tables

Table 1.1-1: Technologies in each category selected for further study 1
Table 1.1-2: Technologies in each category selected for further study continued 2
Table 2.2-1: Stages of lighting load 17
Table 2.2-2: Stages of baseboard heater thermal load 17
Table 2.6-1: Performance Indices for Hydronic Energy in Cooling Tests 23
Table 2.6-2: Performance Indices for Actuator Travel Distance in Cooling Tests 23
Table 2.7-1: Commanded and real chilled water valve position 25
Table 2.7-2: Flow ratio for different chilled water valve position 28
Table 4.3-1: GRNN Models 46
Table 5.1-1: Default 7 x 7 Fuzzy Rule Matrix 60
Table 5.2-1: Default 9 x 9 Fuzzy Rule Matrix 62
Table 5.2-2: Default 11 x 11 Fuzzy Rule Matrix 63
Table 5.2-3: Modified Fuzzy Logic Rule Matrix (11 by 11) 64
Table 5.3-1: Problem Representation of scaling factor problem for ES 77
Table 5.3-2: 7 x 7 Human FRM 80
Table 5.3-3: Modified 7 x 7 Human FRM 81
Table 6.2-1: Offline test results for retraining and adaptive trained neural network model. 100
Table 6.3-1: Results for Offline Tests using GRNN Models for Predicted SAT 111
Table 6.3-2: Offline tests results for predicting SAT using GRNN Models 112
Table 6.3-3: RMS error for predicting SAT using GRNN models II and III 115
Table 6.4-1: RMS error for offline test in predicting SAT using different LCMs 117
Table 6.4-2: RMS error for real time study using LCM IV 119
Table 6.5-1: RMS error for real-time test in predicting SAT using different models 121
Table 7.1-1: Numeric representations of control action (u) for each FMF 125
Table 7.1-2: Best FRM generated by random numbers for offline test 126
Table 7.1-3: Best FRM generated by random numbers for real-time test 128
Table 7.1-4: FRM generated by changing Human FRM by random numbers 130
Table 7.1-5: FRM generated by changing Human FRM without changing specific elements from the Human FRM 132
Table 7.1-6: FRM generated by modifying elements in the Human FRM without changing elements in the outer loop 134
Table 7.1-7: FRM generated without changing elements in the outer loop of the Human FRM for real-time study 135
Table 7.3-1: Maximum and Minimum values for each mapping factor 138
Table 7.3-2: RMS error for real-time test with ES evolving scaling factor 140
Table 7.3-3: RMS error resulted for real-time study of scaling factor 143
Table 8.2-1: Comparison of RMS error/cycle between S for variation in SATSPT test 151
Table 8.2-2: Comparison of hydronic energy/cycle for variation in SATSPT test 153
Table 8.2-3: Comparison of actuator travel distance/cycle for variation in SATSPT test ... 155
Table 8.2-4: Case I - Comparison ofcontrol parameters for variation in SATSPT test 158
Table 8.2-5: Case II - Comparison of control parameters for variation in SATSPT test 159
Table 8.2-6: Case III – Comparison of control parameters for variation in SATSPT test ... 160
Table 8.3-1: Comparison of average hourly RMS error for variation in SACFM test 166
Table 8.3-2: Comparison of average hydronic energy/hr for variation in SACFM test 168

Table 8.3-3: Comparison of total actuator travel distance/hr for variation in SACFM test . 170
Table 8.3-4: Comparison of performance indices for variation in SACFM test 172
Table 8.3-5: Percentage difference in performance indices for variation in SACFM test ... 172
Table 8.4-1: Comparison of average hourly RMS error for variation in SACFM test..... 176
Table 8.4-2: Comparison of average hydronic energy/hr for variation in SACFM test..... 178
Table 8.4-3: Comparison of total actuator travel distance/hr for variation in SACFM test . 180
Table 8.4-4: Comparison of performance indices for variation in SACFM test 182
Table 8.4-5: Percentage difference in performance indices for variation in SACFM test ... 182
Table 8.4-6: Percentage difference in hydronic energy/day consumption 183

Table A-1: AHU Specification 191
Table A-2: Cooling Coil Specifications..... 191
Table A-3: ERS data points and accuracy for the sensors used..... 192
Table C-1: Truth tables for AND, OR, and NOT operators. 198
Table C-2: Sample Data from the ERS AHUA 204
Table E-1: Base Case and Variation in each input for Sensitivity Analysis study 215
Table E-2: Result for Sensitivity Analysis Study 216

List of Figures

Figure 1.4-1: Proposed Adaptive Fuzzy Logic Controller - GA 10
Figure 1.5-1: Proposed Adaptive Fuzzy Logic Controller - ES 11
Figure 2.1-1: Schematic of Energy Resource Station, Ankeny, Iowa. 16
Figure 2.3-1: Schematic of Air Handling Unit 18
Figure 2.4-1: Schematic of Cooling Coils 20
Figure 2.5-1 : Data Acquisition System 21
Figure 2.7-1: Commanded and real chilled water valve position 24
Figure 2.7-2: Commanded Vs actual chilled water valve position 25
Figure 2.7-3: Details for chilled water flow through cooling coil and bypass. 26
Figure 2.7-4: Flow ratio of chilled water through the cooling coils to bypass. 28
Figure 2.7-5: Variation in the ratio for chilled water flow through cooling coil to bypass 29
Figure 3.1-1: Definition of Rise Time, Settling Time and Overshoot 32
Figure 3.1-2: Details for AHU Cooling Coil Instrumentation 33
Figure 4.2-1: Neural Networks 38
Figure 4.3-1: GRNN Model Architecture 45
Figure 4.5-1: Schematic of Cooling Coil (CC) 49
Figure 4.5-2: Steps in development of LCM 53
Figure 4.5-3: Flow chart to find optimal values of constants used in LCM 55
Figure 5.1-1: Fuzzy Membership Function (FMF) for Error 59
Figure 5.1-2: Fuzzy Membership Function for Derivative of Error 59
Figure 5.1-3: Fuzzy Membership Function for Control Signal 59
Figure 5.2-1: Modifying shapes of the fuzzy membership functions 65
Figure 5.2-2: Effect of modifying Scaling Factor 66
Figure 5.3-1: Flow chart for Evolutionary Strategies 69
Figure 5.3-2: Flow chart for evolving scaling factor using ES 70
Figure 5.3-3: Representation of scaling factor problem for ES 71
Figure 5.3-4: Mapping of uniform FMF to non-uniform FMF 73
Figure 5.3-5: Defining FMF in three variables 73
Figure 5.3-6: Uniform fuzzy membership functions in error and derivative of error 74
Figure 5.3-7: Flow chart for evolving mapping factor using ES 76
Figure 5.3-8: Mapping of Uniform FMFs to Non-uniform FMFs in error 79
Figure 5.3-9: Flow chart for GA 82
Figure 5.3-10: Illustration for GA search results 84
Figure 6.1-1: Real-time test results for FLC using default 49 rules FRM 87
Figure 6.1-2: Real-time test results for FLC using default 49 rule FRM 88
Figure 6.1-3: Case I – Real-time test results for FLC using modified 49 rules FRM 89
Figure 6.1-4: Case II – Real-time test results for FLC using modified 49 rules FRM 90
Figure 6.1-5: Real-time test results for FLC using default 11 x 11 rules FRM 91
Figure 6.1-6: Case I - Real-time test results for FLC 92
Figure 6.1-7: Case II - Real-time test results for FLC 92
Figure 6.1-8: Real-time test results for FLC using modified 11 by 11 elements FRM 93
Figure 6.2-1: Offline test results using NN model with 18 inputs 96
Figure 6.2-2: Offline test results using NN model having derivative information 97

Figure 6.2-3: Offline test result using NN model with 8 inputs for unseen data 98
Figure 6.2-4: Offline test results using retraining & adaptive NN model 101
Figure 6.2-5: Offline test results using retraining NN model 102
Figure 6.2-6: Offline test results using adaptive NN model 102
Figure 6.2-7: Offline test result using NN model with 8 inputs having erroneous data 103
Figure 6.2-8: Offline test result using NN model with 8 inputs without erroneous data..... 104
Figure 6.2-9: Real-time test results using NN model with 18 inputs..... 105
Figure 6.2-10: Real-time test results using NN model with derivative information 106
Figure 6.2-11: Case I - Real-time test results using NN model using 8 inputs..... 107
Figure 6.2-12: Case II - Real-time test results using NN model using 8 inputs 107
Figure 6.2-13: Real-time test results for predicting SAT using adaptive NN model 108
Figure 6.2-14: Real-time test results for predicting SAT using adaptive NN model 109
Figure 6.3-1: Offline test results for predicting SAT using GRNN models I and II 112
Figure 6.3-2: Offline test results for predicting SAT using GRNN model I, II and III..... 113
Figure 6.3-3: Variation in the inputs for real-time testing of GRNN model II and III..... 114
Figure 6.3-4: Real-time test results for predicting SAT using GRNN models II and III 114
Figure 6.4-1: Difference between SAT_{PRED} and SAT for different LCMs..... 116
Figure 6.4-2: Variation in the inputs for real-time test using LCM IV..... 118
Figure 6.4-3: LCM IV predicted SAT and actual SAT for real time test 118
Figure 6.5-1: Variation in the inputs for SAT predicting models study 120
Figure 6.5-2: Real-time test results for predicted SAT using GRNN I and GRNN II..... 120
Figure 6.5-3: Errors for real-time tests in predicting SAT using different models..... 121
Figure 6.5-4: Real-time results for predicting SAT using LCM IV and GRNN model II.... 122
Figure 6.5-5: Real-time results for predicting SAT using LCM IV and GRNN model II.... 123
Figure 6.5-6: Real-time results for predicting SAT using LCM IV and GRNN model II.... 123
Figure 7.1-1: Offline test results for the best generated FRM by random numbers 127
Figure 7.1-2: Real-time test results using FRM generated by random numbers 129
Figure 7.1-3: Offline test results for FRM generated by changing Human FRM by random numbers generated amongst –1, 0, and 1 without changing specific elements 131
Figure 7.1-4: Real-time test results using FRM generated by changing Human FRM using random numbers generated amongst –1, 0, and 1 without changing specific elements 133
Figure 7.1-5: Offline test results using FRM generated by modifying elements in the Human FRM without changing elements in the outer loop..... 134
Figure 7.1-6: Real time test results for AFLC-GA with FRM generated by modifying the Human FRM without changing elements in the outer loop of the Human FRM 136
Figure 7.3-1: Offline test results using scaling factor..... 140
Figure 7.3-2: Real time test results with varying scaling factor. 141
Figure 7.3-3: Offline test results for AFLC with evolving mapping factor..... 142
Figure 7.3-4: Real-time test results for AFLC-ES with evolving mapping factor..... 143
Figure 8.1-1: Lighting load schedule in the zones..... 147
Figure 8.1-2: Baseboard load schedule in the zones..... 147
Figure 8.1-3: Total load schedule in the zones 148
Figure 8.2-1: Comparison test results for variation in SATSPT..... 149
Figure 8.2-2: Zoom-in comparison test results for variation in SATSPT..... 149
Figure 8.2-3: Zoom-in comparison test results for variation in SATSPT..... 150

Figure 8.2-4: Comparison of RMS error/cycle for variation in SATSPT test 152
Figure 8.2-5: Comparison of RMS error/cycle with trend line for variation in SATSPT 152
Figure 8.2-6: Comparison of average hydronic energy/cycle for variation in SATSPT test 154
Figure 8.2-7: Comparison of actuator travel distance/cycle for variation in SATSPT test .. 156
Figure 8.2-8: Variation in SAT for AHUA (AFLC-GA) and AHUB (PIDL) 157
Figure 8.2-9: Case I - Comparison of control parameters for variation in SATSPT test 158
Figure 8.2-10: Case II - Comparison of control parameters for variation in SATSPT test .. 159
Figure 8.2-11: Case III - Comparison of control parameters for variation in SATSPT test. 160
Figure 8.3-1: SACFM schedule 161
Figure 8.3-2: Comparison test results for variation in SACFM 162
Figure 8.3-3: Zoom-in comparison test results for variation in SACFM 162
Figure 8.3-4: Zoom-in comparison test results for variation in SACFM 163
Figure 8.3-5: Variation in SATA (AFLC-ES) and SATB (PIDL) and SACFM schedule ... 163
Figure 8.3-6: SATA and SATB along with variation in the SACFM for 3rd test day 164
Figure 8.3-7: SATA and SATB along with variation in the SACFM for 4th test day..... 164
Figure 8.3-8: SATA and SATB along with variation in the SACFM for 5th test day 165
Figure 8.3-9: SATA and SATB along with variation in the SACFM for 6th test day 165
Figure 8.3-10: Comparison of average hourly RMS error for variation in SACFM test..... 167
Figure 8.3-11: Comparison of average hydronic energy/hr for variation in SACFM test.... 169
Figure 8.3-12: Comparison of actuator travel distance/hr for variation in SACFM test 171
Figure 8.3-13: Percentage difference in performance indices for variation in SACFM test 173
Figure 8.3-14: Case I – Comparison of control parameters for variation in SACFM test.... 174
Figure 8.3-15: Case II - Comparison of control parameters for variation in SACFM test... 174
Figure 8.3-16: Case II – Comparison of control parameters for variation in SACFM test .. 175
Figure 8.4-1: Comparison of average hourly RMS error for variation in SACFM test..... 177
Figure 8.4-2: Comparison of average hydronic energy/hr for variation in SACFM test..... 179
Figure 8.4-3: Comparison of total actuator travel distance/hr for variation in SACFM test 181
Figure 8.4-4: Percentage difference in performance indices for variation in SACFM test .. 183
Figure 8.4-5: Comparison of control parameters for variation in SACFM test..... 184

Figure C-1: Triangular Membership Functions 199
Figure C-2: Basic configuration of Fuzzy Logic System 200
Figure C-3: Centroid Defuzzification Method..... 202
Figure C-4: Membership Functions for Error..... 203
Figure C-5: Membership Functions for derivative of error 203
Figure C-6: Membership Functions for Control Signal..... 203
Figure C-7: Defuzzification of Output (Centroid Method)..... 205
Figure D-1: Simple Neuron Network 207
Figure D-2: Hard-Limit Transfer Function..... 208
Figure D-3: Liner Transfer Function 208
Figure D-4: Tan-Sigmoid Transfer Function..... 209
Figure D-5: Neuron with Multiple Inputs..... 209
Figure D-6: Generalized Neural Network with a single Hidden Layer Architecture 209
Figure D-7: General Function Approximator 211
Figure E-1: Training Data Distribution - I..... 214

Figure E-2: Training Data Distribution - II 214
Figure E-3: Results for Sensitivity Analysis Study 216
Figure F-1: Flow chart for GA 219

Nomenclature

C	heat capacity rate, (Btu/lb_m-°F);
c_p	specific heat, (Btu/lb_m -°F)
C_r	heat capacity ratio
d	derivative of error (°F/sampling period)
D_i	distance function of the input space
e	error (°F)
f(x, y)	continuous probability density function
h	enthalphy, Btu/lb_m
i	sample data point
k_e, k_i	starting time, ending time
M_p	Overshoot (seconds)
n	initial guesses
n	umber of data patters/points
n_{da}	number of moles of dry air
n_w	number of moles of water vapor
p	total mixture pressure
p_{da}	partial pressure of dry air
p_ω	partial pressure of water vapor
$p_{\omega s}$	pressure of water vapor in saturated moist air
$p_{\omega sin}$	water vapor saturation pressure, psia;
q	heat transfer rate, Btu/hr
Q	volumetric flow rate, ft^3/min
$\dot{Q}_{bypass}$	Chilled water flow through bypass section (gpm)
$\dot{Q}_{coil}$	Chilled water flow through the cooling coil (gpm)
$\dot{Q}_{flow}$	chilled water flow from pump (gpm)
R	universal gas constant, 1545.32 ft*lb_f/lb mol*°R
RMS $error_o$	RMS error for offspring gene
RMS $error_p$	RMS error for parent gene

SFe_{max}	maximum value of scaling factor
SFe_{min}	minimum value of scaling factor
SFe_{update}	Updating value of scaling factor
T	Temperature (°F)
^{T}	Matrix Transpose
T_{abs}	Temperature (°R)
t_r	rise time (seconds)
t_s	settling time (seconds)
u	control action (% change in the chilled water valve position)
UA	overall heat transfer coefficient, Btu'hr-°F
v	specific volume of air leaving from chilled water cooling coil, ft^3/lb_m
V	volume, ft^3
VLV	actuator position, % Open
X, Y	measured values
x,y	measured variables
β_I	design variables
ε	effectiveness
σ	sample probability of width
φ	relative humidity
ω	humidity ratio lb_{water}/lb_{dryair}
Ω	ohms
^	vector
^	power

<u>subscripts</u>

a	air
da	dryair
min	minimum
max	maximum
ma	mixed air
ra	return air

sa	supply air
ss	steady state
w	water

Experimental Nomenclature

AHUA	Air Handling Unit
CC	cooling coil
CHWC-DAT	cooling water coil discharge air temperature (°F)
CHWC-EWT/EWT	entering chilled water temperature into the cooling coils (°F)
CHWC-LWT/LWT	leaving chilled water temperature from the cooling coils (°F)
CHWC-MWT/MWT	mixed chilled water temperature (°F)
CHWC-VLV/VLV	chilled water coil value position (% Open)
CHWP-GPM/GPM	chilled water pump water flow rate (gpm)
EA-DMPR	exhaust air damper (% Open)
EAT/ OA-TEMP	air temperature entering cooling coil (°F)
HWC-DAT	heating water coil discharge air temperature (°F)
LAT	air temperature leaving cooling coil (°F)
MAT	mixed air temperature (°F)
OA-CFM	outside air flow rate (cfm)
OA-DMPR	outside air damper (% Open)
OAT/OA-TEMP	outside air temperature (°F)
RA-CFM	return air flow rate (CFM)
RA-DMPR	return air damper (% Closed)
RAT/ RA-TEMP	return air temperature (°F)
SACFM/CFM	supply air flow rate (cfm)
SACFM-SPT	supply air flow rate set point (cfm)
SAT	supply air temperature (°F)
SATA	supply air temperature for AHUA (°F)
SATB	supply air temperature for AHUA (°F)
SATSPT	Supply air temperature set point (°F)

List of Acronyms

EMCS Energy Management and Control System

ERS Energy Resource Station

ES Evolutionary Strategies

FL Fuzzy Logic

FLC Fuzzy Logic Controller

FLC Fuzzy Logic Controller/Control

FMF Fuzzy Membership Functions

FRM Fuzzy Rule Matrix

GA Genetic Algorithms

GRNN General Regression Neural Networks

LCM Lump Capacitance Model

NL Negative Large fuzzy membership function

NM Negative Medium fuzzy membership function

NN Neural Networks

NS Negative Small fuzzy membership function

NVS Negative Very Small fuzzy membership function

PIDL Proportional Derivative and Integral Loop Controller/Control

PIDL Proportional Integral Derivative loop

PL Positive Large fuzzy membership function.

PL Proportional and Integral Controller/Control

PM Positive Medium fuzzy membership function

PS Positive Small fuzzy membership function

PVS Positive Very Small fuzzy membership function

RMS Root mean square

RTD Resistance Temperature Detectors

SF Scaling Factor

TES Thermal Energy Storage

VAV Variable air volume

ZE Zero fuzzy membership function

Acknowledgement

I gratefully thank my advisor Professor Ron M. Nelson for his helpful advice, ceaseless patience, endless support, and heartening encouragement throughout the course of this study. Without his inspiration, this study would not be completed.

I would like to give acknowledgement for the valuable assistance from the people at the Iowa Energy Center and Energy Resource Station including Curt Klaassen, David Perry, and Xiao Hui Zhou. The financial support of the Iowa Energy Center for this research project is sincerely acknowledged. I would also like to thank the members of my thesis committee for their contribution to this work.

I appreciate my teachers, my colleagues and my friends for their helpful discussions, suggestions and encouragement.

Finally, I give my most special thank and express my deepest gratitude towards my parents, my brother, Atul, and my beloved wife, Swarada, for their constant love, patience, care, support, understanding and presence throughout this study and support in my life.

Abstract

An adaptive approach to control a cooling coil chilled water valve operation, called adaptive fuzzy logic control (AFLC), is developed and validated in this study. The AFLC calculates the error between the supply air temperature and supply air temperature set point for air in an air handling unit (AHU) of a heating, ventilating, and air conditioning (HVAC) system and determines optimal fuzzy logic parameters to minimize the error between the supply air temperature and its set point. The AFLC uses genetic algorithms and evolutionary strategies to determine the fuzzy rule matrix and fuzzy membership functions for an AHU in HVAC systems. It is shown that the AFLC can reduce hydronic energy consumption while maintaining occupant comfort.

Cooling coil models are developed using neural network, general regression neural network and lump capacitance methods to predict the supply air temperature. These models helped with the development of the adaptive fuzzy logic controller.

Two types of validation experiments were conducted, one with cyclically changing supply air temperatures and the second with cyclically changing supply air flow rates. Experiments conducted on two identical real HVAC systems were used to compare the performances of the AFLC to a conventional proportional, integral and derivative (PID) controller. To remove bias between the testing systems, the controllers were switched from one system to the other.

The validation experiments indicate that the HVAC system operated under the AFLC consumes 1 to 7 % less hydronic energy when compared with a conventional PID controlled system. More actuator travel distance was observed when using the AFLC. The AFLC maintained better occupant comfort conditions when compared with the conventional PID controller. It was observed that the controlled variable for the AFLC system required 0 to 185% more rise time, had 9 to 68% less overshoot and required 11 to 45% less settling time as compared to the conventional PID controlled system.

Chapter 1 Introduction

1.1 Background/Literature Review

The increasing demand for energy efficient buildings has created a need for more energy efficient heating, ventilation, and air-conditioning (HVAC) equipment and systems along with the need to operate and control these systems in the most energy efficient manner. Space heating and cooling is the largest energy expense in most homes, accounting for more than 44% of the average home's utility bill (EERE, 2005). About 32% of the electricity generated in the United States is consumed to heat, cool, ventilate and light commercial buildings (ASHRAE 2000).

U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy published a report on energy savings potential for building technologies program in July 2002. Out of 55 different technologies studied, 15 technologies, shown in Table 1.1-1, were selected for further study based on energy saving potential and the value of further study toward improving estimates of ultimate market-achievable energy savings potential, notably energy savings potential, current and potential future economics, and key barriers facing each option.

Table 1.1-1: Technologies in each category selected for further study (EERE, 2002)

Category	Technology for further study
Component	Electronically Commutated Permanent Magnet Motor (ECPM)
	Improved Duct Sealing
	Microchannel Heat Exchanger
	Smaller Centrifugal Compressors
Equipment	
	Enthalpy/Energy Recovery Heat Exchangers for Ventilation
	Heat Pump for Cold Climates
	Liquid Desiccant Air Conditioner

Table 1.1-2: Technologies in each category selected for further study continued

Systems	
	Dedicated Outdoor Air Systems
	Displacement Ventilation
	Microenvironment (Task-ambient Conditioning)
	Novel Cool Storage
	Radiant Ceiling Cooling/Chilled Beam
	Variable Refrigerant Volume/Flow
Controls / Operations	
	Adaptive/Fuzzy Logic HVAC Control
	System/Component Performance Diagnostics

Adaptive/Fuzzy Logic Controls was one of these 15 technologies and had a technical energy savings potential of 0.23 quads which is about 5% of HVAC energy use (EERE, 2002).

Several studies have been reported about the application of adaptive fuzzy logic control. This literature review focuses on the development and validation of adaptive fuzzy logic controllers, self-tuning methods, artificial neural networks, and genetic algorithms. Studies that relate to building energy applications are emphasized.

1.1.1 Fuzzy Logic Controllers

Fuzzy Logic Controllers (FLCs) are based on a set of fuzzy control rules which make use of people's common sense and experience. The fuzzy logic used in these controllers is very well defined mathematically and is able to take into account the uncertainty in the knowledge used to develop the rules and the uncertainty in the operating conditions. The initial rules for FLCs come from the experience of people who work with the systems that are to be controlled. This experience is obtained in the form of linguistic rules such as "If the temperature is too low, open the hot water valve" and "If the temperature is too low and going down fast, open the hot water valve very quickly", etc. These initial rules are then used to construct a fuzzy control matrix that implements the logic of the controller.

The initial fuzzy control matrix and other FLC parameters need to be refined using adjustment strategies that are usually based on manual trial and error methods to achieve improved performance (Huang and Nelson 1991, 1993, 1994a, 1994b). Several researchers have studied and implemented self-tuning or adaptive fuzzy logic controllers (AFLCs) that improve their performance as they adjust to the controlled process and the environment. The operation of an AFLC relies on past experience that looks at suitable combinations of control strategies (control rules, membership functions, and scale factors) and the effects they produce. A particular feature of AFLCs is that they automatically improve their performance until they converge to a predetermined optimal condition.

The first AFLC suggested evaluating and modifying the control rules by a self-organizing algorithm (Mamdani 1979). Several other types of AFLCs have been proposed by researchers (Xu 1987, Shao 1988, Tansheit 1988, Acosta 1992, and Lee 1992). Most of these AFLCs use the tracking error and derivative of error at every sampling instant as the basis not only for the control algorithm but also for the adaptive algorithm. The control action is improved as it is created. The control outputs often become bang-bang signals and convergence of the modifying process is usually slow.

A self-tuning controller for HVAC systems was presented by Wallenborg (1991). A new algorithm for a self-tuning controller was developed in which a discrete time process transfer function was calculated from the wave form of a periodic oscillation obtained with a relay feedback tuning experiment. A general linear discrete time controller was designed using pole placement based on an input-output model. A substantial reduction in commissioning time was achieved compared with manual tuning of conventional controllers.

A computer simulation was studied by Huang and Nelson (1991) to test a new PID control algorithm developed using a fuzzy rule based system. The results showed impressive performance of the PID law combining fuzzy controller for HVAC applications.

Self tuning control with fuzzy rule based supervision for HVAC applications was studied by Ling et al. (1991). The performance of combinations of fuzzy rule based supervisor and self tuning controller was evaluated using a detailed component based simulator developed to test building energy management systems.

Rule development and adjustment strategies of a fuzzy logic controller (FLC) for an HVAC system were analyzed and experimentally evaluated by Huang and Nelson (1993, 1994a, 1994b) for controlling a pneumatic valve to control the temperature of hot water to a hot water coil.

Some modern control strategies were applied to a HVAC system. Zaheer-uddin (1993) used a multivariable control design in a water heating system. The optimal regulator design was applied for damper control of the supply air to a single zone. The sub-optimal operation was employed in a single zone with a heating coil for a constant air volume (CAV) system. The model reference adaptive control was applied to a single zone with a cooling coil for a variable air volume (VAV) system (Zaheer-uddin 1993). Simulation results showed that these modern control strategies were stable.

The neural network method was applied to a HVAC system for the first time by Curtiss et al. (1993, 1994 and 1996). A hot water coil that was used to heat air was modeled by two types of ANNs and a predictive controller was combined with the ANN to develop a future adaptive neural network (FANN) control. Simulation and experimental results show that the FANN control has a better behavior when compared with PID control. Work of Curtiss was expanded by Jeannette et al. (1998) to test the predictive Neural Network (PNN) in a real building. More experiments were conducted to obtain good control of the hot water supply temperature using a predictive NN. Results for energy savings were not reported.

A self-tuning control algorithm was developed by Ozsoy (1993). The algorithm combined the recursive parameter estimation method with control algorithms. The algorithm was applied to a single environmental space with an air heater, humidifier and ventilator system.

Simulation results showed that the algorithm had a robust behavior. Results for energy savings were not reported.

Arima (1995) applied fuzzy logic and rough sets for heating and humidification controls. Experimental results showing better performance of fuzzy logic reasoning are presented. No information on how the fuzzy rule matrix was developed or how they were tuned is presented.

Ying-guo Piao et al. (1995) and Arima et al. (1995) developed an FLC for controlling the temperature of hot water provided to a heating coil. Results for energy savings were not presented.

So et al. (1995) introduced a controller using a system identification method. The controller was able to predict the new system status based on the past records and suggested optimal control actions. Simulation results showed that the proposed controller behaved better than a conventional PID controller when both controllers were used to control an AHU system. Energy savings of about 6.62% were demonstrated by using the proposed controller.

A new pattern recognition adaptive controller that had the ability to self-tune PID parameters was described by Seem (1998). The controller was computationally efficient and provided near optimal performance. The controller was successfully used in many real buildings and showed good control of the HVAC system. Results for energy savings were not reported.

An adaptive learning algorithm based on genetic algorithms (GAs) for automatic tuning of PID controllers in HVAC systems to achieve optimal performance was presented by Huang (1997). The simulation results showed that genetic algorithm based optimization procedures are useful for automatic tuning of PID controllers for HVAC systems, yielding minimum overshoot and minimum settling time.

Velez-Reyes and Arguello-Serrano (1999) presented a nonlinear controller with a thermal load estimator for the HVAC system. The controller was capable of identifying time varying thermal loads. This capability enabled the controller to have better control when the system endured a large thermal load disturbance. The controller was used in a HVAC system. Better control of the nonlinear controller was demonstrated by simulation results. Results for energy savings were not reported.

Jian (2000) developed an Adaptive Neuro-Fuzzy (ANF) method for the supply air pressure control loop of a HVAC system. The ANF controller was developed to overcome the weakness of a well-tuned PID controller, which performs well around normal working points, but its tolerance to process parameter variations was severely affected due to the limitations of PID controllers for systems that have larger dead times or large parameter variations. No information on the neural network was made available. It was not clear if the controller used neural networks with multiple outputs for updating each membership function parameter in the fuzzy logic part. Also, the membership functions used in the fuzzy logic part were not specified. Also, it was not clear if the neural network was updating membership functions in every step or after a fixed number of steps. Simulation performances were compared without any experimental results.

Osman (2000a and 2000b) developed a feedforward controller using a General Regression Neural Network (GRNN) for controlling the temperature in a laboratory room. This HVAC system used a VAV system. No details about the GRNN model and the data used for training were presented. Simulation results show that by adjusting damper position and supply flow rate, the required room temperature can be achieved.

Wang Qing-Guo (2000) developed an auto-tuning multivariable controller for HVAC systems. In this paper, an advance PID auto-tuner for both single and multi-variable processes was developed with its application for controlling the zone temperature. No cost saving or energy consumption information was presented. No comparison of the PID auto-

tuner to a conventional PID controller was made. Experiments were conducted in a HVAC pilot plant.

Kolokotsa (2001) has developed an advance fuzzy proportional derivative (PD) controller for adjustment and preservation of air quality, thermal and visual comfort for buildings' occupants and achieved a reduction in energy consumption. The adaptive fuzzy PD controller adapts the input and output scaling factors and was based on a second order reference model. The buildings' response to the control signals was simulated using MATLAB/SIMULINK without any experimental results.

Alcala (2003) presented a smartly tuned FLC for an HVAC application which was developed using genetic algorithms. The genetic algorithm was used to obtain better fuzzy rule sets. The problem considered was very particular and complex. Simulation and experimental results showed energy savings. No information related to the test conditions were presented and the tuning method used is very complex.

More work is needed to develop AFLCs for HVAC applications that typically have long (about a minute or so) system time constants and periodically change operating modes (i.e. heating to cooling, using setback, etc.). Having a good adaptive or self-tuning algorithm for fuzzy logic controllers would increase their use, since considerable time is needed to manually tune FLCs. Properly tuned FLCs can be superior to conventional control algorithms (such as Proportional, Integral and Derivative (PID)), because they respond more quickly to setpoint and environmental changes and have minimal overshoot of the controlled variables. The controlled systems reach steady state faster with reduced oscillations about the set point. The quick response improves comfort and, since the systems spend less time in transient operation, energy consumption is reduced. The control logic can also implement electric demand and energy saving measures.

1.1.2 Summary

The literature review in the areas of adaptive fuzzy logic and self tuning controller reveals adaptive techniques that have been applied to both HVAC unit control and system control along with the other industrial problems (Wallenborg 1991, Ling and Dexter, 1991, So et al., 1995). Results for an adaptive controller show stable behavior and better performance compared with conventional control algorithms.

Combining adaptive techniques with optimization techniques leads to minimum overshoot and minimum settling time, thus saving energy (Huang and Nelson, 1999). In recent years, adaptive techniques have been used successfully in simulation studies for HVAC operation.

Several studies (So et al. 1995, Ling and Dexter, 1991, Wang and Jin, 2000) using adaptive control in the HVAC area have demonstrated that adaptive control leads to reduction in energy costs in comparison to conventional control strategies while still maintaining comfort conditions for building occupants.

The literature indicates a lack of applications of simple adaptive techniques that are combined with optimal operation for cooling coil in HVAC systems. Furthermore, most of the current studies stop at the simulation level and do not present results for experiments and energy savings data.

1.2 Project Objectives

The objective of this project is to develop, implement, and test an adaptive fuzzy logic controller on one of the air handling units at the Iowa Energy Center's (IEC) Energy Resource Station (ERS) located in Ankeny, Iowa.

1.3 The Adaptive FLC description

Over the past couple of years, fuzzy logic has played an important role in the implementation of controllers for various processes. FLCs are effective for controlling complex and poorly defined processes as they embody the knowledge of human experts to achieve good control strategies. Adaptive fuzzy logic control systems have been designed for various dynamic

processes. In a FLC, however, it is difficult to construct good rule bases and membership functions.

Some studies have been reported that use Genetic Algorithms (GAs) for the optimization of FLC performance by overcoming the above mentioned difficulties. Karr (1991) has used genetic algorithms to select good membership functions for a FLC that controlled a computer simulated cart-pole balancing problem. Genetic algorithms were used to alter the shape of fuzzy sets used in a given rule base in which each fuzzy label was coded as a 7-bit binary number.

Recently, Gurocak et al. (1999), Park et al. (2000) have shown the use of genetic algorithms for tuning fuzzy logic controllers for the inverted pendulum problem. Kindel et al. (1994) used modified genetic algorithms for designing and optimizing fuzzy controllers for the cart pole system. The task was divided into rule base modification and tuning of fuzzy membership function shape. They used non-binary coding and coded the rule base as matrix type chromosomes instead of general gene type chromosomes. Borut et al. (1999) used genetic algorithms to tune 25 consequent parameters keeping the membership functions the same. The efficiency of this approached was verified and validated on a hydraulic control system. Park et al. (1995) used genetic algorithms for fine tuning the fuzzy logic controller. The effectiveness of this method was verified through a series of simulations for a water level control system.

1.4 Proposed Adaptive Fuzzy Logic Controller - GA

Figure 1.4-1 shows a schematic of a proposed adaptive fuzzy logic controller using genetic algorithms (AFLC-GA). In this project, Genetic Algorithms (GAs) are used to modify the fuzzy logic controller's rule matrix. Total RMS error between the supply air temperature (SAT) and supply air temperature set point (SATSPT) is calculated for each Fuzzy Rule Matrix (FRM). This total RMS error value is used to evaluate the performance of all the generated FRMs. FRMs that have better FLC performance are reproduced to find improved FRMs. Thus, with the repeated application of GAs, near-optimal fuzzy logic controllers can

be achieved (See Appendix E for details).All the programming required for FLC and GA was done using Matlab software (Matlab 2001).

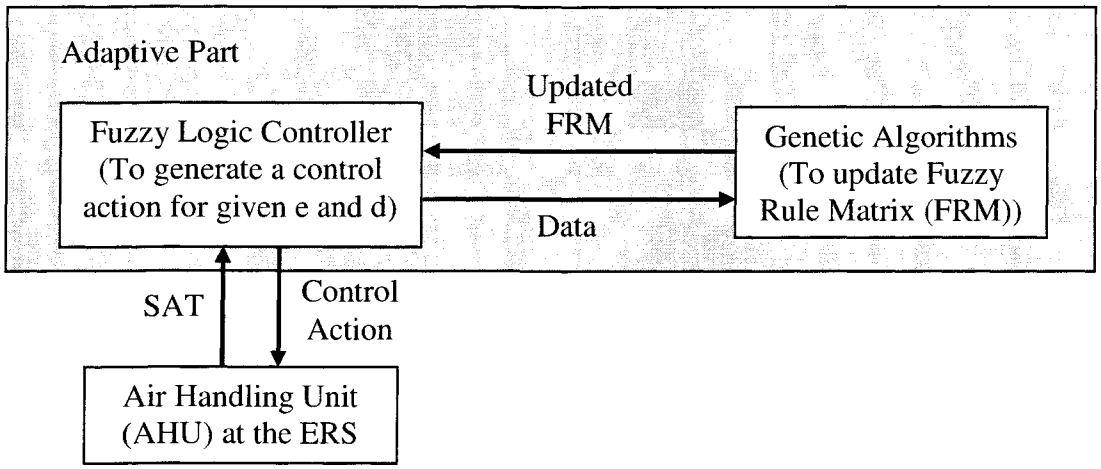

Figure 1.4-1: Proposed Adaptive Fuzzy Logic Controller - GA

As shown in Figure 1.4-1, for a set of error (e) and derivative of error (d), a control signal will be generated by the FLC. Depending upon the control signal calculated using the current FRM; the chilled water valve position is altered thus changing the SAT. For the offline study, SAT values were calculated using a general regression neural network (GRNN) model for the cooling coil instead of the real Air Handling Unit block as illustrated in Figure 1.4-1. The GRNN model was also developed using Matlab's NN toolkit. After a specified number of iterations of the control cycle are completed, the total RMS error corresponding to the FRM being used is calculated. Similarly, the total RMS error for each FRM generated for the study is calculated to evaluate the FLC performance.

Based on the total RMS error, all the generated FRMs are then sorted with the FRMs having minimum RMS error being best. After the sorting process, the two top most FRMs having minimum total RMS error are used for the reproduction of two new FRMs called offspring. The total RMS error is calculated for these offspring as mentioned above. Depending upon the total RMS error for these offspring namely, none, one or both the worst FRMs from the selected tournament (tournament is selecting either 4 or 7 FRMs randomly from the total generated FRMs) are replaced by the offspring FRMs. This process of generating new FRMs

(i.e. offspring) is continued for a pre-defined number of cycles, called generations. Thus, the best FRM is generated which results in a better performance of the FLC. The same method used for offline test was applied in real time test at the ERS to find a better FRM.

1.5 Proposed Adaptive Fuzzy Logic Controller - ES

Pham et al. (1994) proposed an evolutionary strategy (ES) design of an adaptive fuzzy logic controller for a process with time delays. An adaptive fuzzy logic controller incorporating a Smith predictor was designed to maintain performance during process changes. Simulation results demonstrated the effectiveness of this approach.

Figure 1.5-1 shows a schematic of a proposed adaptive fuzzy logic controller using Evolutionary Strategies (ES). In this method, ES is used to evolve better fuzzy membership functions. The required program of FLC, GRNN model and ES was done using Matlab software.

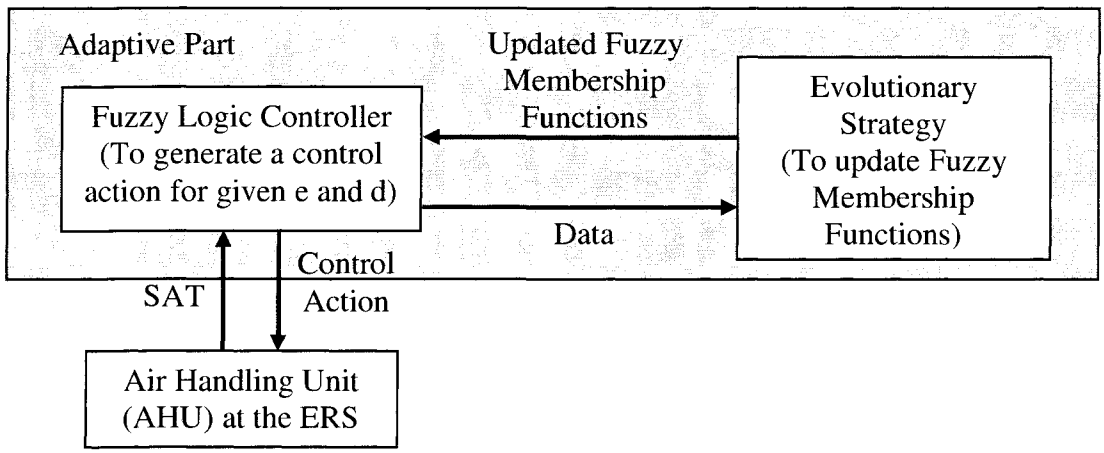

Figure 1.5-1: Proposed Adaptive Fuzzy Logic Controller - ES

The ES technique is similar to the GA technique as explained in the previous section except that GAs work with many genes at a time and ESs work with only one gene at a time. The task for this research is to develop and implement an adaptive fuzzy control algorithm at the ERS.

1.6 Plan of Work

The plan for this research has been to develop and implement an adaptive fuzzy control algorithm at the ERS. The ERS is an excellent facility to use for testing and developing FLCs for controlling commercial HVAC systems. The ERS has two side-by-side air handling units with identical thermal loads. Each system is controlled by a computerized energy management and control system (EMCS). The EMCSs can be programmed to implement and compare control strategies, such as the fuzzy logic controllers on one air handling unit and other control strategies, such as a standard PID controller, on the second air handling unit.

Tasks which are followed in this research are:

1. Develop and implement a FLC at the ERS
2. Compare performance of FLC with that of current control strategies
3. Develop procedures for dealing with missing and erroneous data
4. Develop and implement an AFLC at the ERS

Task 1 – Develop and implement a FLC at the ERS

For this task, a working fuzzy logic controller was developed, manually tuned, and used to control one of the air handling units. The control variables were selected and initial FLC development was done off-line. Then, real-time (online) experiments were conducted on one of the air handling units (AHU) to test the operation of the FLC.

Task 2 – Compare performance of FLC to current control strategies

The performance of a FLC was compared to the performance of the PIDL (proportional/integral/derivative loop) controller implemented by the EMCS at the ERS. The FLC developed in Task 1 was used to control one of the two identical air handling units and its operation and energy use were compared to the other one which had the PIDL controller. The use of the FLC was periodically switched from one air handling unit to the other to eliminate any bias in one unit over the other. The performance of both the systems in terms

of reduced energy usage, faster response time, reduced overshoot, and reduced oscillations was recorded and compared.

Task 3 – Develop procedures for dealing with missing and erroneous data

Most EMCSs have to deal with missing and/or erroneous data. This is caused by sensor malfunctions and failures or other problems within the computer control system. Poor sensor readings degrade the performance of any control system. There are two types of erroneous data: (1) complete failures and (2) minor drifting. The complete failures can easily be detected by readings that are out of range. Replacing missing and erroneous data with the previous correct value was studied. Also, in future work, Auto Associative Neural Networks will be studied as one of the tools to deal with minor drifting.

Task 4 – Develop and implement an adaptive FLC at the ERS

For this task, an adaptive FLC (AFLC) was developed to automatically tune itself for optimal performance using GA or ES. Once an adaptive FLC was developed, its performance was compared with the PIDL control strategy used at the ERS with each one controlling side-by-side air handling units. Both the systems are compared based on the following evaluation parameters:

1. Error (SAT deviation from the SATSPT)
2. Energy usage
3. Response time to setpoint and environmental changes
4. Overshoot of the controlled variables, settling and rise time required
5. Chilled water valve travel

1.7 Outline of Contents

Chapter 2 is a description of the experimental facility, including information related to building layout, air handling units, cooling coil and data acquisition system. A brief discussion of a bias study between the two systems is reported along with the uncertainty values for hydronic energy and actuator travel distance. A discussion of a chilled water valve position study is also presented. Chapter 3 discusses various performance indices used in this

study. Chapter 4 gives detailed information for the cooling coil model development using neural networks, general regression neural network and lump capacitance models. An introduction to fuzzy logic controllers and adaptive fuzzy logic controllers is discussed in Chapter 5. Experimental results for the manually tuned FLC and various cooling coil models are given in Chapter 6. Chapters 7 reports the results obtained for offline and real time tests for adaptive fuzzy logic performance using genetic algorithms and evolutionary strategies. Chapter 8 reports the results for real-time comparison tests between the adaptive fuzzy logic controller and a standard proportional, derivative and integral loop controller. Finally, conclusions, contributions, and recommendations of this study are given in Chapter 9

Chapter 2 Experimental Facility Description

2.1 Introduction

The Energy Resource Station (ERS) was used as the test facility to compare the AFLC to a conventional PID controller. The ERS was built to compare different energy efficiency measures and to record energy consumption. The ERS combines laboratory testing capabilities with real building characteristics and is capable of simultaneously testing two full-scale commercial HVAC systems side-by-side with identical thermal loadings. This method of performance testing emphasizes the importance of the entire HVAC system and the interactions between individual building elements.

To perform side-by-side testing, the facility is equipped with three AHUs. AHU-1 serves the common areas of the building. The remaining AHUs serve the A and B Test Systems. AHUA and AHUB are identical. AHUA and AHUB each serve four zones which are identical in construction and thermal loads. A schematic of the ERS along with the location of each AHU and the zones are shown in Figure 2.1-1. Of the four zones for each AHU, three have external exposures and one sees only internal conditions. The A and B zones are mirror images of each other. The zones have the capability for identical internal thermal loads.

The side-by-side zones and HVAC systems allow a conventional control methodology to be operated in one system, for example, the A-Test System, and a proposed control methodology to be operated in the other system, for example, the B-Test System. Because each test system has the identical loads (external and internal) and construction, the only difference in the tests should be the control methodologies. Hence, the energy consumption and other performance indices of the two test systems can be compared and the benefits of the proposed control methodology can be quantified.

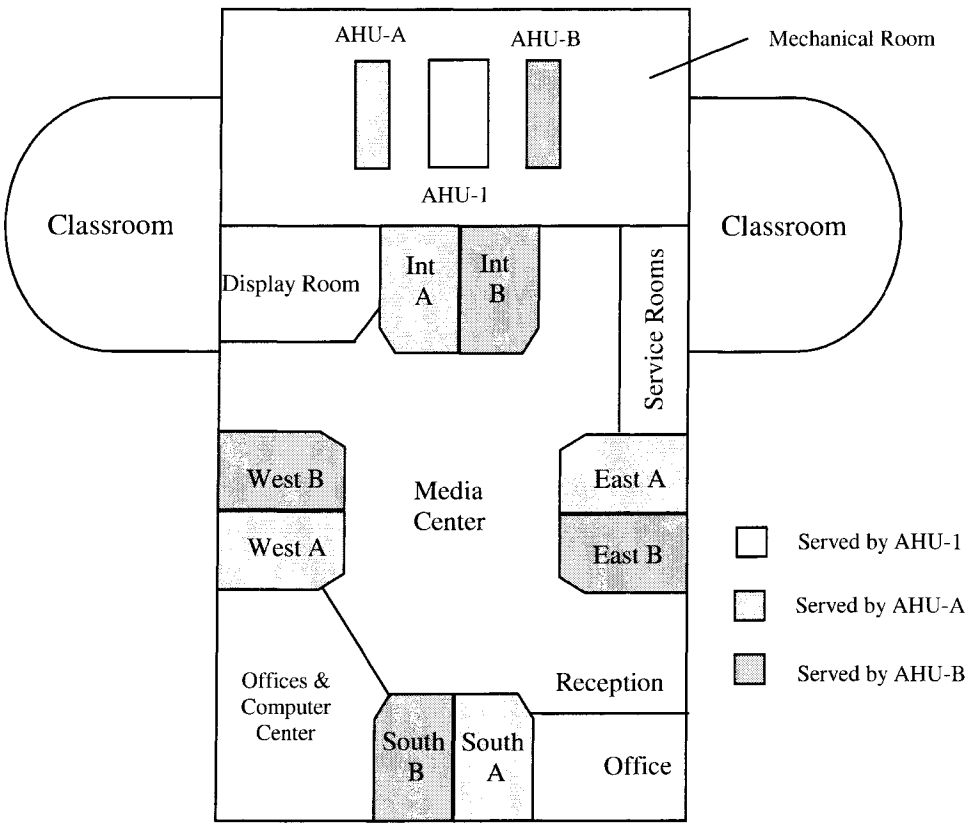

Figure 2.1-1: Schematic of Energy Resource Station, Ankeny, Iowa.

2.2 Thermal Loads

Each zone is equipped with 2-stage baseboard electric heaters and overhead lighting. The baseboard electric heaters and lighting can be scheduled to simulate various usage patterns as shown in Table 2.2-1 and Table 2.2-2. These zones are equipped with 2-stage lighting that can be in one of three modes shown in Table 2.2-1 with the corresponding thermal load generated in each stage. Also zones are equipped with 2-stage baseboard heater that can be in one of three modes shown in Table 2.2-2 with the corresponding thermal load generated in each stage. The lighting and baseboard heater thermal load was used in this study.

Table 2.2-1: Stages of lighting load

Lighting			
Mode	Stage		Power
	1	2	(W)
1	ON	OFF	195
2	OFF	ON	390
3	ON	ON	585

Table 2.2-2: Stages of baseboard heater thermal load

Baseboard Heater			
Mode	Stage		Power
	1	2	(kW)
1	OFF	OFF	0
2	ON	OFF	1
3	ON	ON	2

2.3 Air Handling Unit

A schematic of the AHUs with the major components is shown in Figure 2.3-1. Major components shown are the supply air and return air fans, preheat, cooling, and heating coils, heating and cooling control valves, recirculated air, exhaust air, and outdoor air dampers, and the ducts to transfer the air to and from the conditioned spaces. The preheat coil was not used in this study. The equipment capacities for the components of the AHUs are listed in Appendix A.

Instrumentation consisting of humidity, pressure, and temperature sensors, air flow stations, and electric power meters are available to monitor the operational characteristics of the AHUs. The accuracy of the sensors as stated by the sensor manufacturer are listed in Appendix A.

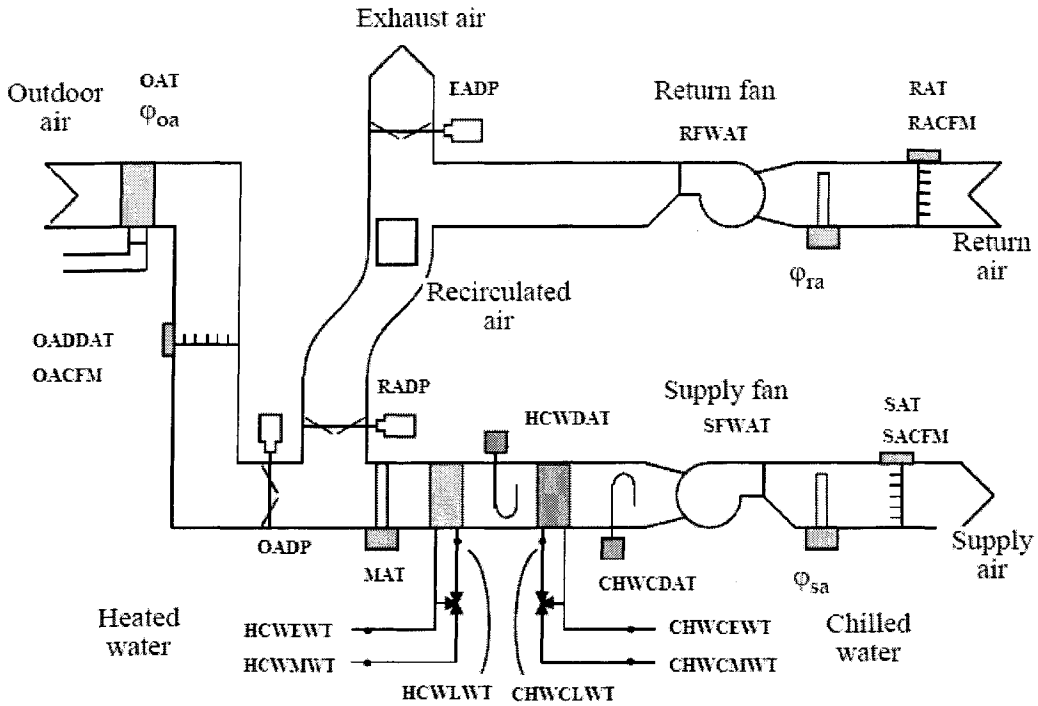

Figure 2.3-1: Schematic of Air Handling Unit (Wen 2003)

Referring to Figure 2.3-1, outdoor air at temperature OAT enters the AHU through the outdoor air duct and mixes with recirculated air at temperature RAT to form mixed air at temperature MAT. The OAT, MAT and RAT are measured with average temperature arrays of 4 platinum 1000 Ω RTD temperature probes connected to form a single average 1000 Ω RTD input. The mixed air passes through the heating coil and cooling coil at supply flow rate of SACFM. The air passes through a cooling coil where heat is removed from the air.

The cooling coil was designed for a nominal capacity of 140 kBtu/hr. The discharge air temperatures of the heating and cooling coils (HWCDAT and CHWCDAT) are measured with average temperature arrays of 4 platinum 1000 Ω RTD temperature probes connected to form a single average 1000 Ω RTD input. The flow rate of chilled water through the cooling coils is controlled by varying the position of the three-way heating and cooling valves.

The air is drawn through the supply fan. The speed of the supply fan is controlled with a variable frequency drive (VFD). The electrical power for the supply fan and VFD is

measured with an electric power transducer. The air flow rate of the supply air (SACFM) is measured at the supply air flow station and the supply air temperature (SAT) is measured with an RTD average temperature array. The humidity of the supply air (φ_{sa}) is measure using a humidity probe. The supply air is distributed to the zones through the supply air duct.

An Energy Management and Control System (EMCS) is a Johnson Controls Inc. (JCI) Metasys system (JCI, 1995) used to regulate the conditions within the AHU by providing signals to maintain SAT at the supply air temperature set-point (SATSPT).

2.4 Cooling Plant

The cooling plant consists of a nominal 10 ton air-cooled chiller, a 149 ton thermal energy storage (TES) unit, chilled water supplied from an outside source, pumps, necessary valves and piping to circulate chilled water through the HVAC components. The TES unit is not used as an energy storage device, but acts as a large capacitance that reduces the temperature oscillations of the chilled water supplied to the cooling coil.

Instrumentation consisting of flow meters, immersion temperature probes, electric power meters, and pressure sensors are available to monitor the operational characteristics of the cooling plant. Equipment capacity for each of the components of the chilled water system is listed in Appendix A. The working fluid in the cooling plant is water with 15% ethylene glycol.

Figure 2.4-1 shows a schematic of the cooling coils. The chilled water generated by the local chiller is supplied to the cooling coils in AHUA and B. The chilled water generated from the local chiller is allowed to pass through the TES unit, which is filled with water to stabilize the temperature fluctuations generated by the stage controlled local chiller. The various temperature probes are shown in Figure 2.4-1 measure the entering, leaving, and mixed water temperatures to cooling coil. A flow meter measure the flow of the chilled water through the cooling coils.

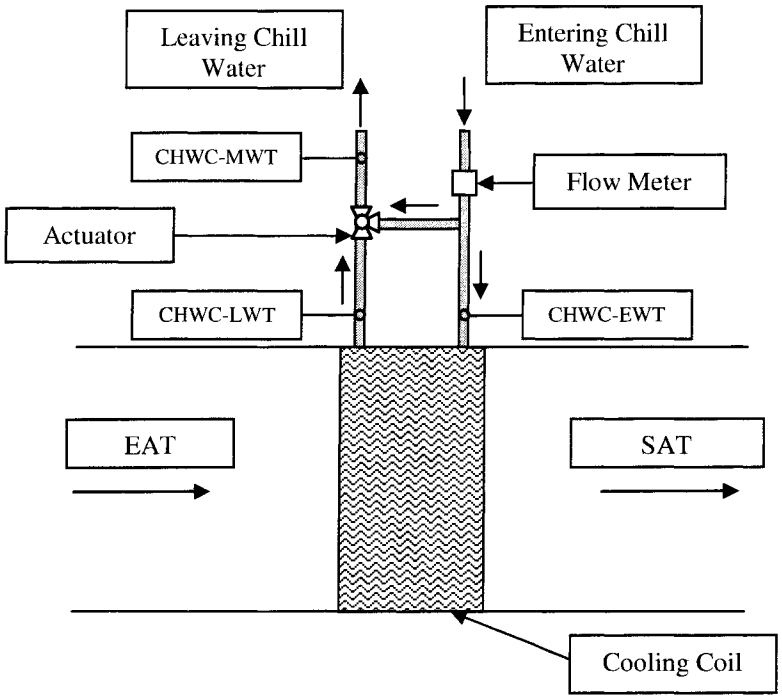

Figure 2.4-1: Schematic of Cooling Coils

CHWC-EWT - Chilled Water Entering Temperature (°F)

CHWC-LWT - Chilled Water Leaving Temperature (°F)

CWC-MWT - Chilled Water Mixed Temperature (°F)

EAT - Entering Air Temperature (°F)

SAT - Supply Air Temperature (°F)

2.5 Data Acquisition

The data exchange between the Matlab program (Matlab 2001) and the EMCS is implemented using a network dynamic data exchange (DDE) (See Appendix B for details) protocol that is supplied by the EMCS manufacturer.

The EMCS is a Johnson Controls Inc. (JCI) Metasys system (JCI 1995). For the communication between Matlab and Metasys, Metalink served as a gateway to get information from Matlab and then deliver it to the Metasys controller. Figure 2.5-1 illustrates Data Acquisition System.

Figure 2.5-1 : Data Acquisition System.

For example when a request for chilled water valve position is made by the Matlab program, this request is communicated to Metasys controller via Metalink using the network DDE protocol. The Metasys controller acquires required information and is communicated to the Matlab program through Metalink using the network DDE protocol. Similarly when a need for change in chilled water valve position is required, the Matlab program communicates this need using a "poke" command on the network DDE protocol to the Metasys controller via Metalink. Once the information for change in the chilled water valve position is received, the Metasys controller executes the required action.

The Matlab program requests values for required sensors, namely various temperatures, chilled water valve positions and damper positions, approximately after every 30 seconds, since little variation in time was observed for completion of each program cycle. One program cycle was defined as starting with the request to the Metasys system for data from the required sensors, obtain that data, calculate the error between SAT and SATSPT and then the derivative of error, calculate the FLC action to change the chilled water valve position, poke this action into the Metasys system to execute, calculate the value of the predicted SAT using different models, namely LCM, NN or GRNN models (to be defined later), and lastly storing all the values in a text file. Time variation in executing every request by Metasys system was observed due network traffic and calculation of control actions using the fuzzy logic program. All the data received from Metasys was stored using Matlab software in a text files.

2.6 Bias Study for System A and System B

Even though test systems A and B were designed to be identical, there are always factors existing in the equipment, installation, and operation that cause the two systems to have different values for the performance indices when the same operation methodology is applied to both the systems. Before conducting the comparison tests, it was necessary to determine the bias between the A and B systems from energy, thermal comfort, and actuator travel distance points of view. It is desired to know:

- The magnitudes of the differences between the performance indices (bias) for the A and B test systems when both systems are under the same operating conditions, and
- The repeatability of the bias.

Wen (2003) documented an extensive study for the bias test between the two systems at the ERS. A group of tests, called AOOM normalization tests (AOOMnorm), was developed to serve the purpose of verifying the bias between the A and B test systems. The AOOMnorm tests were configured so that the systems A and B were operated and controlled using the same conditions under PIDL control. Each AOOMnorm test lasted about one week. AOOMnorm was executed ten times to verify the bias for different weather conditions. Results for these tests are given in Table 2.6-1 and Table 2.6-2.

From the results in Table 2.6-1 and Table 2.6-2, it is observed that there is a difference in the hydronic energy consumption and the actuator travel distance for both systems when using the same PIDL control strategy. System B consumed almost 7.5% more hydronic energy than System A and the chilled water valve movement on AHUA was almost 9% more than for AHUB, calculated after excluding the outliers.

The uncertainties due to the measurement errors were also examined by Wen (2003). The uncertainties for the energy performance indices were less than 2% of the minimum performance indices value. The uncertainties for the ATD indices were less than 0.5% of the minimum performance indices value (Wen 2003).

Table 2.6-1: Performance Indices for Hydronic Energy in Cooling Tests (Wen 2003)

Performance Indices for Hydronic Energy in Cooling Test (kBtu)			
AOOMnorm2.2			
Test Day	AHUA	AHUB	% Difference (B-A)*100/B
1	135.55	147.14	8.6
2	154.01	168.34	9.3
3	120.3	132.92	10.5
AOOMnorm2.10			
1	70.79	73.64	4
2	57.25	57.19	0.1
3	84.32	89.58	6.2
4	79.31	84.8	6.9

Table 2.6-2: Performance Indices for Actuator Travel Distance in Cooling Tests (Wen 2003)

Performance Indices for Actuator Travel Distance in Cooling Test (kBtu)			
AOOMnorm2.2			
Test Day	AHUA	AHUB	% Difference (B-A)*100/B
1	311	277	-12
2	374	326	-15
3	421	385	-9
AOOMnorm2.10			
1	241	261	-9
2	279	276	-1
3	243	265	-9
4	252	298	-18

2.7 Chilled Water Valve Position Study

2.7.1 Introduction

During the initial experimental test for the cooling coil at the ERS, there was evidence that the chilled water valve position which was being recorded may not have been the real valve position. Further this could result in significant errors for predicting SAT using LCM, NN or GRNN model. So the study of the chilled water valve position was conducted to obtain the chilled water valve characteristic curve.

2.7.2 Commanded and Real Chilled Water Valve Position

For this study, the valve position was changed from 0 to 100% open position in steps of 10% after every 50 minutes and then back from 100 to 0% open position. This cycle was repeated several times. The real valve position was recorded from the available feedback signal for every commanded valve position. Figure 2.7-1 shows the commanded and the real valve position for 0 – 100% open position and Table 2.7-1 shows variation in the commanded and the real valve position for 0 – 100% and 100 – 0% open position.

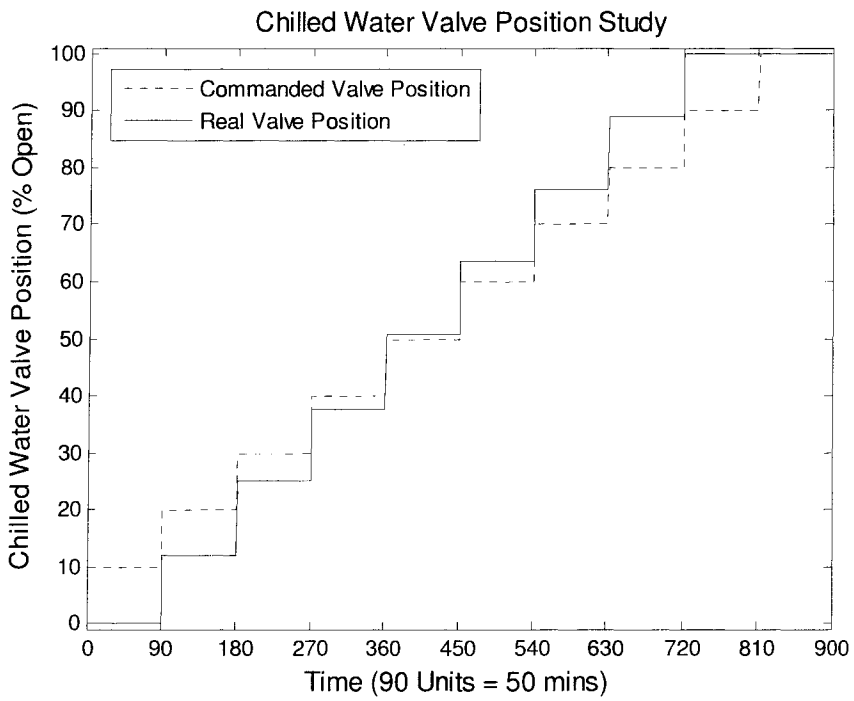

Figure 2.7-1: Commanded and real chilled water valve position

For a commanded valve position less than 50% open, the real valve position was lower and for a commanded valve position more than 60% open, the real valve position was higher.

Table 2.7-1: Commanded and real chilled water valve position

Chilled Water Valve Position (% Open)				
Commanded	Forward	Reverse	Average	Difference |For – Rev|
10	0	0	0	0
20	11.8	12.4	12.1	0.6
30	24.7	25.3	25	0.6
40	37.4	38	37.7	0.6
50	50.4	50.8	50.6	0.4
60	63.2	63.8	63.5	0.6
70	75.9	76.3	76.1	0.4
80	88.8	89.3	89.05	0.5
90	100	100	100	0
100	100	100	100	0

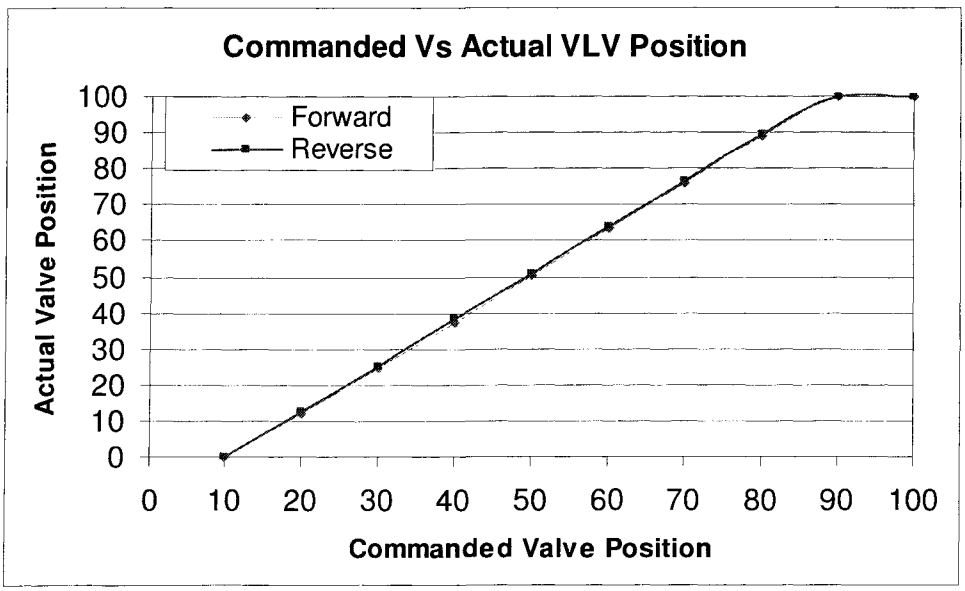

Figure 2.7-2: Commanded Vs actual chilled water valve position

From the above study, it was observed that there was little hysteresis between the forward and reverse cycles. Also significant differences between commanded and real valve

positions were observed for lower and higher valve positions as shown in Figure 2.7-2. Due to these differences, the valve position predicted by FLC did not result in the right action. This also resulted in poor performance of the NN models.

2.7.3 Chilled Water Flow through the Cooling Coil

Using results obtained from the valve position study, little improvement in the FLC and NN model performance was observed, so further investigation was necessary. The chilled water flow through the cooling coil and chilled water valve position was studied.

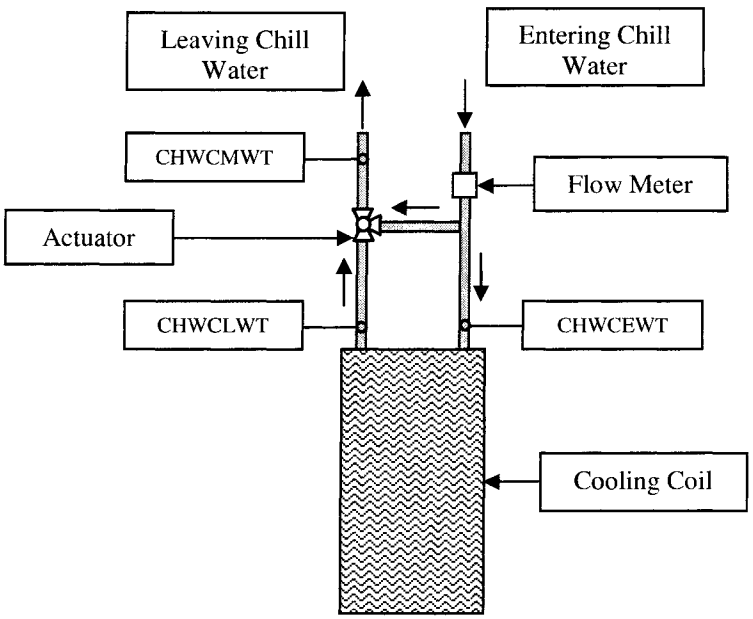

Figure 2.7-3: Details for chilled water flow through cooling coil and bypass.

Figure 2.7-3 shows the flow details through the cooling coil and bypass. Chilled water enters the system at CHWC-EWT (°F) and total flow of $\dot{Q}_{flow}$ (gpm). Depending upon the chilled water valve position, the total flow is split as $\dot{Q}_{coil}$ (gpm) flowing through the cooling coil and $\dot{Q}_{bypass}$ flowing through the bypass section.

$$\dot{Q}_{flow} = \dot{Q}_{coil} + \dot{Q}_{bypass}\ ; \qquad \text{... 2.7-1}$$

Applying the basic energy balance equation across the 3-way valve,

$$\dot{Q}_{flow} * c_p * \text{CHWC-MWT} = \dot{Q}_{coil} * c_p * \text{CHWC-LWT} + \dot{Q}_{bypass} * c_p * \text{CHWC-EWT}; \quad \ldots 2.7\text{-}2$$

where;

CHWC-LWT = Chilled water leaving temperature from cooling coil (°F)

CHWC-MWT = Chilled water mixed temperature (°F)

c_p = specific heat (Btu/lb_m-°F)

Above equations 2.7-1 and 2.7-2 can be solved to get:

$$\dot{Q}_{flow} = \dot{Q}_{bypass} * \left[\frac{(\text{CHWC-EWT} - \text{CHWC-LWT})}{(\text{CHWC-MWT} - \text{CHWC-LWT})} \right]; \quad \ldots 2.7\text{-}3$$

Simplifying above equations 2.7-3 to get following ratio

$$\frac{\dot{Q}_{coil}}{\dot{Q}_{bypass}} = \left[\frac{(\text{CHWC-EWT} - \text{CHWC-MWT})}{(\text{CHWC-EWT} - \text{CHWC-LWT})} \right]; \quad \ldots 2.7\text{-}4$$

Thus using the equation 2.7-4 and knowing the temperatures, the ratio of the flow through the cooling coil to bypass was calculated for chilled water valve positions between 0 – 100% open in steps of 10%.

Theoretically the ratio given in equation 2.7-4 should be equal to the ratio of percentage of chilled water valve position open to (100 - chilled water valve position open). Experiments with different chilled water valve positions were conducted. Figure 2.7-4 shows the ratio of flow through the cooling coil to the bypass. Experiments showed that the theoretical and experimental ratios were different. Theoretically, flow ratio given by equation 2.7-4 should not be greater than 1. But Figure 2.7-5 indicates flow ratio of greater than 1 and is due to uncertainly in reading temperatures. During the experiments, it was observed that, as the commanded valve position increases, the ratio of the flow through cooling coil to bypass got higher as expected but was not equal to the calculated theoretical ratio, shown in Table 2.7-2. These variations in the theoretical and experimental flow ratios for different valve positions lead for further investigation into the problem.

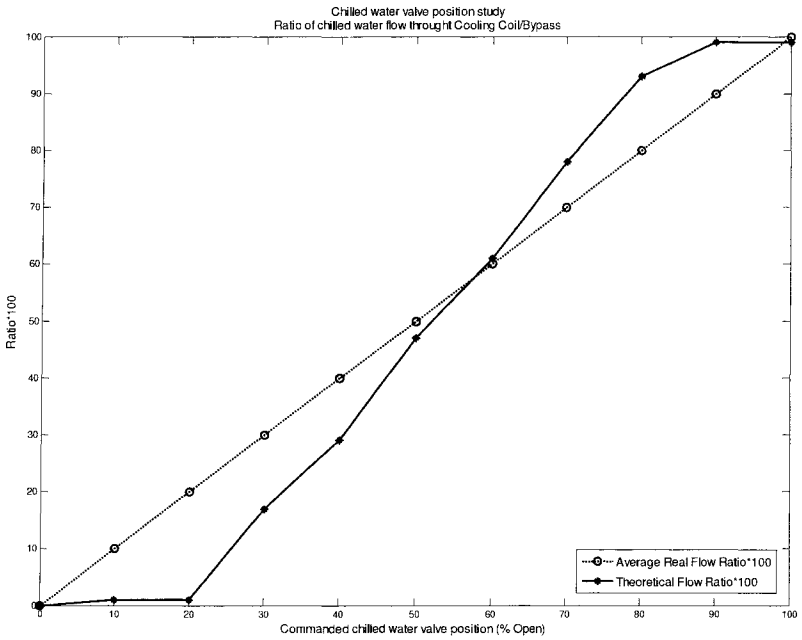

Figure 2.7-4: Flow ratio of chilled water through the cooling coils to bypass.

Table 2.7-2: Flow ratio for different chilled water valve position

Commanded Valve Position (% Open)	Actual Valve Position (% Open)	Flow ratio of chilled water through the cooling coils to bypass
10	0	0.4365
20	12.1	1.545
30	25	18.32
40	37.7	31.65
50	50.6	49.42
60	63.5	62.3
70	76.1	78
80	89.05	85.3
90	100	96
100	100	96

Using the experimental results obtained in the previous section and shown in the Table 2.7-2, a relation between the commanded valve position and the real chilled water valve position is obtained as shown in Figure 2.7-5.

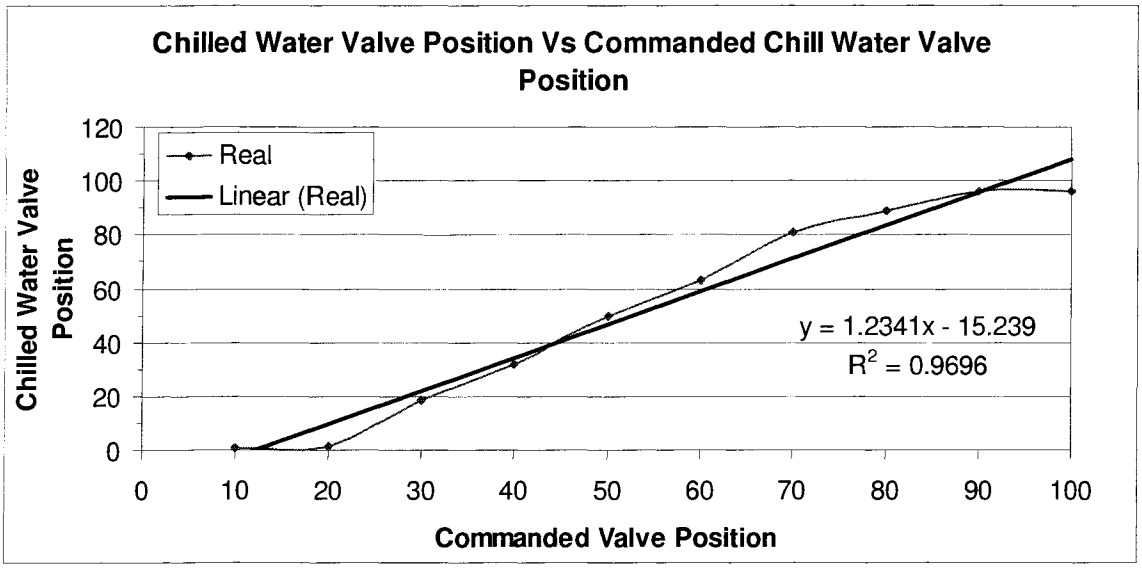

Figure 2.7-5: Variation in the ratio for chilled water flow through cooling coil to bypass

Using results shown in Figure 2.7-5, the following correlation was developed:

$$\begin{bmatrix}\text{Chilled Water}\\ \text{Valve Position}\end{bmatrix} = (1.2341 * \begin{bmatrix}\text{Commanded chilled}\\ \text{water valve position}\end{bmatrix} - 15.239); \quad \text{... 2.7-5}$$

Using the above correlation, the updated chilled water valve position is calculated. Following equation 2.7-5,

if (chilled water valve position) > 100;

then (chilled water valve position) = 100; ... 2.7-6

if (chilled water valve position) < 0;

then (chilled water valve position) = 0; ... 2.7-7

Using the above equations 2.7-5, 2.7-6, and 2.7-7 in the flow calculations, better results for the NN models and LCMs were obtained.

2.8 Summary

The Energy Resource Station (ERS) was used as the test facility to compare the AFLC to the PID controller. The ERS combines laboratory testing capabilities with real building

characteristics and is capable of simultaneously testing two full-scale commercial HVAC systems side-by-side with identical thermal loadings.

The data exchange between the Matlab program and EMCS was implemented using a network dynamic data exchange. Along with the system details, air handling unit, cooling coil, data acquisition, and a bias study between the two test systems were reviewed.

The bias study between the System A and B from energy, thermal comfort, and actuator travel distance points of view was done by Wen (2003). The bias study showed that system B consumed almost 7.5% more hydronic energy than system A and the chilled water valve movement on AHUA was almost 9% more than for AHUB. The uncertainties for the energy performance indices were less than 2% of the minimum value. The uncertainties for the ATD indices were less than 0.5% of the minimum value

In this chapter, details for the chilled water valve position study and chilled water flow through the coil study were explained. Initially, the variation in the commanded and real chilled water valve position was studied. Also, details for the study to find the variations in theoretical and real chilled water flow rate through the cooling coil for different chilled water valve positions was explained. An equation was obtained to find the corrected flow rate through the cooling coil for a given chilled water valve position and chilled water flow rate.

Chapter 3 Performance Indices

3.1 Introduction

In order to quantify and compare the performances of the Fuzzy Logic Controller (FLC) and the standard PID Loop (PIDL) controller, performance indices were developed. The performance indices used in this study are:

- Root Mean Square (RMS) Error
- Controller performance parameters
- Hydronic Energy consumed
- Actuator travel distance

In any comparison, the uncertainties for each performance index that are attributed to the errors in the measurements are required. The procedures used to calculate the uncertainties for the performance indices are introduced in Wen (2003). The two test systems, AHUA and AHUB, were designed and set up to be nearly identical. Experiments to compare the FLC and PIDL performances were performed with the FLC on one system and the PIDL on the other. The systems were switched for alternate test runs to remove any bias of one system over the other. The performance indices are discussed in this section and then used to present the results in later sections.

3.1.1 Root Mean Square Error

Performance of FLC was evaluated using the standard statistical measures of root mean square error (RMS error), which is defined as

$$\text{RMS error} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\text{SAT-SATSPT})^2} \qquad \text{.. 3.1-1}$$

where,

SAT = Supply Air Temperature (°F)

SATSPT = Supply Air Temperature Set-Point (°F)

n = number of data patterns

3.1.2 Controller Performance Parameters

The controller performance was assessed using measurable variables. For control problems, where the aim is to keep a process-state variable at its specific set-point, the following performance related variables, as shown in Figure 3.1-1 are used. These variables are defined for a step change in conditions (the set point).

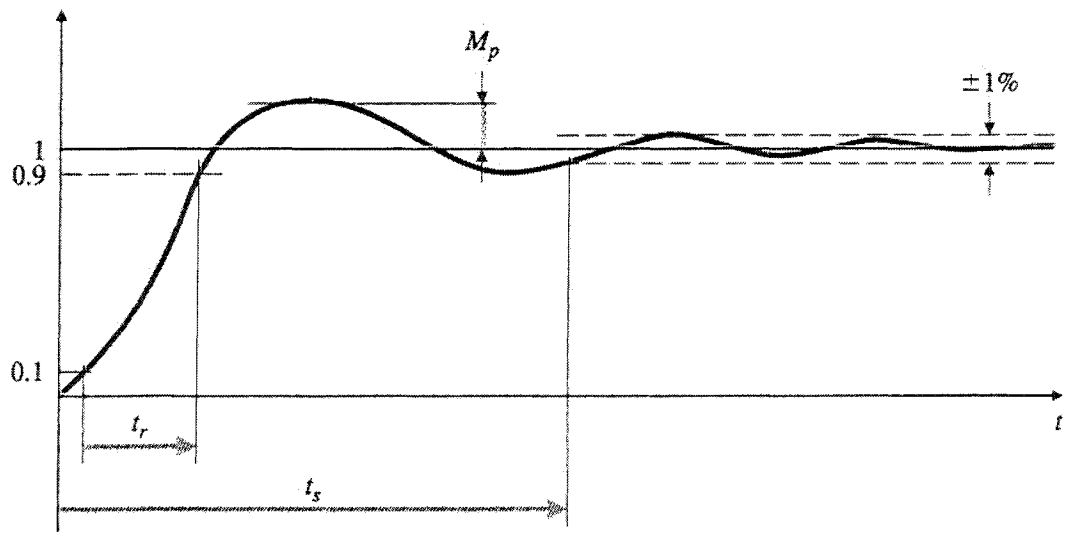

Figure 3.1-1: Definition of Rise Time, Settling Time and Overshoot

Rise Time:

The rise time t_r is the time it takes for the system to go from 10% to 90% of the step change.

Settling Time:

The settling time t_s is the time it takes the system transients to decay to less than 1% of the step change.

Magnitude of Overshoot:

The overshoot M_p is the maximum amount the system overshoots its final value.

3.1.3 Hydronic Energy

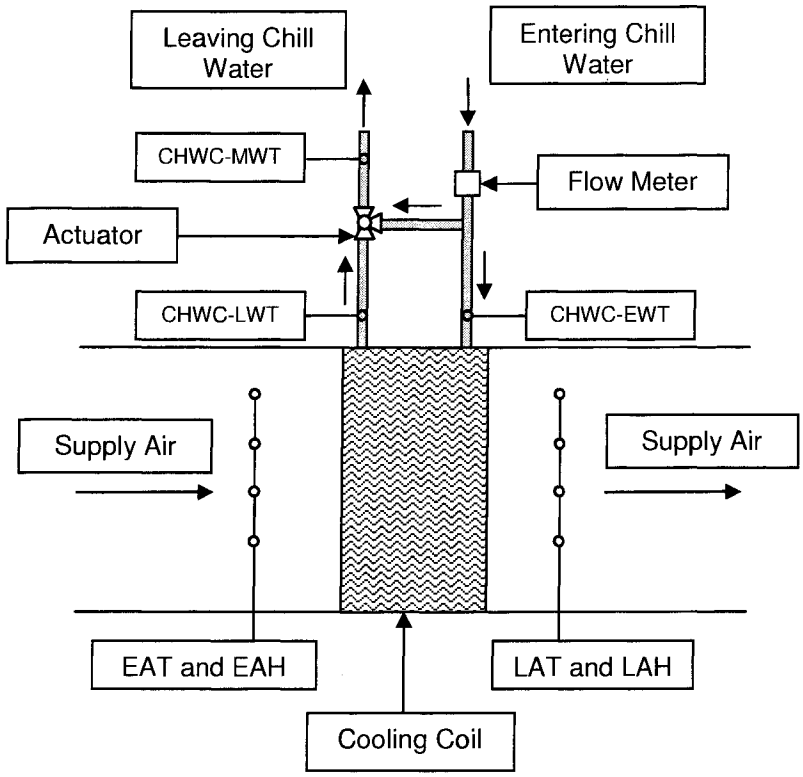

Figure 3.1-2: Details for AHU Cooling Coil Instrumentation

This section deals with the energy balance across the cooling coil and provides details to calculate energy gain by air and energy loss by water. Since cooling coil are exposed to moist air, psychrometric properties are considered to analyze moist air conditions for the calculation of energy balance. Kuehn et al. (1998) showed that perfect gas relations can be used to calculate humidity ratio, enthalpy and specific volume of saturated air of moist air at standard atmospheric pressure for a temperature range of –60°F to 120°F with errors less than 0.7%.

The water vapor saturation pressure is required to determine different moist air properties. Hyland and Wexler (1983) developed a formula for finding water vapor saturation pressure over liquid water for temperature of 32°F to 392°F and is given by:

$$p_{wsin} = \exp(c_8/EAT_{conv} + c_9 + c_{10} * EAT_{conv} + c_{11} * (EAT_{conv})^2 + c_{12} * (EAT_{conv})^3 + c_{13} * \log(EAT_{conv})) \quad \ldots 3.1\text{-}2$$

where,

$p_{\omega sin}$ = water vapor saturation pressure, psia;

EAT = temperature of air entering cooling coil, °F

EAH = relative humidity of air entering cooling coil,

EAT_{conv} = EAT + 459.67;

c_8 = -1.0440397E+04;

c_9 = -1.1294650E+01;

c_{10} = -2.7022355E-02;

c_{11} = 1.2890360E-05;

c_{12} = -2.4780681E-09;

c_{13} = 6.5459673E+00;

Moist air is considered as a mixture of dry air and water vapor obeying perfect gas relationship, which is given by:

$$pV = nRT_{abs} \quad \text{... 3.1-3}$$

where;

p = total mixture pressure

p = $p_{da} + p_w$,

p_{da} = partial pressure of dry air

p_ω = partial pressure of water vapor

V = total mixture volume

n = $n_{da} + n_w$

n_{da} = number of moles of dry air

n_w = number of moles of water vapor

R = universal gas constant, 1545.32 ft*lb_f/lb mol*°R

T_{abs} = absolute temperature, °R

Also, $p_{\omega s}$ is a function of temperature only and differs slightly from the vapor pressure of water in saturated moist air. The relative humidity (φ) is given by (ASHRAE F6, ASHRAE 1987)

Relative Humidity (φ)=$p_{\omega in}/p_{\omega sin}$

Thus,

$$p_{\omega in} = \varphi * p_{\omega sin} \quad \text{...3.1-4}$$

The total mixture pressure of moist air entering in the cooling coil is:

$$p_{in} = 14.67 + p_{\omega in} \quad \text{... 3.1-5}$$

where,

pressure of dry air is taken as 14.67 psia,

Thus Humidity Ratio (lb_w/lb_{da}) of entering moist air into the cooling coil is calculated as per ASHRAE F6 as:

$$\omega_{in} = 0.62198 * p_{\omega in}/(p_{in} - p_{\omega in}) \quad \text{... 3.1-6}$$

Now, using equation 32 from ASHRAE F6, enthalpy (BTU/lb_{dryair}) of entering moist air into the cooling coil is calculated as:

$$h_{airin} = 0.24 * EAT + \omega_{in} * (1061 + 0.444 * EAT) \quad \text{... 3.1-7}$$

Similarly, the enthalpy (BTU/lb_{dryair}) of leaving air from the cooling coil is calculated using following equations:

$$p_{\omega sout} = \exp(c_8/LAT_{conv} + c_9 + c_{10} * LAT_{conv} + c_{11} * (LAT_{conv})^2 + c_{12} * (LAT_{conv})^3 + c_{13} * \log(LAT_{conv})) \quad \text{... 3.1-8}$$

where,

$p_{\omega sout}$ =water vapor saturation pressure of air coming out from cooling coil, psia;

LAT =temperature of air leaving cooling coil; °F

LAH =relative humidity of air leaving cooling coil,

LATconv = LAT + 459.67;

Total mixture pressure of air leaving from the cooling coil is:

$$p_{out} = 14.67 + p_{\omega sout} \quad \text{... 3.1-9}$$

where,

pressure of dry air is taken as 14.67 psia,

$$\omega_{out} = 0.62198 * p_{\omega out}/(pout - p_{\omega out}) \quad \text{... 3.1-10}$$

$$h_{airout} = 0.24 * LAT + \omega_{out} * (1061 + 0.444 * LAT) \quad \text{... 3.1-11}$$

With known h_{airin} and h_{airout}, rate of heat transferred into air q_{air} (BTU/hr) is calculated as:

$$q_{air} = Q * (h_{airin} - h_{airout}) * 60/v_{airout} \quad \text{... 3.1-12}$$

where,

Q = volumetric flow rate of air, ft^3/min and

v_{airout} = specific volume of air leaving from cooling coil, ft^3/lb_{da} and is calculated using ASHRAE F6 equation 28.

$$v_{airout} = \left[\frac{0.7543 * (LAT + 459.67) * (1 + 1.6078 * \omega_{out})}{(p_{out}/0.4898)} \right] \quad \text{... 3.1-13}$$

From recorded mixed water temperature (CHWC-MWT, °F), entering water temperature (CHWC-EWT, °F), flow rate of chilled water through the cooling coil $\dot{m}_{water}$, (gpm) and with known 15% glycol in chilled water, heat transferred from water q_{water} (Btu/hr) is calculated as:

$$q_{water} = \left[\frac{63.98 * \dot{m}_{water} * (CHWC\text{-}MWT - CHWC\text{-}EWT) * 0.936 * 60}{7.48052} \right] \quad \text{... 3.1-14}$$

The uncertainties for the energy performance indices were less than 2% of the minimum performance index value (Wen 2003).

3.1.4 Actuator Travel Distance

Actuators (valve) movement plays an important role in building control and operation. Frequent variations of set points or faulty control algorithms yield more movement of actuators that eventually wear them out and result in actuator failures. Hence, the actuator travel distance (ATD) index is a performance index to evaluate the valve travel distance. The larger is the value of the ATD index, the more the actuator travels. Actuators that have large values of the ATD index are considered to have a high probability of failure.

The ATD index is defined as:

$$ATD=\frac{\sum_{i=k_i+1}^{k_e} |VLV(i)-VLV(i-1)|}{k_e-k_i} \quad \text{... 3.1-15}$$

where,

VLV(i) = the control signal to the actuator at the current time step and

VLV(i-1) = the control signal to the actuator at the previous time step

k_i = staring time step

k_e = ending time step

Since actuators have a feedback signal, real actuator position was used for calculation of ATD. The uncertainties for the ATD indices were less than 0.5% of the minimum performance index value (Wen 2003).

3.2 Summary

In this chapter, various performance indices for comparison of two systems were defined. Various performance indices defined are the RMS error between the supply air temperature and supply air temperature setpoint, the hydronic energy used by the system, and the actuator travel distance. For hydronic energy calculations, an ASHRAE method was used. Also controller performance parameters were assessed using measurable variables. For control problems, where the aim is to keep a process-state variable at its specific set point, the rise time, settling time and overshoot values were used to compare the performance of each system.

Chapter 4 Cooling Coil Models

4.1 Introduction

The objective for developing a cooling coil model is to capture the dynamic response of the HVAC process and assist in designing a FLC. If the FLC could be used to control a model of the system, then the FLC might be trained quicker. Four different models were developed for this study: Neural Network, General Regression Neural Network, Adaptive Neural Network, and Lumped Capacitance models.

4.2 Model using Neural Networks

4.2.1 Introduction

A Neural Network (NN) is a network of neurons having a local memory. Unidirectional connections exist between the neurons. Each connection carries a weight and a bias. They operate on the input/s they receive and produce an output/s depending on the weights and biases. These weights and biases are adjusted to minimize the error between the actual and predicted output.

As shown in Figure 4.2-1, a NN is a black box which learns mapping of the inputs to the outputs. It does this by adjusting the neurons weights and biases using data available for training. Neural Networks have been proven to be successful for many complex problems for which hard and fast rules can not easily be applied.

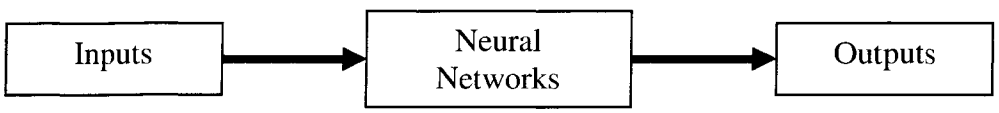

Figure 4.2-1: Neural Networks

Matlab's NN toolbox was used to develop the cooling coil models. To create a NN model of the cooling coil, the following information must be specified about its architecture:

1. Number of inputs
2. Number of outputs

3. Number of hidden layers
4. Number of neurons in each hidden layer
5. Transfer function in each hidden layer
6. Transfer function in the output layer
7. Algorithm for adjusting weights and biases
8. Performance function

Once the basic NN model's architecture is prepared; training was done with available data. The NN used in this study had one hidden layer. The *tansig* transfer function was chosen as the activation function for the hidden layer, while the input and output layers used *linear* transfer function for activation (See Appendix D for details).

4.2.2 Normalization

The NN training process was made more efficient by normalizing the input data in a preprocessing step of the NN model development. The output was also normalized and used a postprocessing step to obtain the output data.

Once the most appropriate raw input data has been selected, it was preprocessed to obtain better NN performance. Transformation and normalization are two widely used preprocessing methods. Transformation involves manipulating raw input data to create a single input to a network, while normalization is a transformation performed on a single input data to distribute it evenly and scale it into an acceptable range for the network.

Data should be scaled within an input range for each neuron, which is typically between -1 to 1 or zero to 1. Knowledge of the input domain is important in choosing preprocessing methods to highlight underlying features in the data, which can increase the network's ability to learn the association between inputs and outputs.

In the present case, the actual data was normalized in the range from -1 to 1. So minimum normalized input data was -1 and the maximum was 1.

$$\text{NormalizedValue}=2 * \left[\frac{(\text{Value} - \text{MinValue})}{(\text{MaxValue} - \text{MinValue})}\right] - 1; \quad \text{... 4.2-1}$$

For example, if minimum and maximum values for CHWC-MAT are set as 60 and 80 degrees respectively, then the normalized value for CHWC-MAT of 72 degree is calculated as:

$$\text{NormalizedValue}=2 * \left[\frac{72 - 60}{80 - 60}\right] - 1 = 0.2; \quad \text{... 4.2-2}$$

Similarly if CHWC-MAT is 64 degrees, the normalized value is -0.6. All input values are normalized in a similar way.

The output of a NN is obtained as a normalized value, to get real output, the output must be denormalized. For example, if the minimum and maximum value of a target is set as 65 and 75 respectively and if the network normalized value output is 0, then corresponding denormalized value is calculated as:

$$\left[\begin{matrix}\text{Denormalized} \\ \text{Value}\end{matrix}\right] = \left[(\text{NNOutput} + 1) * \frac{(\text{MaxTargetValue} - \text{MinTargetValue})}{2}\right] + \text{MinTargetValue}; \quad \text{... 4.2-3}$$

$$\text{DenormalizedValue} = (0 + 1) * \frac{(75-65)}{2} + 65 = 70; \quad \text{... 4.2-4}$$

Some rules of thumb when performing data normalization are:

- Normalize all inputs and outputs.
- Don't restrict to the same type of normalization for all inputs/outputs.
- Use the same normalization method and maximum and minimum values for training and test data.
- Make sure that normalization of output data is sufficiently reversible.

Once the network architecture has been selected and the inputs chosen and preprocessed, the NN is ready to be trained.

4.2.3 Training of Neural Networks Model

During the network training phase, the backpropagation algorithm was used to adjust weights and biases to minimize network error. A feedforward network architecture using a backpropagation algorithm was used for training as it provides fast and accurate model development (Curtiss 1993). A technique which involves performing computations backward through the network is called Backpropagation (See Appendix D for details). The default performance function for a feedforward NN is the root mean square error between the network outputs and the specified target values.

Development of a NN model started with a study of the different parameters which affect the desired supply air temperature from the cooling coil. Initially following input parameters were used as inputs for NN model:

1. Chilled Water Entering Temperature (CHWC-EWT);
2. Chilled Water Leaving Temperature (CHWC-LWT);
3. Chilled Water Mix Temperature (CHWC-MWT);
4. Mixed Air Temperature (CHWC-MAT);
5. Supply Air Flow Rate (SA-CFM);
6. Chilled Water Coil Valve Position (CHWC-VLV);
7. Chilled Water Flow Rate (CHWP-GPM)
8. Supply Air Set Point (SAT-SPT);

Using the above inputs, several different NN architectures obtained by varying the number of neurons in hidden layers and trying different training algorithms were trained with the available data to get predicted Supply Air Temperature (SAT_{PRED}) as an output. Performance of each NN was evaluated using the standard statistical measures of root mean square error (RMS error), which is defined as

$$RMSError = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(SAT_{PRED} - SAT)^2} \quad \text{... 4.2-5}$$

where,

SAT_{PRED} = Predicted Supply Air Temperature by NN (°F)

SAT = Supply Air Temperature (°F)

n = number of data patterns in test set

Another study was conducted using eight inputs, namely:

- Chilled Water Entering Temperature (CHWC-EWT);
- Mixed Air Temperature (CHWC-MAT);
- Supply Air Flow Rate (SA-CFM);
- Chilled Water Coil Valve Position (CHWC-VLV);

and their derivative information for single and two hidden layer NN models. Adding derivative inputs should help NN in mapping the time dependence of input parameters on the output.

To simplify the network architecture and reduce training time and memory requirements a NN model was studied with four inputs, namely:

1. Chilled Water Entering Temperature (CHWC-EWT);
2. Mixed Air Temperature (CHWC-MAT);
3. Supply Air Flow Rate (SA-CFM);
4. Chilled Water Coil Valve Position (CHWC-VLV);

These inputs are chosen based on their availability in a typical commercial building.

4.2.4 Summary

This section provided brief information of different NN models studied. The need for normalization and data preprocessing was explained.

4.3 Model using General Regression Neural Networks

4.3.1 Introduction

Another type of feedforward NN, namely General Regression Neural Networks (GRNN) was studied. The GRNN is a one pass learning algorithm that can be used for estimation of continuous variables and converges to the underlying regression surface. A GRNN captures the input-output regression characteristics of the system. The GRNN for this study is based on the Parzen window estimator (Parzen 1962) and was the first GRNN application done by Specht (1991). The principal advantages of the GRNN are its quick learning and fast convergence to an optimal regression surface as the number of samples becomes large.

4.3.2 **GRNN Theory** (Specht 1991)

The GRNN has been proposed as an alternative to statistical regression equations and conventional artificial neural-networks. GRNNs are memory-based feedforward networks based on the estimation of probability density functions. GRNNs feature fast training times, can model non-linear functions, and have been shown to perform well in noisy environments.

The GRNN is theoretically based on the estimation of a probability density function from observed samples using Parzen window estimation (Specht 1991). The following is a brief description of the GRNN that illustrates its implementation for estimation of AHU subsystem process-variables.

Assume that X and Y are the measured values for the variables x and y, respectively. If f(x, y) represents the known joint continuous probability density function, the expected value of y given X can be obtained from the following equation:

$$E[y|X] = \frac{\int_{-\infty}^{\infty} yf(X,y)dy}{\int_{-\infty}^{\infty} f(X,y)dy} \qquad \text{... 4.3-1}$$

When the density function f(x,y) is not known, it can be estimated from sample data sets of x and y. Assuming that the density is continuous and the first partial derivatives of the function evaluated at any x are small, the probability estimator is

$$\hat{f}(X,Y)=\frac{1}{2\sigma^{(p+1)}\Pi^{(p+1)/2}}\frac{1}{n}\sum_{i=1}^{n}\exp\left[\frac{(X-X_i)^T(X-X_i)}{2\sigma^2}\right]\exp\left[\frac{(Y-Y_i)^2}{2\sigma^2}\right] \quad \text{... 4.3-2}$$

where Xi and Yi are the i^{th} sample data point, n is the number of samples and p is the dimension of the vector variable x.

A physical interpretation of the probability estimate $\hat{f}(X,Y)$ is that it assigns a sample probability of width σ for each sample of Xi and Yi, and the probability estimate is the sum of those sample probabilities.

Substituting equation 4.3-2 into equation 4.3-1, the expected value $\hat{Y}(X)$ given X can be calculated as

$$\hat{Y}(X)=\frac{\sum_{i=1}^{n}Y_i\exp\left(\frac{-D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{n}\exp\left(\frac{-D_i^2}{2\sigma^2}\right)} \quad \text{... 4.3-3}$$

where Di is given by the distance function of the input space and calculated as:

$$D_i^2=(X-X_i)^T(X-X_i) \quad \text{... 4.3-4}$$

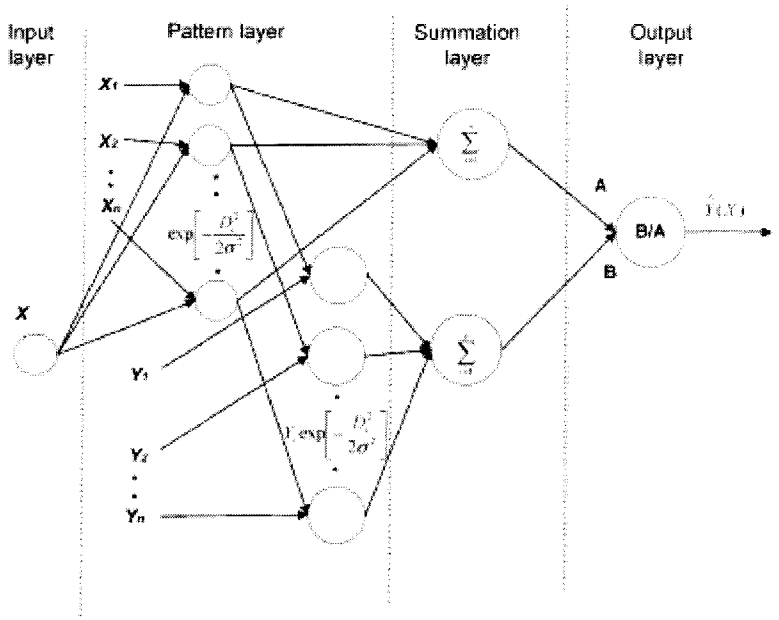

Figure 4.3-1: GRNN Model Architecture (Specht 1991)

The GRNN topology consists of four layers as shown in Figure 4.3-1. The primary advantage of the GRNN is the speed at which the network can be trained. Training a GRNN is performed in one pass. The training data (also referred to as reference data) Xi are simply copied into the pattern layer of the neural net. In the second layer, for a given X, the exponent based on the distance from X to each training data input Xi is computed and the product of the exponent value and the corresponding training data output Yi is also calculated. The summation layer has two nodes, termed A and B. The B node computes the summation of each exponential value weighted by the known output Yi, while the A node simply computes the summation of the exponents of the distances. The output layer simply divides B by A to produce the predicted output. The output layer is simply a weighted average of the training values close to the input values. The only adjustable parameter in a GRNN is the smoothing factor, s. The smoothing factor can be found through a simple and effective scheme known as the "Holdout" method (Specht 1991).

4.3.3 GRNN Models

To predict next sampling time SAT, an offline GRNN model was developed using 8 inputs, namely;

1. Chilled Water Entering Temperature (CHWC-EWT);
2. Chilled Water Leaving Temperature (CHWC-LWT);
3. Chilled Water Mix Temperature (CHWC-MWT);
4. Chilled Water Flow Rate (CHWP-GPM)
5. Mixed Air Temperature (CHWC-MAT);
6. Supply Air Flow Rate (SACFM);
7. Chilled Water Coil Valve Position (CHWC-VLV);
8. Supply Air Set Point (SATSPT);

The performance of this GRNN model was evaluated by calculating the RMS errors between SAT and predicted SAT value for all the data patterns.

Offline testing of the GRNN model was performed using Matlab's Neural Network Toolkit. For offline testing of a GRNN model on unseen data resulted in higher RMS error. Previous work on NN (Curtiss 1996, Jeannette 1998, Osman 2000a, 2000b) showed that past information of output/s as an input provides better results. So, two other GRNN models (GRNN models II and III) using past output information for SAT were studied. GRNN model II used 18 inputs, namely the 8 inputs used in GRNN Model I and 10 time steps of past information for the SAT. GRNN model III was developed since it was though having derivative information of output for the past time steps would help GRNN mapping inputs to provide better output results. GRNN model III was almost similar to GRNN model II except GRNN model III used 10 time steps of past derivative information for the SAT. Table 4.3-1 summarizes the GRNN model types.

Table 4.3-1: GRNN Models

GRNN Model Type	Input Information
GRNN Model I	8 inputs
GRNN model II	8 inputs with 10 time steps of past information for the output
GRNN model III	8 inputs with 10 time steps of past derivative information for the output

4.3.4 Summary

In this section details of three GRNN models developed for cooling coil are discussed along with GRNN theory. The three GRNN models used different number of inputs

4.4 Model using Adaptive Neural Networks

4.4.1 Introduction

To develop an adaptive NN model, an updating NN technique was proposed. The NN model developed had a feedforward architecture and was first trained off-line using training data that was obtained from the ERS. This NN model was then adaptively retrained off-line and then again on-line using an adaptive process in order to minimize the difference between SAT and SAT_{PRED}. Two different techniques namely, re-training and adaptive, were used for updating the NN models.

In the re-training updating process, the network's weight and bias values are updated using all of the data for a run. This training was continued until the maximum number of epochs was reached, the performance goal was met, or a stopping condition of the function occurred.

In the adaptive updating process, the network's weight and bias values are updated after each step, before the next step in the sequence is presented

4.4.2 Summary

In this section, brief information related to the adaptive NN models is discussed. Retraining and adaptive techniques are used for developing an adaptive NN model

4.5 Model using Lumped Capacitance Method

4.5.1 Introduction

Limited tools are available to evaluate the performance of an installed chilled water cooling coil. To design a good control system, steady state and dynamic behavior of the chilled water cooling coil are required.

Clark (1985) studied the behavior of heating coils and applied lumped capacitance concepts to develop a dynamic model. But this model is neither completely understood for the time-dependent coefficients used in the polynomial for the delay function nor experimentally validated. Moreover, application of this model for varying flow rates requires four constants to be varied along with the flow rate. Lebrun (1990) presented a dynamic model of chilled water cooling coils using the lumped capacitance concept. Starting with the steady state model, dynamic behavior is explained with a simple first order model using the dry and wet regimes. But this model needs improvements on the water side considering the transport delay.

A wide range of heat exchanger models are available with varying complexities. But most of the models, Lebrun (1990), Stoecker (1975), need detail information about the coil construction, inlet condition for air and water, humidity ratios, air and water flow rates. The required information might not always be ready available. Moreover, these models may perform better in a limited range of operation and coil geometric parameters.

This section presents a technique for modeling a cooling coil without condensation. The results are used for designing a better control system. This technique is based on the fundamental heat and mass transfer principles and equations that are modified to lump all the geometric parameters of a cooling coil into characteristic parameters and to use Genetic Algorithms (GAs) to find these characteristic parameters which are the overall heat transfer/area coefficient (UA), two chilled water cooling coil valve constants, air flow constant, thermal time constant for air, and thermal time constant for water.

4.5.2 Lump Capacitance Model Development

The Lump Capacitance Model (LCM) for chilled water cooling coil (CC) was developed with two objectives: first, to use in the development of a control system and, second, to investigate CC performance for different operating conditions. Non-linear characteristics of CCs are shown in the studies conducted by Clark (1985) and Shavit (1982). Existing control models are very complex, non-linear and need detailed information about the CC. This creates difficulties in the real-time application of a model. The model developed for this

study is to be trained using operating data for a specific cooling coil. Enough parameters are needed to provide a good model while not having too many parameters to make the model difficult to train.

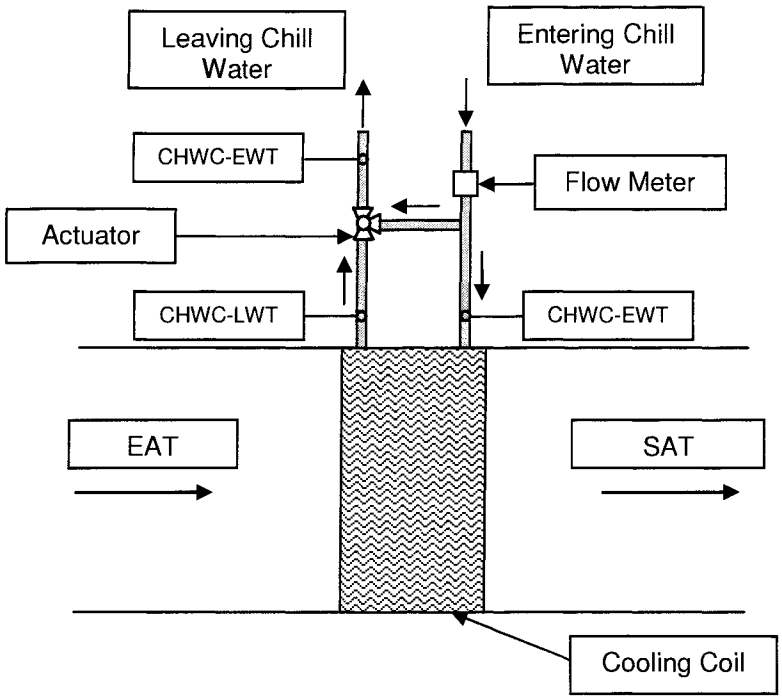

Figure 4.5-1: Schematic of Cooling Coil (CC)

CHWC-EWT = entering chilled water temperature into the cooling coils (°F)

CHWC-LWT = leaving chilled water temperature from the cooling coils (°F)

CHWC-MWT = mixed chilled water temperature (°F)

EAT = entering air temperature (°F)

SAT = supply air temperature (°F)

CC = Cooling Coil

Schematic of the CC is shown in Figure 4.5-1. The chilled water entering the CC either passes through the CC or through the bypass depending upon the three-way valve position. The three-way valve controls the volume of chilled water passing through the CC. Chilled water enters the CC at CHWC-EWT. Due to heat transfer with the entering air (EAT) in the CC, the temperature of chilled water increases and it leaves the CC at CHWC-LWT. The

leaving chilled water mixes with the bypass chilled water and is then returned to storage tank at CHWC-MWT. Air enters the CC at EAT and leaves at SAT.

This model uses an approximate equation for effectiveness (ε) as a function of number of transfer units (NTU) and heat capacity ratio (C_r) to determine steady state water and air temperatures for a cross-flow heat exchanger with both fluids unmixed (Incropera 1996).

$NTU = UA/C_{min}$; ... 4.5-1

where,

UA = constant value of overall heat transfer coefficient area product;

C_{min} = minimum capacity rate (J/K) of air (C_a) or water (C_w);

$C_a = \dot{m}_a * C_{pa}$; ... 4.5-2

$C_w = \dot{m}_w * C_{pw}$; ... 4.5-3

C_{pa} = specific heat of air; (J/KgK)

C_{pw} = specific heat of water; (J/KgK)

$\dot{m}_a$ = mass flow rate of air; (kg/s)

= CFM * AF_c; ... 4.5-4

CFM = Supply Air Flow Rate; (cfm)

AF_c = Air Flow constant;

$\dot{m}_w$ = mass flow rate of water; (kg/s)

= chilled water valve position * VLV_c; ... 4.5-5

VLV_c = chilled water valve position constant;

$\varepsilon = 1-\exp((1/C_r)*(NTU^{0.22})*(\exp(-C_r *(NTU^{0.78}))-1))$; ... 4.5-6

Also, effectiveness can be calculated as:

if $C_a < C_w$

$$\varepsilon = \frac{(EAT - SAT_{ss})}{(EAT - CHWC\text{-}EWT)} \quad \text{... 4.5-7}$$

if $C_w < C_a$

$$\varepsilon = \frac{(CHWC\text{-}LWT_{ss} - CHWC\text{-}EWT)}{(EAT - CHWC\text{-}EWT)}. \quad \text{... 4.5-8}$$

where,

SAT_{ss} = Steady state supply air temperature; (°F) and

$CHWC\text{-}LWT_{ss}$ = Steady state leaving chilled water temperature; (°F)

Depending upon whether the minimum value of heat capacity rate is for water or air, using appropriate equations 4.5-9 and 4.5-11 or 4.5-13 and 4.5-15 steady state water and air outlet temperatures are calculated (Incropera 1996). Using water and air side thermal time constant values, which are found using GAs, water and air outlet temperatures are calculated as given in equations 4.5-10 & 4.5-12 or 4.5-14 & 4.5-16 respectively.

if $C_w < C_a$

$$CHWC\text{-}LWT_{ss} = CHWC\text{-}EWT + \varepsilon * (EAT - CHWC\text{-}EWT); \quad \text{... 4.5-9}$$

$$CHWC\text{-}LWT_{PRED} = (CHWC\text{-}LWT - CHWC\text{-}LWT_{ss}) * \exp\left[-\frac{\text{Time Step}}{T_{cw}}\right] + CHWC\text{-}LWT_{ss} \quad \text{... 4.5-10}$$

$$SAT_{ss} = EAT - \frac{C_w}{C_a} * (CHWC\text{-}LWT_{ss} - CHWC\text{-}EWT); \quad \text{... 4.5-11}$$

$$SAT_{PRED} = (SAT - SAT_{ss}) * \exp\left[-\frac{\text{Time Step}}{T_{ca}}\right] + SAT_{ss}; \quad \text{... 4.5-12}$$

if $C_a < C_w$

$$SAT_{ss} = EAT - \varepsilon * (EAT - CHWC\text{-}EWT); \quad \text{... 4.5-13}$$

$$SAT_{PRED} = (SAT - SAT_{ss}) * \exp\left[-\frac{\text{Time Step}}{T_{ca}}\right] + SAT_{ss}; \quad \text{... 4.5-14}$$

$$CHWC\text{-}LWT_{ss} = CHWC\text{-}EWT - \frac{C_a}{C_w} * (SAT_{ss} - EAT); \quad \text{...4.5-15}$$

$$CHWC\text{-}LWT_{PRED} = (CHWC\text{-}LWT - CHWC\text{-}LWT_{ss}) * \exp\left[-\frac{\text{Time Step}}{T_{cw}}\right] + CHWC\text{-}LWT_{ss}; \quad \text{... 4.5-16}$$

where,

Time Step = Time duration between each reading; (seconds)

T_{cw} = Thermal time constant for water loop; (seconds) and

T_{ca} = Thermal time constant for air loop; (seconds)

Thus, if UA is assumed to be constant and finding values of AFc, VLVc, T_{cw} and T_{ca}, SAT_{PRED} value is calculated. Figure 4.5-2 shows the various steps followed to calculate SAT_{PRED} and various constants used in the equations for the calculations.

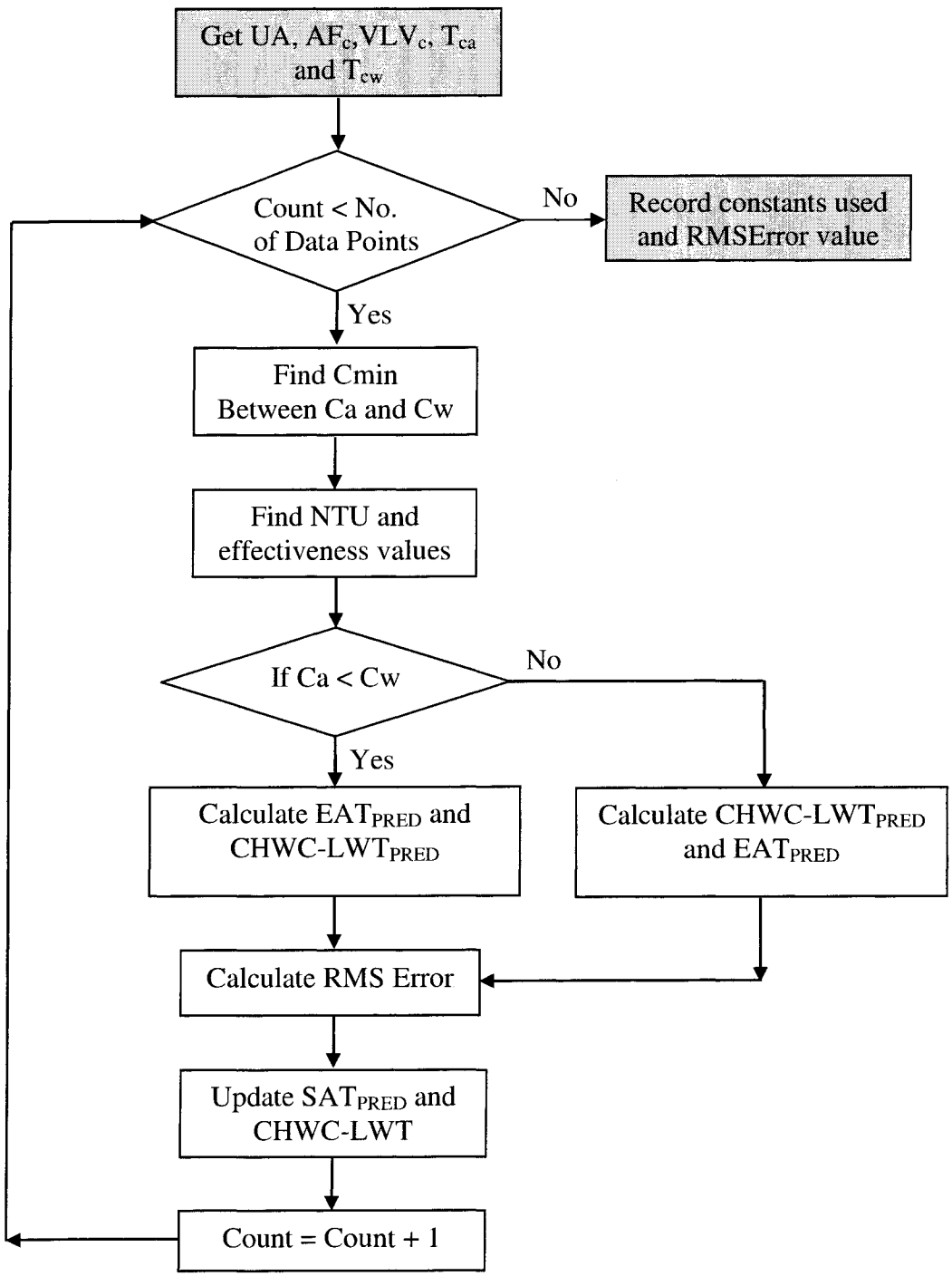

Figure 4.5-2: Steps in development of LCM

To improve the LCM, the constant values in the model were obtained using GAs. The performance of each set of constant values was compared based on the RMS error value obtained for those sets. The RMS error is the root mean square error between SAT and SAT_{PRED}.

$$\text{RMS error} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\text{SAT}-\text{SAT}_{\text{PRED}})^2}; \qquad \ldots 4.5\text{-}17$$

where,

SAT = Value of SAT from the experimental data used

SAT_{PRED} = Value of SAT calculated by LCM

n = number of data points

SAT and CHWC-LWT are updated by SAT_{PRED} and CHWC-LWT_{PRED} respectively for the next iteration.

Thus, lower RMS error is better. For every set of constants, the RMS error value was calculated and recorded along with the constants used. GA is a search tool used to find the optimal values of the five constants, namely; UA, AF_c, VLV_c, T_{ca} and T_{cw}, to get minimum RMS error. Figure 4.5-3 shows a GA flow chart.

To implement this problem using GAs, (See Appendix F for more information on GAs), the constants were represented by defining 60-bit binary digit genes which corresponded to the five constants, namely, UA, AF_c, VLV_c, T_{ca} and T_{cw}. The first set of 12 bits corresponded to a UA value, second set with AF_c, third set with VLV_c, fourth set with T_{ca} and fifth set with T_{cw}. Using 12 digits for each constant provided a resolution of $1/2^{12}$ or 0.000244. All the constants were optimized within a specific range of pre-specified values. The ranges for these pre-specified values were selected based on the results obtained from previous experiments.

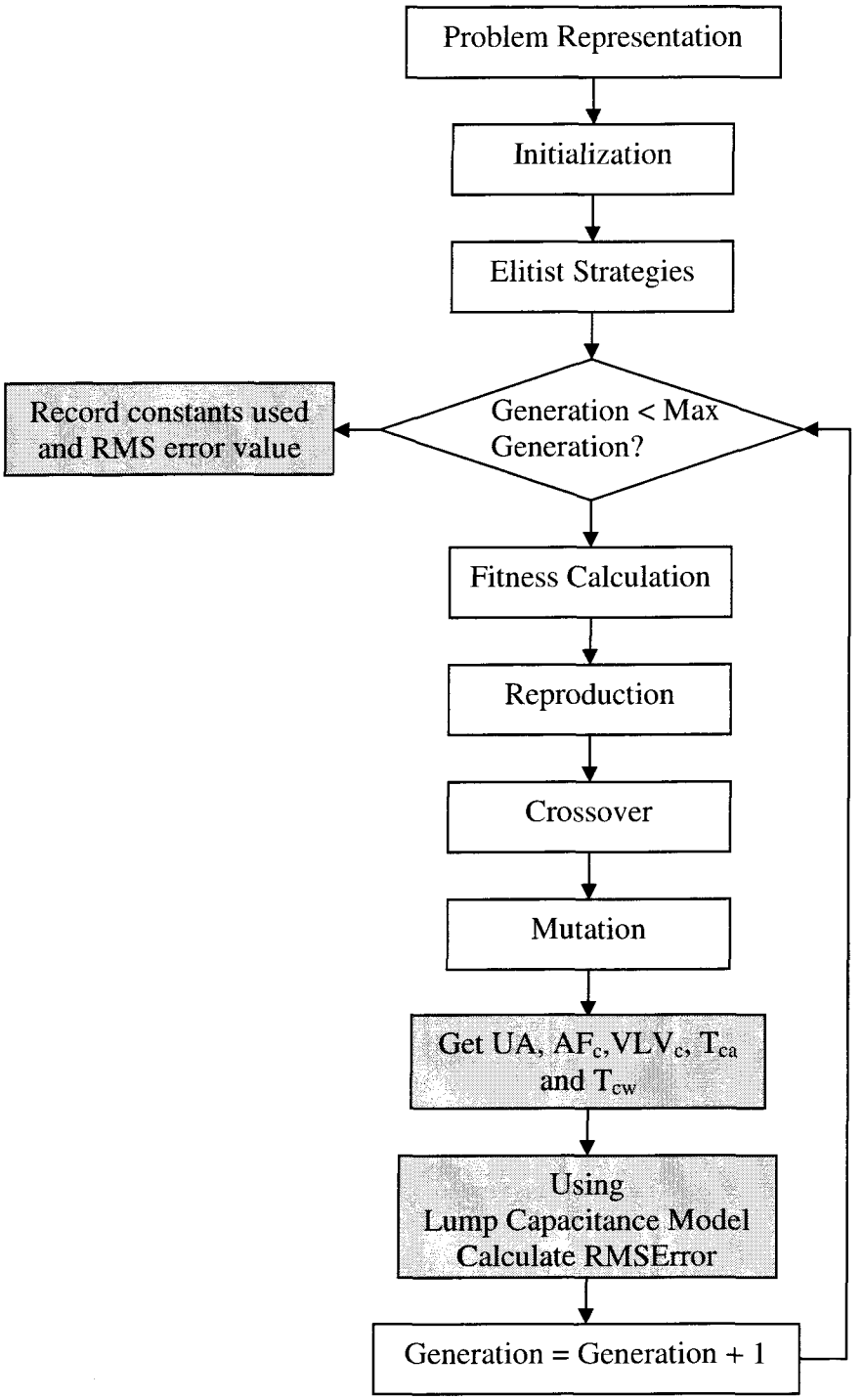

Figure 4.5-3: Flow chart to find optimal values of constants used in LCM

Different GA techniques, namely; single and two point crossovers, single and two point mutations and tournament sizes of 4 and 7 were studied. The best configuration of all the

studied techniques was representing the constants with 60 binary digits and using two point crossover; two point mutations, and a tournament size of 4. This configuration was used to develop the lump capacitance model for this study.

Using results from the chilled water valve position study, the chilled water flow through the valve can be modeled as:

$\dot{m}_w$ = (1.0572*chilled water valve position) - VLV_c; ... 4.5-18

The updated equation, obtained from the flow through the chilled water valve for various valve positions, for the chilled water flow through the valve could be modeled as:

$\dot{m}_w$ = (1.0184*chilled water valve position) - VLV_c; ... 4.5-19

A further generalization of equation 4.5-19 is:

$\dot{m}_w$ = (VLVca*chilled water valve position) - VLVcb; ...4.5-20

where,

VLVca and VLVcb are chilled water flow constants

Thus using above equations, namely 4.5-5, 4.5-18, 4.5-19, and 4.5-20, four different LCMs for SAT_{PRED} studied were:

LCM I

This LCM used GAs to find values of UA, AFc VLVc, Tcw and Tca. A linear relationship between flow through the valve and chilled water valve position was assumed and is given in 4.5-5.

LCM II

This LCM used the corrected equation to find the mass flow rate of chilled water. GAs were used to find values of UA, AFc, Tcw, Tca and VLVc for equation 4.5-18.

LCM III

This LCM used the updated equation to find the mass flow rate of chilled water. GAs were used to find values of UA, AFc, Tcw, Tca and VLVc for equation 4.5-19.

LCM IV

This LCM used GAs to find the values of UA, AFc, Tcw, Tca, VLVca and VLVcb for equation 4.5-20.

For all the lump capacitance models, the same data and GA technique were used to find optimal values for the constants. Results are discussed later.

4.5.3 Summary

In this section, basics for the Lump Capacitance model are discussed. The detail information of different lump capacitance models developed along with the equations are given. Also the application of Genetic Algorithms for finding the constants used in different equations used in the lump capacitance model are briefly explained.

4.6 Summary

In this chapter, different models for predicting SAT using Neural Networks, General Regression Neural Networks and Lump capacitance technique were discussed. The normalization process and need data preprocessing were explained. Also, adaptive NN models were briefly discussed. Different LCMs for predicting SAT were explained.

Chapter 5 Development of a Fuzzy Logic Controller

The FLC was developed and implemented on both Air Handling Unit A (AHUA) and Air Handling Unit B (AHUB) at the Iowa Energy Center's Energy Resource Station (ERS) in Ankeny, Iowa. The FLC was developed using Matlab software.

AHUA and AHUB at the ERS are controlled using a Johnson Controls Metasys system. The communication between the Matlab program and the Metasys system was successfully implemented using Direct Data Exchange (DDE) functions. Appendix B provides more information about some of the DDE commands that were used in this work.

5.1 Basics of Fuzzy Logic Controller

Initially, the FLC was used to control the chilled water coil valve position on AHUA while the standard PID control was used on AHUB. For implementation of the FLC, two inputs are needed, namely error (e, °F) and derivative of error (d, °F/second) to provide a control signal (u, % change in chilled water valve position) for the chilled water coil valve position. The error is the difference between the supply air temperature (SAT) and supply air temperature set point (SATSPT) while the derivative of error is the rate of change of error with respect to the sampling period, de/dt.

5.1.1 Fuzzification

For the first implementation of the FLC, the error and derivative were divided into 7 fuzzy classes as given below:

NL = Negative Large,

NM = Negative Medium,

NS = Negative Small,

ZE = Zero,

PS = Positive Small,

PM = Positive Medium,

PL = Positive Large

Figure 5.1-1, shows the fuzzy membership function (FMF) (going from zero to one on the vertical axis) of the fuzzy variable for various values of error. Figure 5.1-2, shows the FMF (going from zero to one on the vertical axis) of the fuzzy variable for various values of the derivative of the error. Figure 5.1-3, shows the FMF (going from zero to one on the vertical axis) of the fuzzy variable for various values of the output.

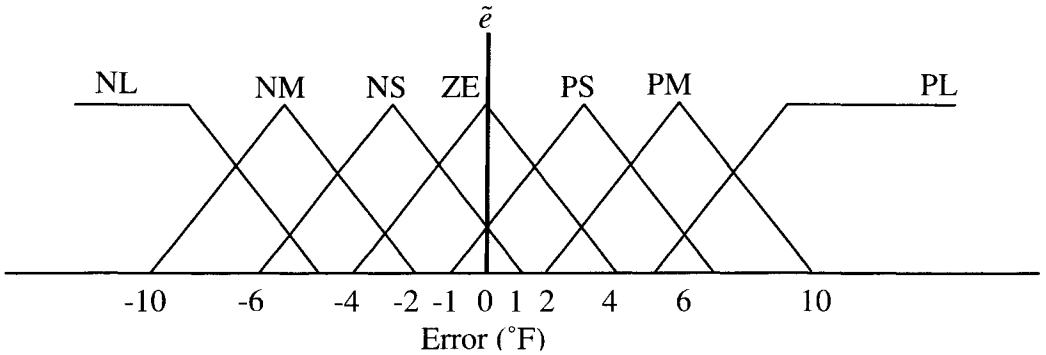

Figure 5.1-1: Fuzzy Membership Function (FMF) for Error

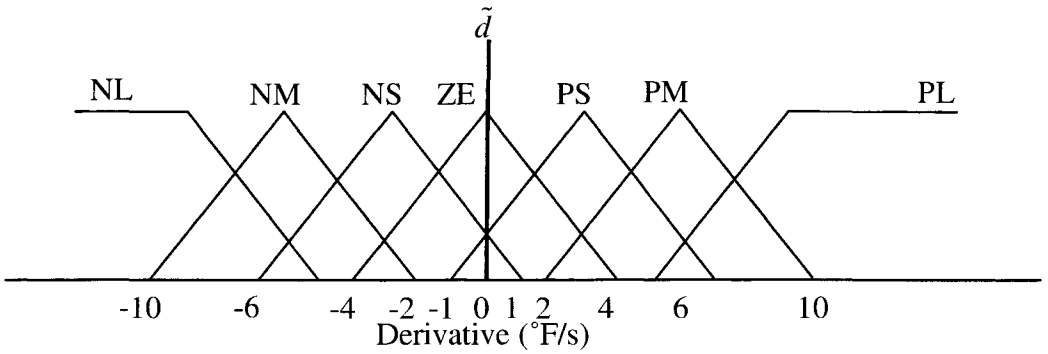

Figure 5.1-2: Fuzzy Membership Function for Derivative of Error

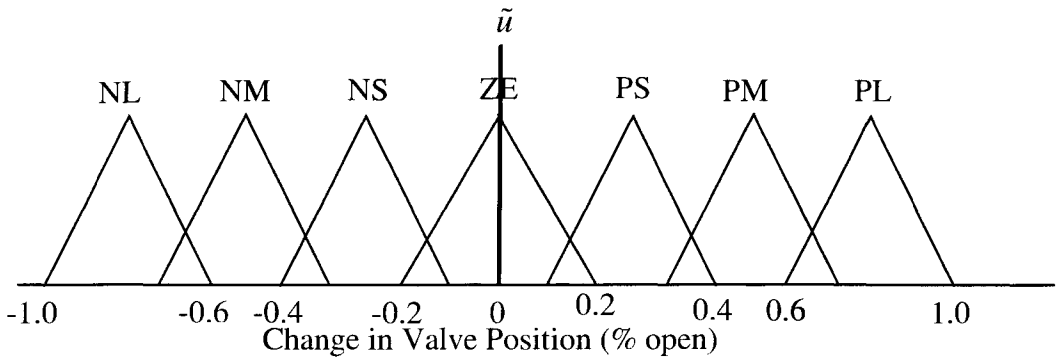

Figure 5.1-3: Fuzzy Membership Function for Control Signal

5.1.2 Fuzzy Rule Matrix

The initial fuzzy rule matrix (FRM) is shown in Table 5.1-1.

Table 5.1-1: Default 7 x 7 Fuzzy Rule Matrix

Error (e)	*Derivative (d)*						
	NL	*NM*	*NS*	*ZE*	*PS*	*PM*	*PL*
NL	NL	NL	NL	NL	NM	NS	**ZE**
NM	NL	NL	NL	NM	NS	**ZE**	PS
NS	NL	NL	NM	NS	**ZE**	PS	PM
ZE	NL	NM	NS	**ZE**	PS	PM	PL
PS	NM	NS	**ZE**	PS	PM	PL	PL
PM	NS	**ZE**	PS	PM	PL	PL	PL
PL	**ZE**	PS	PM	PL	PL	PL	PL

The rule matrix gives the control action for various values of the error and derivative. This matrix represents 49 rules that are used by the FLC. Fuzzy logic allows more than one rule to be active at a time based on the membership of each rule for a particular error and derivative.

To implement a FLC, the error and derivative of error are fuzzified. Using these fuzzified inputs and FRM, the fuzzy control outputs are determined (See Appendix C for details). The fuzzy outputs are defuzzified to obtain a real control signal.

5.1.3 Defuzzification Methods

Defuzzification is the process of converting the various fuzzy output values into one real output for the controller. There are several different methods used for defuzzification. In this study, the centroid method is used. It calculates and returns the center of gravity for the aggregated fuzzy outputs. (See Appendix C for details).

Some criteria for defuzzification methods are:

1. ***Continuity***: A small change in the input variable should not result in a large change in the output.

2. ***Disambiguity***: The defuzzification method should be clear and should not create any ambiguity as sometimes is observed in the Center of Largest Area defuzzification method.
3. ***Computational Complexity***: This criterion is particularly important in practical applications.
4. The defuzzification methods must be clear and should provide correct results.

Once the communication between Matlab and Metasys was implemented using DDE commands, experiments were performed to confirm that the FLC could control the chilled water valve position to maintain the supply air temperature at its setpoint. Experiments demonstrated that the FLC could control the chilled water valve position using the default 49 rules without any tuning.

5.2 Manual Tuning of FLC

In order to improve the FLC performance, tuning of the FLC is necessary. Manual tuning of the FLC was done using the following methods:

- Modifying Fuzzy Rule Matrix
- Modifying specific rules in the Fuzzy Rule Matrix
- Modifying shape of Fuzzy Membership Functions and
- Modifying Scaling Factors

5.2.1 Modifying FRM

The fuzzy rule matrix provides control action for a given set of error and derivative of error values. For the same set of values, a larger fuzzy rule matrix could provide finer control action depending upon the fuzzy rule matrix design.

The default fuzzy rule matrix of 49 rules resulted in unacceptably long settling times. Similar results were obtained after the fuzzy rule matrix was modified manually. Hence, a 9 by 9 fuzzy rule matrix was tried to get better control of the process. The 9 by 9 fuzzy rule matrix used two additional elements in error, derivative of error, and output fuzzy sets, namely:

NVS = Negative Very Small

PVS = Positive Very Small

The new rule matrix is shown in Table 5.2-1.

Table 5.2-1: Default 9 x 9 Fuzzy Rule Matrix (9 by 9)

Error (e)	*Derivative (d)*								
	NL	*NM*	*NS*	*NVS*	*ZE*	*PVS*	*PS*	*PM*	*PL*
NL	NL	NL	NL	NL	NL	NM	NS	NVS	**ZE**
NM	NL	NL	NL	NL	NM	NS	NVS	**ZE**	PVS
NS	NL	NL	NL	NM	NS	NVS	**ZE**	PVS	PS
NVS	NL	NL	NM	NS	NVS	**ZE**	PVS	PS	PM
ZE	NL	NM	NS	NVS	**ZE**	PVS	PS	PM	PL
PVS	NM	NS	NVS	**ZE**	PVS	PS	PM	PL	PL
PS	NS	NVS	**ZE**	PVS	PS	PM	PL	PL	PL
PM	NVS	**ZE**	PVS	PS	PM	PL	PL	PL	PL
PL	**ZE**	PVS	PS	PM	PL	PL	PL	PL	PL

More experiments were performed using the 81 fuzzy rules defined by the 9 by 9 rule matrix shown in Table 5.2-1. It was observed that the FL output was always higher, thus suggesting a need for more tuning.

In order to achieve faster response and increase stability of FLC, two more elements were added to error, derivative of error, and output fuzzy sets, namely:

NT = Negative Tiny

PT = Positive Tiny

The new 11 by 11 rule matrix, with 121 rules is shown in Table 5.2-2.

Table 5.2-2: Default 11 x 11 Fuzzy Rule Matrix

Error	*Derivative (d)*										
(e)	*NL*	*NM*	*NS*	*NVS*	*NT*	*ZE*	*PT*	*PVS*	*PS*	*PM*	*PL*
NL	NL	NL	NL	NL	NL	NL	NM	NS	NVS	NT	**ZE**
NM	NL	NL	NL	NL	NL	NM	NS	NVS	NT	**ZE**	PT
NS	NL	NL	NL	NL	NM	NS	NVS	NT	**ZE**	PT	PVS
NVS	NL	NL	NL	NM	NS	NVS	NT	**ZE**	PT	PVS	PS
NT	NL	NL	NM	NS	NVS	NT	**ZE**	PT	PVS	PS	PM
ZE	NL	NM	NS	NVS	NT	**ZE**	PT	PVS	PS	PM	PL
PT	NM	NS	NVS	NT	**ZE**	PT	PVS	PS	PM	PL	PL
PVS	NS	NVS	NT	**ZE**	PT	PVS	PS	PM	PL	PL	PL
PS	NVS	NT	**ZE**	PT	PVS	PS	PM	PL	PL	PL	PL
PM	NT	**ZE**	PT	PVS	PS	PM	PL	PL	PL	PL	PL
PL	**ZE**	PT	PVS	PS	PM	PL	PL	PL	PL	PL	PL

Using fuzzy rules shown in Table 5.2-3 reasonable results were obtained, but there was still a need for more tuning.

5.2.2 Modifying Selective Rule in the FRM

The performance of the FLC may be improved by modifying a specific rule from the default fuzzy rule matrix. From the real-time test results obtained for the previous 11 by 11 fuzzy rule configuration, it was observed that when the error was PT and the derivative of error was PT, a larger action was taken than required. Similarly, there were other rules for which the fuzzy logic output did not result in the desired action. Those particular outputs were modified. The modified fuzzy rule set is given in the Table 5.2-3 with the modifications from the standard rule matrix highlighted in grey color cells. Results obtained using this modified fuzzy rule set suggested still more need for tuning.

Table 5.2-3: Modified Fuzzy Logic Rule Matrix (11 by 11) (Modified cells highlighted)

Error (e)	*Derivative (d)*										
	NL	*NM*	*NS*	*NVS*	*NT*	*ZE*	*PT*	*PVS*	*PS*	*PM*	*PL*
NL	NL	NL	NL	NL	NL	NL	NM	NS	NVS	NT	**ZE**
NM	NL	NL	NL	NL	NL	NM	NS	NVS	NT	**ZE**	PT
NS	NL	NL	NL	NL	NM	NS	NVS	NT	**ZE**	PT	PVS
NVS	NL	NL	NL	NM	NS	NVS	NT	**ZE**	PT	PVS	PS
NT	NL	NL	NM	NS	NT	NT	**ZE**	PT	PVS	PS	PM
ZE	NL	NM	NS	NT	NT	**ZE**	PT	PVS	PS	PM	PL
PT	NM	NS	NVS	NT	**ZE**	PT	PT	PS	PM	PL	PL
PVS	NS	NVS	NT	**ZE**	PT	PT	PS	PM	PL	PL	PL
PS	NVS	NT	**ZE**	PT	PVS	PS	PM	PL	PL	PL	PL
PM	NT	**ZE**	PT	PVS	PS	PM	PL	PL	PL	PL	PL
PL	**ZE**	PT	PVS	PS	PM	PL	PL	PL	PL	PL	PL

5.2.3 Modifying Shape of Fuzzy Membership Functions:

Modifying the shapes of fuzzy membership functions alters the fuzzy logic output. Increasing the number of fuzzy rules near the center increases sensitivity of the controller to small values of the inputs as shown in Figure 5.2-1. Modifying the shape of fuzzy membership functions also has some effect on the controller's output within the specific region of the input range.

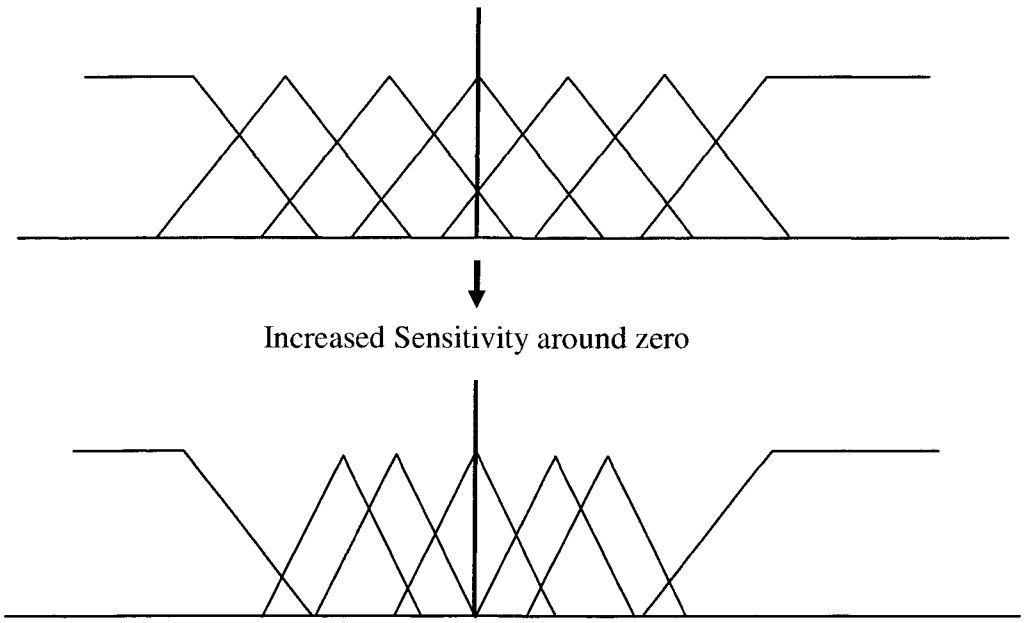

Figure 5.2-1: Modifying shapes of the fuzzy membership functions

5.2.4 Modifying Scaling Factor

A Scaling Factor (SF) can be used to scale the input variables before the fuzzification process. A SF can also be used to scale the output variable after the defuzzification process. Figure 5.2-2 shows the effect of modifying the scaling factor for fuzzy membership functions for the error and derivative of error. For example, the input value of 1 is classified in PL fuzzy membership function when the scaling factor is 1. Same input value, 1, is classified in PM fuzzy membership function when the scaling factor is 2 and in ZE fuzzy membership function when the scaling factor is 10. This changes the sensitivity of the controller to the inputs.

Use of the scaling factor helps in manipulating input and output variables and it modifies the controller's output uniformly across the input range.

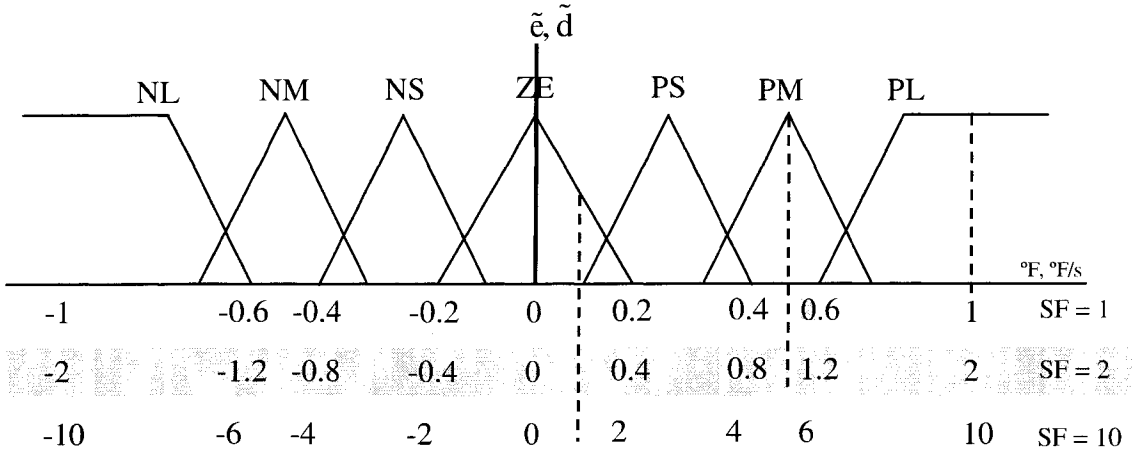

Figure 5.2-2: Effect of modifying Scaling Factor

5.3 Adaptive Fuzzy Logic Controller

5.3.1 Introduction

Most of the real world processes that require automatic control are nonlinear in nature and their parameter values alter as the operating point changes. A typical controller can only be tuned to give good performance at a particular operating point or for a limited period of time. These controllers need to be retuned if the operating point changes. This necessity to retune has driven the need for adaptive controllers that can automatically retune themselves to match the changes made in the operating point and/or current process characteristics.

An adaptive controller contains two components, namely a process monitoring component and a self adaptation mechanism. In the process monitoring component, changes in the process characteristics are detected and performance of the controller is measured. In the adaptation mechanism, process monitoring information is used to update the controller parameters and adapt the controller to the changes in the process.

A FLC contains a number of parameters that can be altered to modify the controller performance. These parameters are:

1. Fuzzy Rule Matrix
2. Scaling Factor

3. Shape of Fuzzy Membership Functions

5.3.2 Adaptation Mechanism

The adaptation mechanism must modify the controller parameters to improve the controller performance on the basis of the output from the performance indices. Adaptation mechanisms for FLCs can be classified according to the parameters being adjusted. One set of parameters which can be adjusted is the scaling factors. Figure 5.2-2 shows the effect of modifying the scaling factor for fuzzy membership functions for the error and derivative of error. Figure 5.2-2 shows now sensitivity of the controller is changed for the input values.

Following Figure 5.2-2, if, for example, the input value is multiplied by a scaling factor of 2, the input is mapped from -2 to 2. Similarly, if the scaling factor is 10, the input is mapped to the range from -10 to 10. Changing the scaling factor changes the sensitivity of the controller to the input parameters. Use of the scaling factor helps in manipulating input and output variables and it modifies the controller's output uniformly across the input range.

Another mechanism to adjust the sensitivity is to modify the shapes of the fuzzy membership function. An example where the fuzzy membership functions are altered to increase the sensitivity of the controller to small values of the inputs is shown in Figure 5.2-1.

Modifying scaling factor alters sensitivity uniformly across the entire operating region of input/output whereas changing the fuzzy membership functions modifies sensitivity within specific region of the range of input/output.

This adaptive mechanism is to modify the FRM. Modifying FRM is altering fuzzy rules and thereby the action taken by FLC.

5.3.2.1 Adaptive Mechanism by Modifying Scaling Factor and Fuzzy Membership Functions

Evolutionary Strategies (ES) were used to study the adaptive mechanism for modifying scaling factors and fuzzy membership functions. Evolutionary Strategies (ES) are stochastic search methods that mimic the metaphor of natural biological evolution (See Appendix G for

details). Evolutionary Strategies operate on a single parent gene which produces a single offspring gene of potential solutions. Applying the principle of survival of the fittest between the parent and offspring will produce better and better approximations to a solution. Figure 5.3-1 shows a flow chart for ES.

At each generation, a new offspring gene is created by mutating the parent gene. Mutation is a operation which promotes the exploration of new regions in the search space and ensures that all the points in the solution space have a chance of being searched. Since binary numbers are used in the representation system, mutation was performed by replacing a few bits from the original gene with the randomly generated binary numbers.

The RMS error for the offspring is then computed. If the RMS error (RMS $error_O$) value for this offspring is less than that for the parent (RMS $error_P$), then the parent gene is replaced by offspring gene. Then, for the next generation, this offspring gene will be the parent gene. This cycle is iterated until the optimization criteria or the predefined number of generations is reached. If RMS error value for offspring gene is higher than that for parent, offspring gene is discarded.

This process of generating offspring gene leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.

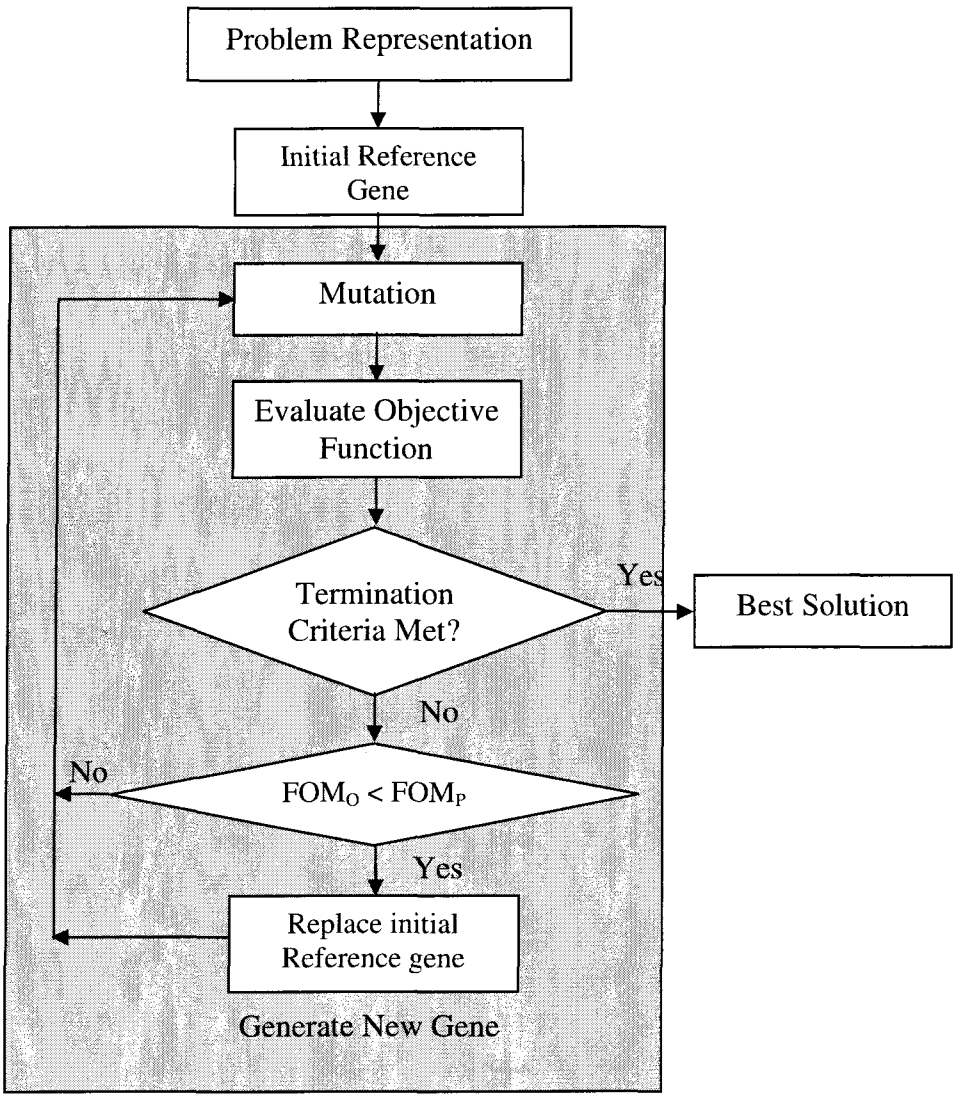

Figure 5.3-1: Flow chart for Evolutionary Strategies

ES was used to modify two different FLC parameters, namely;

1. evolving scaling factors, and
2. evolving mapping factors which define fuzzy membership functions in error, derivative of error and control signal

5.3.3 Use of ES for Evolving Scaling Factor:

One of the techniques using ES for development of an adaptive FLC was evolving scaling factors to get better FLC performance. Using ES techniques, real-time tests were conducted

to develop an adaptive FLC by evolving scaling factors. Figure 5.3-2 shows the flow chart for evolving the scaling factors.

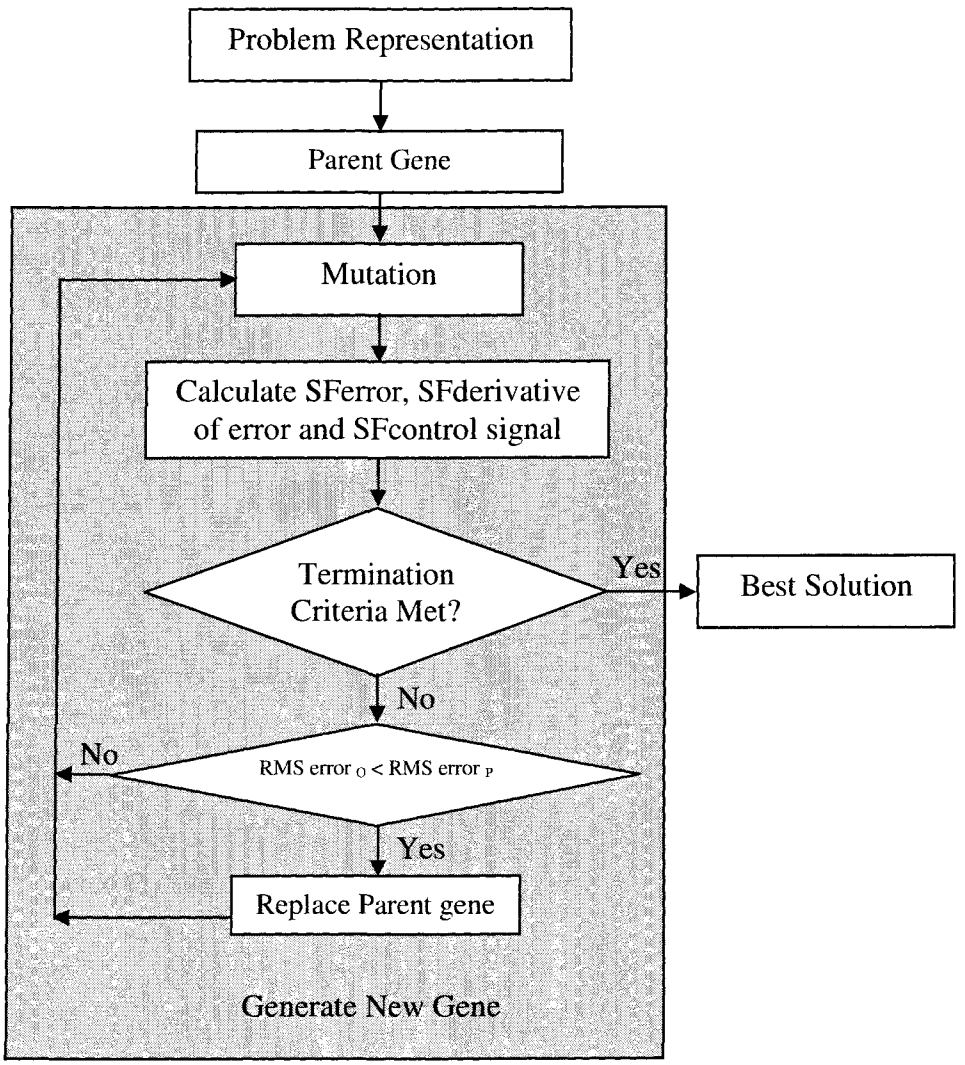

Figure 5.3-2: Flow chart for evolving scaling factor using ES

Problem Representation

In this case, a 12 bit binary number was evolved which defines the scaling factors namely SFerror, SFderivative of error and SFcontrol signal. First 4 bits define the SFerror value, next 4 bits define SFderivative of error and last 4 bits define SFcontrol signal and are shown below:

Bit Number	1	2	3	4	5	6	7	8	9	10	11	12
Bit Value	0	1	1	0	1	1	0	0	1	0	1	0
Scaling Factor (SF)	SFerror				SFderivative of error				SFcontrol signal			

Figure 5.3-3: Representation of scaling factor problem for ES

Once a gene is generated and using the pre-defined maximum and minimum limits for each scaling factors, scaling factors are calculated as below:

SFerror:

SFerror is defined by first 4 bits in the gene and calculated as:

gene = [0 1 1 0];

$SFe_{update} = 0$;

SFe_{min} = minimum value;

SFe_{max} = maximum value;

for i = 1:1:4

j = 4-i;

$SFe_{update} = SFe_{update} + gene(1,i) * 2^{(j)}$;

end

$SFerror = SFe_{min} + SFe_{update} * 0.0667 * (SFe_{max} - SFe_{min})$;

0.0667 is the resolution and is calculated as $1/ 2^4$.

Similarly, SFderivative of error and SFcontrol signal values are calculated using the appropriate binary bits from the parent gene.

In the next generation, few bits from the parent gene are randomly mutated to reproduce a new offspring (See Appendix G for details). Minimum and maximum values of each scaling

factor were selected based on previous experimental results and the researcher's insight about the process.

5.3.4 Use of ES for Evolving Mapping Factor

Mapping Factor

A Mapping Factor is a number which modifies uniform fuzzy membership functions to non-uniform fuzzy membership functions as shown in Figure 5.3-4. Three different mapping factors are evolved: powere, powerd, and poweru. These three numbers represent how the shape of the fuzzy membership functions in error, derivative of error and control signal will be mapped to non-uniform fuzzy membership functions. For example, Figure 5.3-4 shows non-uniform fuzzy membership function in error if mapping factor is 4 and calculated as shown below:

powere = 4;
errorplr = 5
errorpmr = 1
errorpll = 0.8^4
errorpsr = 0.8^4
errorpml = 0.6^4
errorpvsr = 0.6^4
errorpsl = 0.4^4
errorptr = 0.4^4
errorpvsl = 0.2^4
errorzer = 0.2^4
errorptl = 0

errorplc = (errorplr + errorpll)/2;
errorpmc = (errorpmr + errorpml)/2;
errorpsc = (errorpsr + errorpsl)/2;
errorpvsc = (errorpvsr + errorpvsl)/2;
errorptc = (errorptr + errorptl)/2;

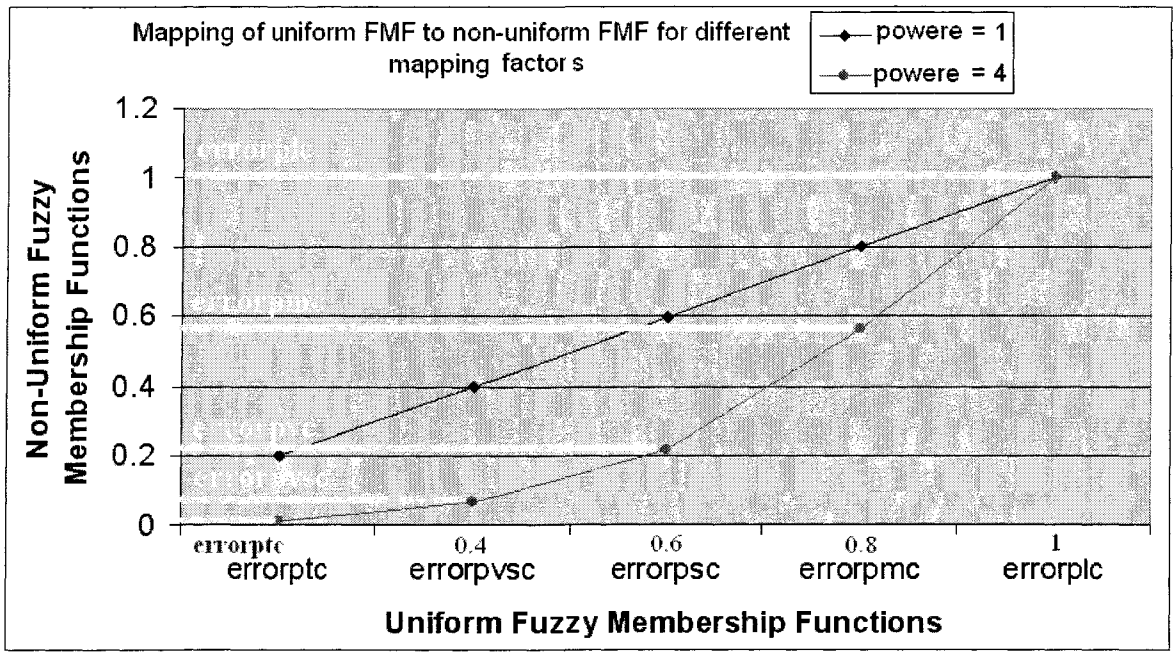

Figure 5.3-4: Mapping of uniform FMF to non-uniform FMF

Problem representation

In order to evolve mapping factor, each of the membership functions, which are triangular, was given three variables, namely right, left and center variables. For example, Positive Tiny (PT) fuzzy membership function was divided into PT-right (PTR), PT-left (PTL) and PT-center (PTC) as shown in Figure 5.3-5. Similarly all other fuzzy membership functions where divided into right, left and center variables. Figure 5.3-5 shows the division of few FMFs.

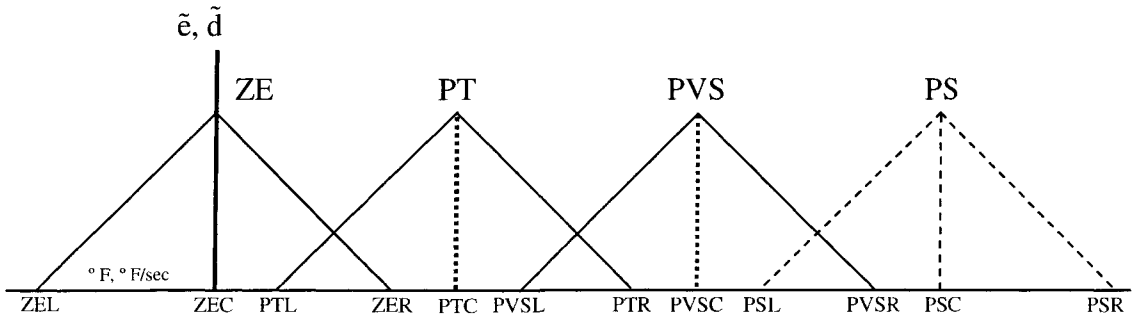

Figure 5.3-5: Defining FMF in three variables

As introduced in the FLC section, Figure 5.3-6 shows uniform fuzzy membership functions along with the center of each membership function.

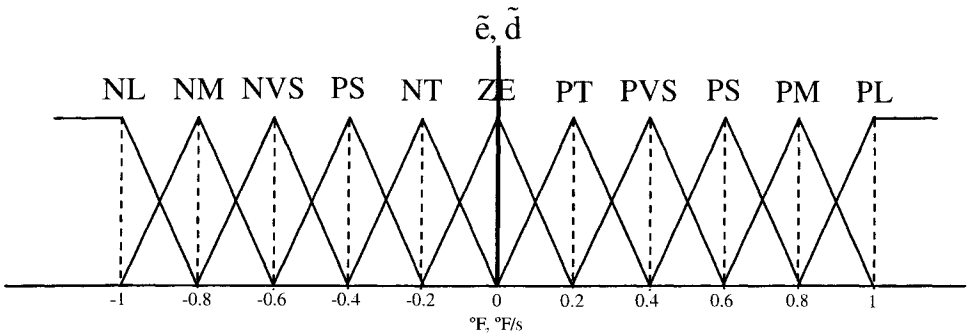

Figure 5.3-6: Uniform fuzzy membership functions in error and derivative of error

On the basis of this uniform fuzzy membership function, non-uniform fuzzy membership functions were calculated as:

errorplr = 5

errorpmr = 1

errorpll = 0.8^powere

errorpsr = 0.8^powere

errorpml = 0.6^powere

errorpvsr = 0.6^powere

errorpsl = 0.4^powere

errorptr = 0.4^powere

errorpvsl = 0.2^powere

errorzer = 0.2^powere

errorptl = 0

errornll = -errorplr;

errornml = -errorpmr;

errornlr = -errorpll;

errornsl = -errorpsr;

errornmr = -errorpml;

errornvsl = -errorpvsr;

errornsr = -errorpsl;

errorntl = -errorptr;

```
errornvsr    = -errorpvsl;
errorzel     = -errorzer;
errorntr     = -errorptl;

errorplc     = (errorplr+errorpll)/2;
errornlc     = -errorplc;
errorpmc     = (errorpmr+errorpml)/2;
errornmc     = -errorpmc;
errorpsc     = (errorpsr+errorpsl)/2;
errornsc     = -errorpsc;
errorpvsc    = (errorpvsr+errorpvsl)/2;
errornvsc    = -errorpvsc;
errorptc     = (errorptr+errorptl)/2;
errorntc     = -errorptc;
```

Similarly, all the fuzzy membership functions for derivative of error and control signal were calculated using powerd and poweru values respectively.

Another technique used for development of an adaptive FLC was evolving mapping factors using ES, to obtain better FLC performance. Using ES, real-time tests were conducted to develop an adaptive FLC by evolving mapping factors. Figure 5.3-7 shows a flow chart for evolving mapping factors.

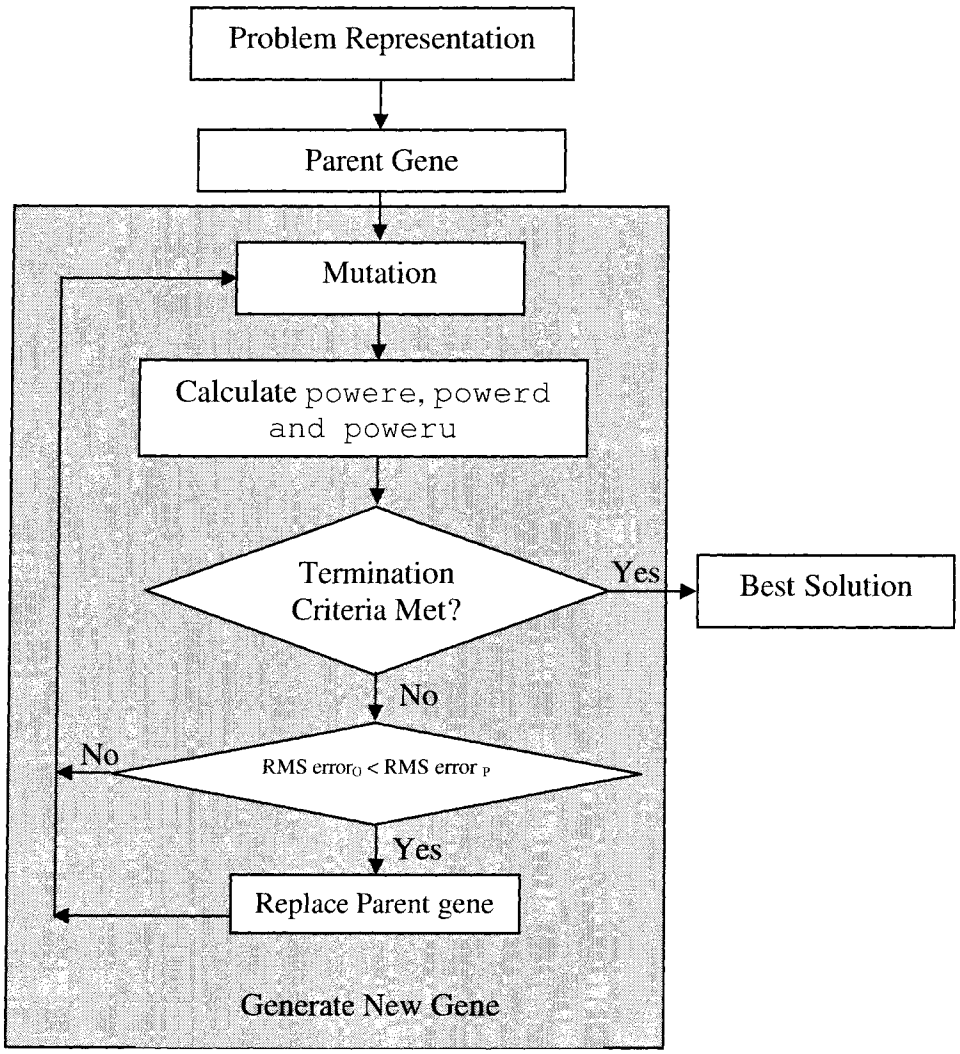

Figure 5.3-7: Flow chart for evolving mapping factor using ES

In this case, a 12 bit binary number was being evolved which defines mapping factors, namely powere, powerd and poweru. The first set of 4 bits defines the powere value, the second set of 4 bits defines the powerd value and the last set of 4 bits defines the poweru value and show in Table 5.3-1

Table 5.3-1: Problem Representation of scaling factor problem for ES

Bit Number	1	2	3	4	5	6	7	8	9	10	11	12
Bit Value	0	1	1	0	1	1	0	0	1	0	1	0
Mapping Factor	powere				powerd				poweru			

Once a gene is generated and maximum and minimum limits for each mapping factor, then powere value is calculated as defined below:

powere

powere is defined by first 4 bits in the gene and calculated as given below:

gene = [0 1 1 0];

$powere_{update}$ = 0;

$powere_{min}$ = minimum value;

$powere_{max}$ = maximum value;

for i = 1:1:4

j = 4-i;

$powere_{update} = powere_{update} + gene(1,i) * 2^{(j)}$;

end

$powere = powere_{min} + powere_{update} * 0.0667 * (powere_{max} - powere_{min})$;

0.0667 is resolution and calculated as $1/ 2^4$.

Similarly powerd and poweru values are calculated using the appropriate binary bits from the gene. Once powerd and poweru are obtained, every fuzzy membership function can be calculated as explained before.

In the next generation, a few bits from the parent gene are randomly mutated to reproduce a new offspring. Minimum and maximum values for each mapping factor were selected based on previous experimental results and the researcher's insight about the process.

Figure 5.3-8 shows the variation in the fuzzy membership functions for different values of powere. Similarly, variation in the fuzzy membership functions for different values of powerd and poweru can be shown.

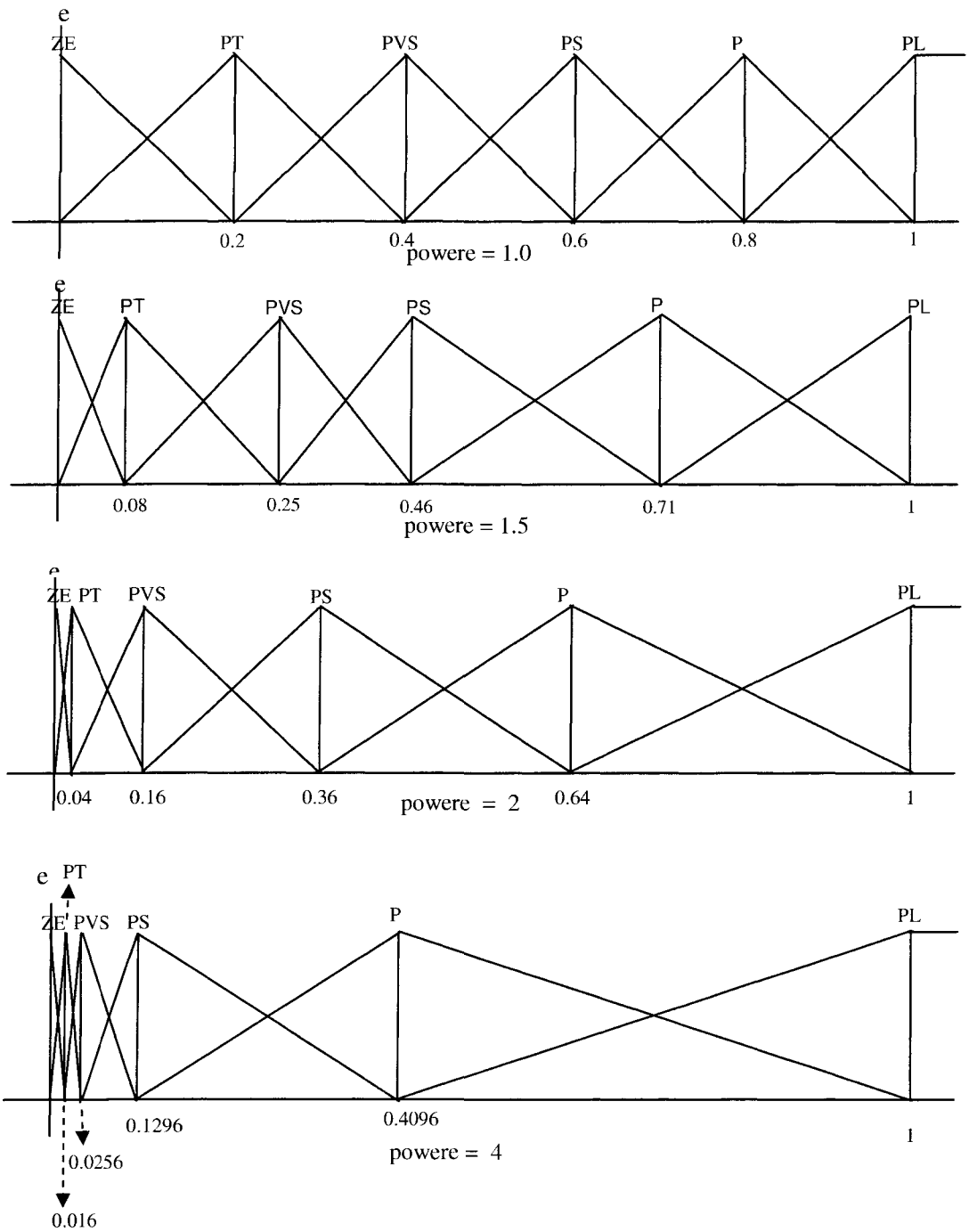

Figure 5.3-8: Mapping of Uniform Fuzzy Membership function to Non-uniform Fuzzy Membership Functions in error for powere value of 1.5, 2 and 4

5.3.5 Use of GA for Evolving FRM

Another adaptive mechanism was to modify the FRM using Genetic Algorithms. Based upon the values in the FRM, control actions are taken by FLC for that particular system. For most applications, it is difficult to optimize the rule bases and membership functions manually.

Table 5.3-2 shows a 49 rule FRM based on human intuition and expertise. This human intuition based FRM is very basic and designed based on the designer's experience with various system operating parameters, like system control, types of flow control valves, and response times of various equipment in the system. Performance may be sub-optimal in some situations.

Table 5.3-2: 7 x 7 Human FRM

Error (e)	Derivative (d)						
	NL	*NM*	*NS*	*ZE*	*PS*	*PM*	*PL*
NL	NL	NL	NL	NL	NM	NS	**ZE**
NM	NL	NL	NL	NM	NS	**ZE**	PS
NS	NL	NL	NM	NS	**ZE**	PS	PM
ZE	NL	NM	NS	**ZE**	PS	PM	PL
PS	NM	NS	**ZE**	PS	PM	PL	PL
PM	NS	**ZE**	PS	PM	PL	PL	PL
PL	**ZE**	PS	PM	PL	PL	PL	PL

If FLC performance for particular FRM is unacceptable, then the FRM must be modified. An example of a modified 49 rules FRM is shown in Table 5.3-3.

Table 5.3-3: Modified 7 x 7 Human FRM

Error (e)	Derivative (d)						
	NL	*NM*	*NS*	*ZE*	*PS*	*PM*	*PL*
NL	NL	NL	NL	NL	NM	**ZE**	**ZE**
NM	NL	NL	NL	NM	**ZE**	**ZE**	**ZE**
NS	NL	NL	NM	**ZE**	**ZE**	**ZE**	PM
ZE	NL	NM	**ZE**	**ZE**	**ZE**	PM	PL
PS	NM	**ZE**	**ZE**	**ZE**	PM	PL	PL
PM	**ZE**	**ZE**	**ZE**	PM	PL	PL	PL
PL	**ZE**	**ZE**	PM	PL	PL	PL	PL

Using a modified FRM, real-time tests were conducted. For real-time tests, it was observed that the modified FRM improved the system performance. In order to improve the system performance further and develop an adaptive FLC, it was necessary to develop a continuous adaptive technique to modify this FRM. One method considered was to start with a modified FRM and check the system performance for this modified FRM. If the system performance is better than the human FRM, then the modified FRM is retained for future modification. Later, changes are made in this modified FRM and system performance is rechecked. This process of generating modified FRMs is repeated until satisfactory near optimal performance of the system is achieved. But working with the manual process of modifying FRMs is very time consuming and expensive. Also, a limited number of cases can be studied when using the controller in a real building. As the number of rules in the FRM increases, the complexity of finding a better FRM also increases.

Genetic Algorithms overcome the problems discussed above. A Genetic Algorithm is an optimization technique based on the genetic processes found in nature. It is very flexible and robust and thus makes it suitable for optimization problems.

The GA is a stochastic search algorithm developed from the mechanics of natural selection and natural genetics (Goldberg 1989). Compared with other stochastic search methods, GA has the following features (Goldberg 1989):

1. GA works with a coding of the parameter set, not the parameters themselves;
2. GA searches from a population of points (multiple points), not a single point;
3. GA uses the objective function information rather than the derivatives or other auxiliary knowledge; and
4. GA uses probabilistic transition rules, not deterministic rules.

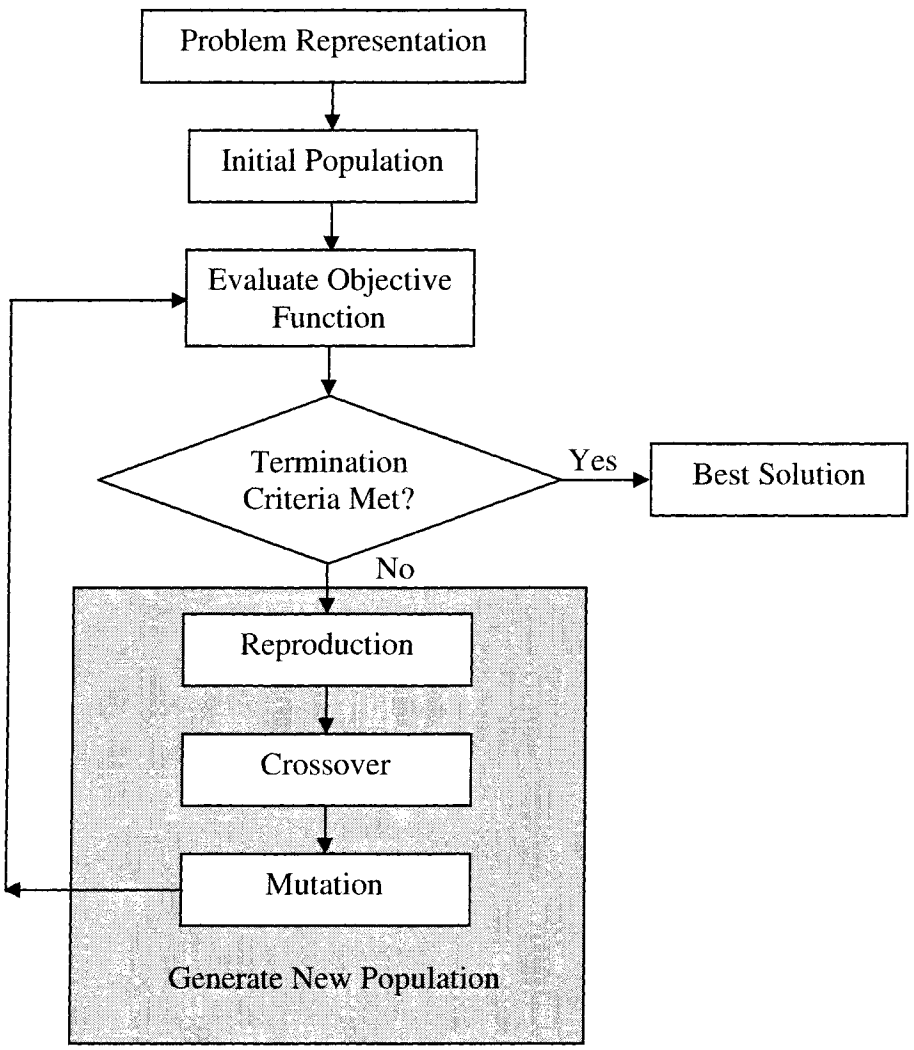

Figure 5.3-9: Flow chart for GA

Figure 5.3-9 shows the basic procedure that is used during a GA optimization. The user supplies n initial guesses or these could be random guesses for the design variables as the initial population. For example, if an optimization problem involves k design variables β_1 to β_k, then initial population is the following n vectors:

$$\left\{\begin{matrix}\beta_{11}\\ \beta_{21}\\ \beta_{31}\\ .\\ .\\ .\beta_{k1}\end{matrix}\right\},\left\{\begin{matrix}\beta_{12}\\ \beta_{22}\\ \beta_{32}\\ .\\ .\\ .\beta_{k2}\end{matrix}\right\},\left\{\begin{matrix}\beta_{13}\\ \beta_{23}\\ \beta_{33}\\ .\\ .\\ .\beta_{k3}\end{matrix}\right\},....\left\{\begin{matrix}\beta_{1n}\\ \beta_{2n}\\ \beta_{3n}\\ .\\ .\\ .\beta_{kn}\end{matrix}\right\}$$

For each vector, the objective function is calculated and compared. The vector that generates the best value of the objective function is called the best "parent". A second population is generated based on the information of the objective functions corresponding to each design variable. Depending on problem definition, the objective function may be either minimization or maximization of the value. The goal is to generate new populations so that the "features" that make one vector yielding better values of the objective function remain in the new population. Once a new population is generated, the values of the objective function are calculated and compared again. And the third population is then generated from good parents that yield better values of the objective function. This process is repeated until certain optimization criteria are satisfied. For example, the criteria could state that the optimized value of the objective function from the m^{th} population compared with the one from the n^{th} population is less than 10^{-5}. The best parent from the last population is the optimized solution. During the process of generating new populations, the design variable vectors (parents) are coded into 0 and 1 genes such as:

$$\left\{\begin{matrix}\beta_{11}\\ \beta_{21}\\ \beta_{31}\\ .\\ .\\ .\beta_{k1}\end{matrix}\right\},\left\{\begin{matrix}\beta_{12}\\ \beta_{22}\\ \beta_{32}\\ .\\ .\\ .\beta_{k2}\end{matrix}\right\},\left\{\begin{matrix}\beta_{13}\\ \beta_{23}\\ \beta_{33}\\ .\\ .\\ .\beta_{k3}\end{matrix}\right\},....\left\{\begin{matrix}\beta_{1n}\\ \beta_{2n}\\ \beta_{3n}\\ .\\ .\\ .\beta_{kn}\end{matrix}\right\}\rightarrow 11010,10010,11100....00101$$

The words reproduction and crossover shown in Figure 5.3-9 represent processes to generate new populations from old populations guided by the information of the objective function for each vector and are not introduced in this thesis (See Appendix F, Goldberg 1987 for more information). To prevent converging to a local optimal solution, a process called Mutation, which generates a new vector randomly, is involved in the process. The description of these processes is outside of the scope of this study.

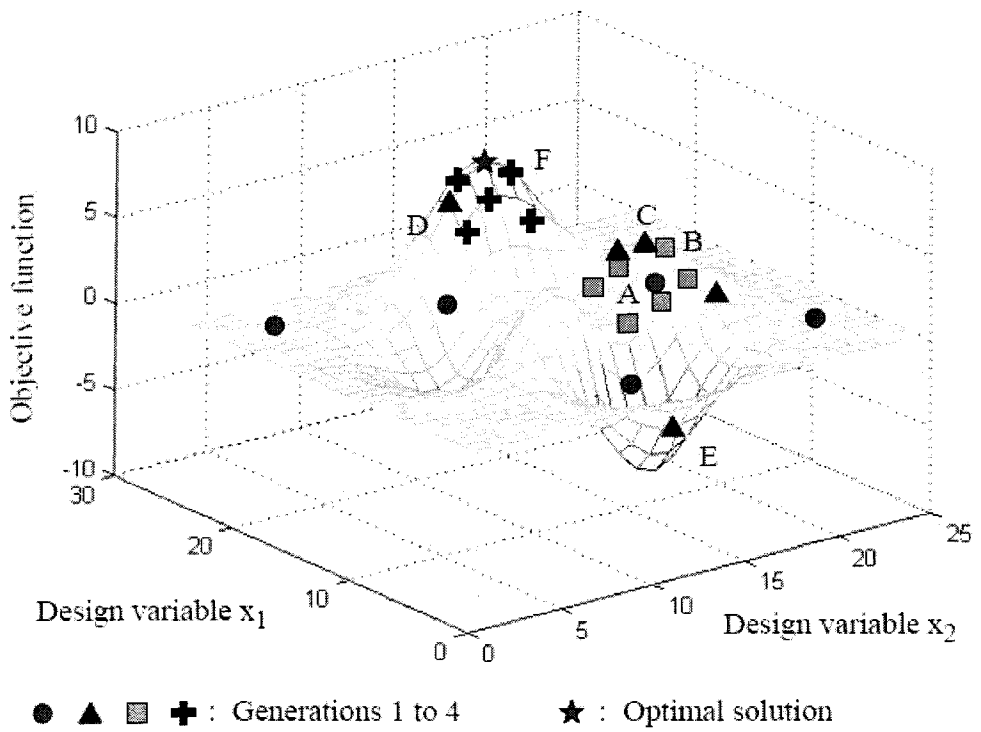

Figure 5.3-10: Illustration for GA search results (Wen 2003)

To have a better understanding for the GA searching process, an optimization (maximization) problem is illustrated in Figure 5.3-10. The star on the contour stands for the optimal solution that generates the maximum objective function. The round-shape points stand for the initial population. It is observed that among the points in the initial population, point A has a higher value of the objective function. The second generation, which is represented by the square-shape points, is generated to be close to point A by reproduction and crossover processes. Point B stands for the best parent in the second generation. If no mutation process is

involved, then the local optimal solution - point C from the third population (represented by the triangular-shape points) is believed as the optimal point. By using the mutation process, two random points, point D and E are generated in the third population. The algorithm then generates the next population near point D among which the point F is the best parent. The process is repeated until it finds the optimal point. The population size, gene length, optimization criteria, and mutation frequency are all adjustable.

Using random numbers, a GA generates the values for each element in a FRM. The process of generating the FRM using random numbers is called as an Initialization Process.

5.3.5.1 Different Initialization Methods:

Depending upon the method used to generate the random numbers for the FRM, different initialization methods studied were:

1. Completely Random Method,
2. Changing Human FRM by random number generated amongst -1, 0, 1
3. Changing Human FRM by random number generated amongst –1, 0, 1 without changing specific elements from the Human FRM
4. Changing Human FRM by random number generated amongst –1, 0, 1 without changing specific elements in the outer loop from the Human FRM

Completely Random Method

In the completely random method for the initialization process, FRMs are developed by generating random numbers amongst the different possible Fuzzy Rules Actions. Possible fuzzy actions were divided into 11 numerical values staring from 1 to 11, where 1 corresponded to NL and 11 corresponded to PL.

Changing Human FRM by random number generated amongst -1, 0, and 1

In this method for the initialization process, FRMs were generated by modifying every element in the human FRM by randomly generated numbers amongst -1, 0, and 1.

Changing Human FRM by random number generated amongst –1, 0 and 1 without changing the specific elements from the Human FRM

In this method for the initialization process, FRMs were generated by modifying every element in the human FRM, except the elements in the four corners and the center element, by randomly generated numbers amongst -1, 0, and 1.

Changing Human FRM by random number generated amongst –1, 0 and 1 without changing the specific elements in the outer loop from the Human FRM

In this method for the initialization process, FRMs were generated by modifying every element in the human FRM, except the elements in the outer loop, by randomly generated numbers amongst -1, 0, and 1. Using this type of FRM assured that when the system is out of control or is progressing towards the extreme conditions correct action is taken.

5.4 Summary

In this chapter, different methods used for development and implementation of an adaptive FLC were discussed. Adaptive mechanisms used for modifying scaling factors and fuzzy membership functions using Evolutionary Strategies were presented in detail. Also, adaptive mechanisms for evolving mapping factors in fuzzy membership functions were presented. Another method for developing an adaptive FLC using Genetic Algorithms for evolving the fuzzy rule matrix was presented.

Chapter 6 Results for manually tuned FLC and SAT models

6.1 Results for Manual Tuned FLC

Real-time tests were conducted using the FLC for an initial FRM of 49 rules and then using 121 rules. Also, experiments were conducted with modified FRMs and scaling factors.

6.1.1 Real-time Results for 49 Rules FRM

Initially, experiments were conducted using a default 49 rule FRM. Then the FRM was manually tuned by changing scaling factors and the fuzzy rule matrix.

6.1.1.1 Real-time results for FLC using default 49 rules FRM

Real-time tests were conducted starting with the initial default 49 rules FRM given in Table 5.1-1. Figure 6.1-1 shows real-time results for a FLC having 49 default rules FRM. This test demonstrated usefulness of FLC for controlling SAT by changing the chilled water valve position.

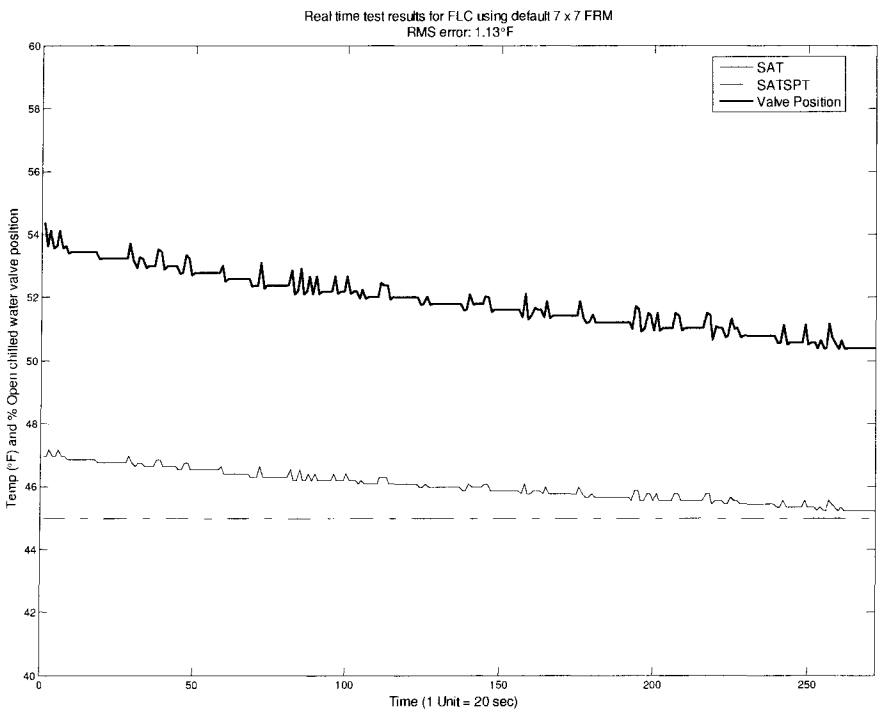

Figure 6.1-1: Real-time test results for FLC using default 49 rules FRM

Due to the long settling time for the SAT, the FLC performance was unacceptable. To improve the FLC performance, manual tuning was done by using the scaling factors and then changing a few elements in the 49 rules FRM.

6.1.1.2 Results for FLC by changing scaling factor in error and derivative of error for 49 rules FRM

To achieve faster response with lower settling times, the scaling factors were increased. Increasing the scaling factor makes the controller response faster and thus attains the required set-point earlier. Real-time tests were conducted using the default 49 rules FRM but fuzzy membership functions in error and derivative of error were multiplied by scaling factors of 5. Results in Figure 6.1-2 shows that using scaling factor is not helpful. More real-time tests were conducted with different scaling factors but none resulted in acceptable FLC performance.

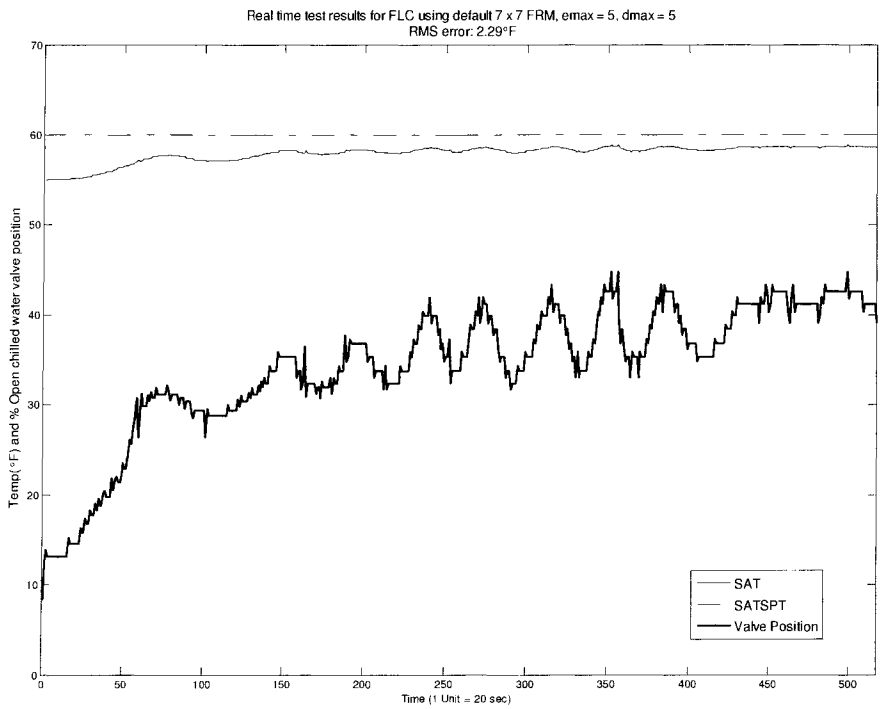

Figure 6.1-2: Real-time test results for FLC using default 49 rule FRM and SF of 5 for error (e) and derivative of error (d)

6.1.1.3 Results for manual tuning of FLC by changing few elements from the 49 rules FRM

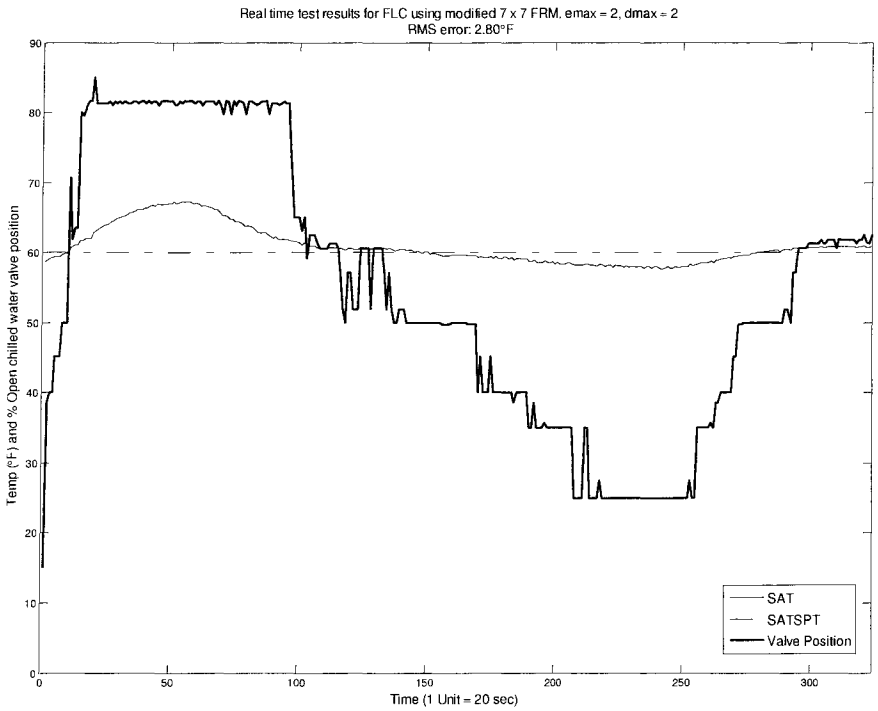

Figure 6.1-3: Case I – Real-time test results for FLC using modified 49 rules FRM

When altering the scaling factor did not work, real-time tests with manual tuning of the FLC by changing few elements in the default 49 rules FRM were conducted. But, as shown in Figure 6.1-3, no significant improvement in the FLC performance was observed.

Real-time results show that with the modified FRM, the FLC does not execute maximum action, i.e. chilled water valve position was not opened to entire extent. If maximum action would have been taken, the valve should have had achieved the 100% open position to reduce the error sooner. Further modifications were made in the 49 rules FRM. Figure 6.1-4 shows real-time test results with this modified FRM which shows little improvement in performance from the previous ones.

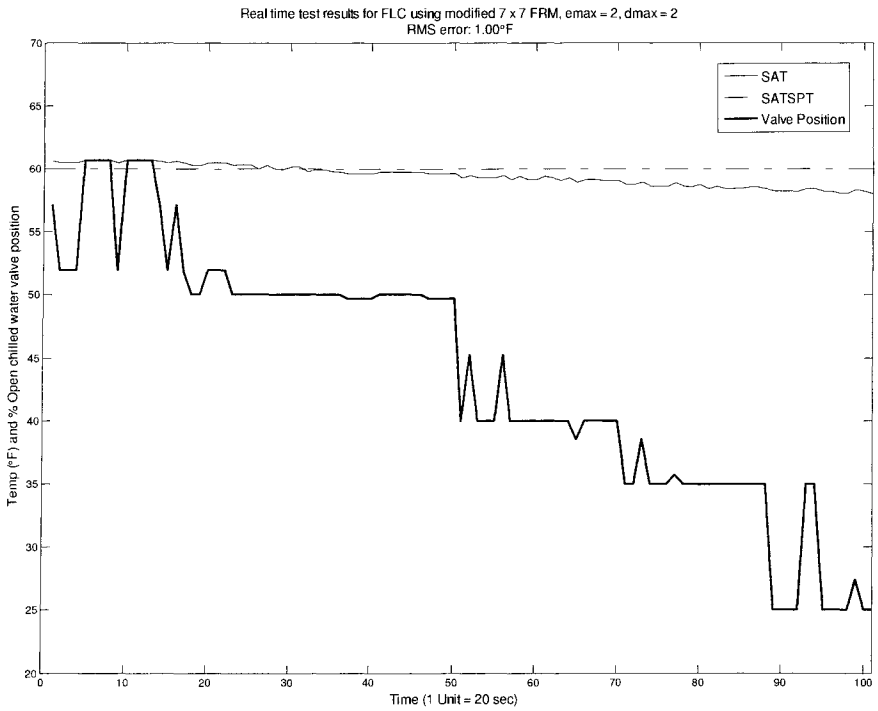

Figure 6.1-4: Case II – Real-time test results for FLC using modified 49 rules FRM

Though the modified 49 rules FRM results in better control for one set-point, its response is unacceptable when the set-point is changed (results not shown). Thus, the FRM having 49 rules results in unacceptably long settling times. Hence, experiments were performed with a larger 9 by 9 FRM having 81 fuzzy rules. But, similar unacceptable results were obtained. So, a larger 11 by 11 FRM was developed which allows 121 fuzzy rules.

6.1.2 Real-time test results for 121 rules FRM

For real-time tests using a 49 rule FRM poor FLC performance was observed, hence to have better FLC performance a 121 rule FRM was developed.

6.1.2.1 Real-time test results for default 121 rules FRM

For the initial real-time tests using a default 121 rules FRM, larger oscillatory responses with higher chilled water valve movement were observed as shown in Figure 6.1-5. Table 5.2-2 shows default 121 rules FRM. Unacceptable results for the FLC lead to further studies with manual tuning of FRMs.

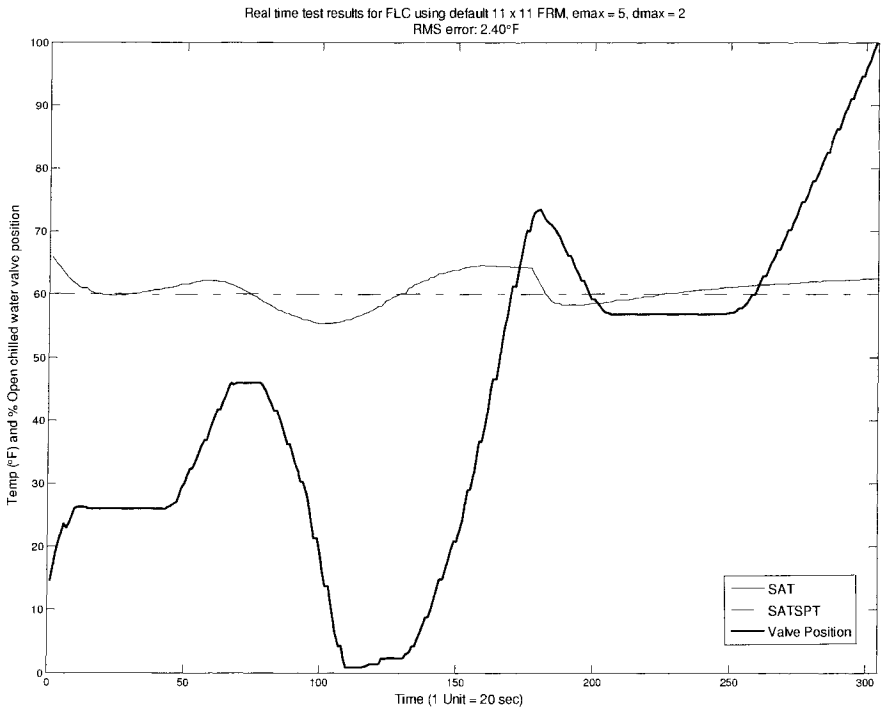

Figure 6.1-5: Real-time test results for FLC using default 11 x 11 rules FRM

As discussed above, manual tuning was done by modifying the scaling factor and changing some elements in the FRM.

6.1.2.2 Results for changing scaling factor in error and derivative of error for 121 rules FRM

Real-time tests were conducted with increased scaling factor for the error and derivative of error. From Figure 6.1-6 it is observed that when the error is negative and increasing, no action is executed as desired. Whereas when the error is negative and decreasing, the desired immediate actions are executed. As the result of which error is always negative.

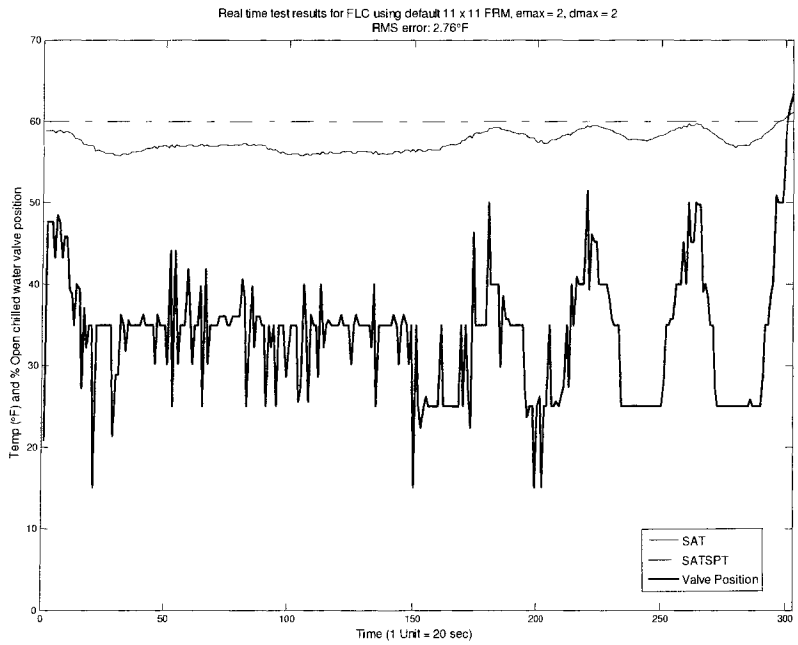

Figure 6.1-6: Case I - Real-time test results for FLC using default 121 rules FRM and SF of 2 for error and derivative of error

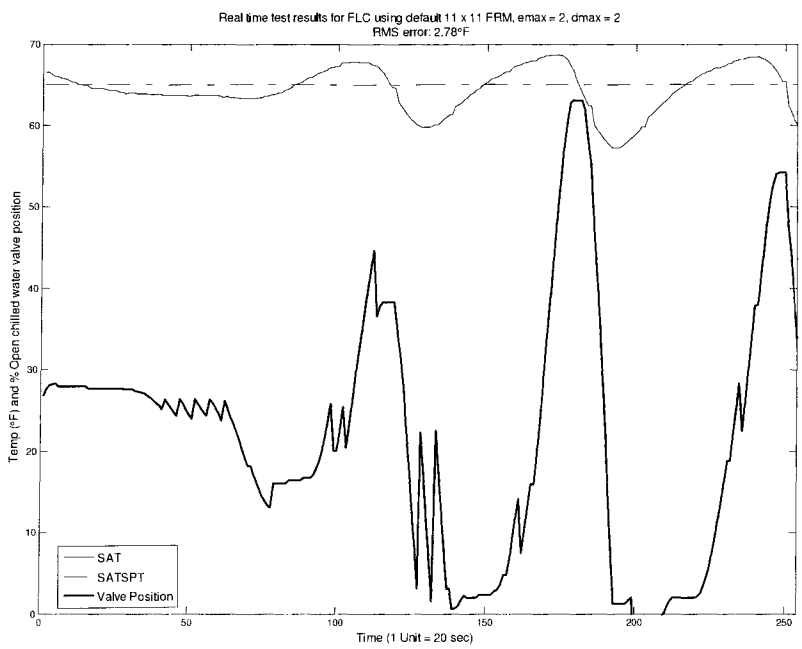

Figure 6.1-7: Case II - Real-time test results for FLC using default121 rules FRM and SF of 2 for error and derivative of error

More real-time tests were conducted with different scaling factors, but none resulted in acceptable FLC performance. Figure 6.1-7 shows the result when a scaling factor of 2 in error and derivative of error is used.

6.1.2.3 Results for changing few elements from the default 121 rules FRM

Some rules in the fuzzy rule matrix were changed and additional real-time tests were conducted along with a change in the SATSPT. Figure 6.1-8 shows the results obtained with the manually adjusted rule matrix. Again, the control is okay, but the results suggest that more work is needed to tune the FLC.

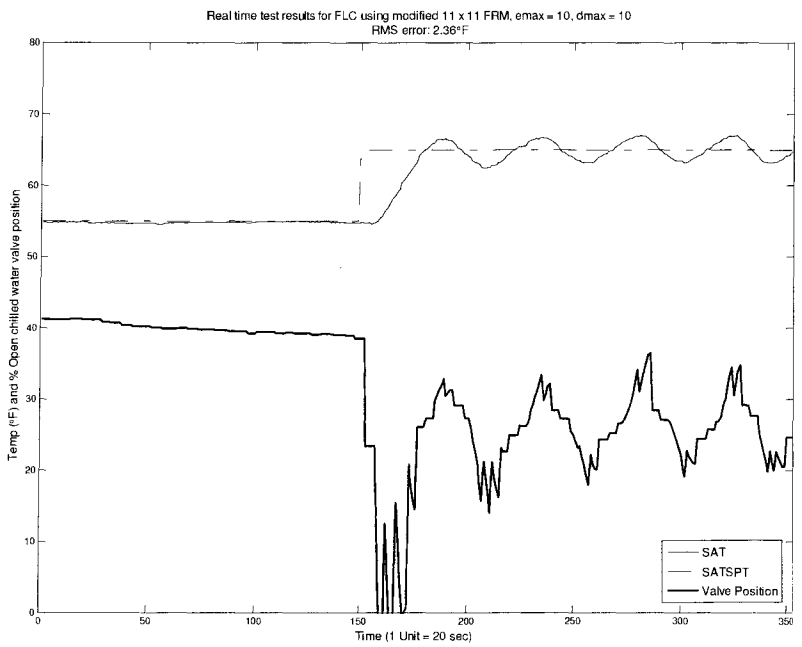

Figure 6.1-8: Real-time test results for FLC using modified 11 by 11 elements FRM

Figure 6.1-8 shows the results when the scaling factors of 10 in error and derivative of error. This configuration works very well for maintaining the temperature at the lower SATSPT, but when the SATSPT is changed, unacceptable oscillatory response is observed. Since some of the fuzzy rules are not correctly tuned oscillatory response is observed. Thus after conducting various experiments with different FLC configurations, namely altering the number of rules, changing the scaling factors, and changing values of the elements in the FRM, it was decided to develop an adaptive technique to take care of tuning the FLC.

6.1.3 Summary

Initially, experiments were conducted using a 7 by 7 default fuzzy rule matrix having 49 rules. These experiments demonstrated that the FLC was able to control the SAT by controlling the chilled water valve position. But the performance of this FLC was poor. Hence a larger 11 by 11 fuzzy rule matrix was developed which allowed 121 fuzzy rules and manual tuning methods for improving FLC performance were studied.

The larger 11 by 11 fuzzy rule matrix allowed use of more rules and hence finer control action. Two manual tuning methods studied were changing the scaling factor and changing the fuzzy rule matrix. In manual tuning, error, derivative of error and control signal scaling factors were changed. Whereas manual tuning of the FRM was done by changing selective elements in the FRM for which control action taken for error and derivative of error was not correct. FLC performance with the manually tuned 121 fuzzy rule matrices were better compared with manually changed scaling factors.

6.2 Neural Network Model Results for predicting SAT

The purpose of developing a neural network (NN) model is to use a computational model to study more cases for variations in the inputs and their effects on SAT. The offline NN models were developed and tested using Matlab's Neural Network Toolkit.

6.2.1 Offline Neural Network Models

For development of offline NN model previous experimental data was used. This data included the system behavior for various input conditions. In order to develop a good NN model, it is necessary that the training data set includes most of the operating conditions that system experiences.

6.2.1.1 Neural Network model for predicting SAT using 18 inputs

The initial NN model was studied with 18 inputs, namely:

1. Entering Water Temperature in Cooling Coil (°F)-(CHWC-EWT)
2. Leaving Water Temperature in Cooling Coil (°F)-(CHWC-LWT)
3. Mixed Water Temperature in Cooling Coil (°F)-(CHWC-MWT)
4. Entering Water Temperature in Heating Coil (°F)-(HHWC-EWT)

5. Leaving Water Temperature in Heating Coil (°F)-(HHWC-LWT)
6. Mixed Water Temperature in Heating Coil (°F)-(HHWC-MWT)
7. Chilled Water Valve Position (% Open)-(CHWC-VLV)
8. Chilled Water Pump Flow Rate (GPM)-(CHWP-GPM)
9. Exhaust Air Damper (% Open)-(EA-DMPR)
10. Mixed Air Temperature (°F)-(MAT)
11. Outside Air Flow Rate (CFM)-(OA-CFM)
12. Outside Air Damper (% Open)-(OA-DMPR)
13. Outside Air Temperature (°F)-(OA-TEMP)
14. Return Air Flow Rate (CFM)-(RA-CFM)
15. Return Air Damper (% Closed)-(RA-DMPR)
16. Return Air Temperature (°F)-(RA-TEMP)
17. Supply Air Temperature (°F)-(SA-CFM)
18. Supply Air Temperature Set Point (°F)-(SATSPT)

Several different NN architectures with varying number of neurons in the hidden layers and different training algorithms were studied. The best results were obtained for a single hidden layer using the *tansig* transfer function and with the output layer using the *purelin* transfer function. (See Appendix D for details). The hidden layer has 8 neurons. Results for the best configuration are shown in Figure 6.2-1. Figure 6.2-1 shows that the total RMS error is 0.70425 °F, thus this NN model was able to predict SAT well.

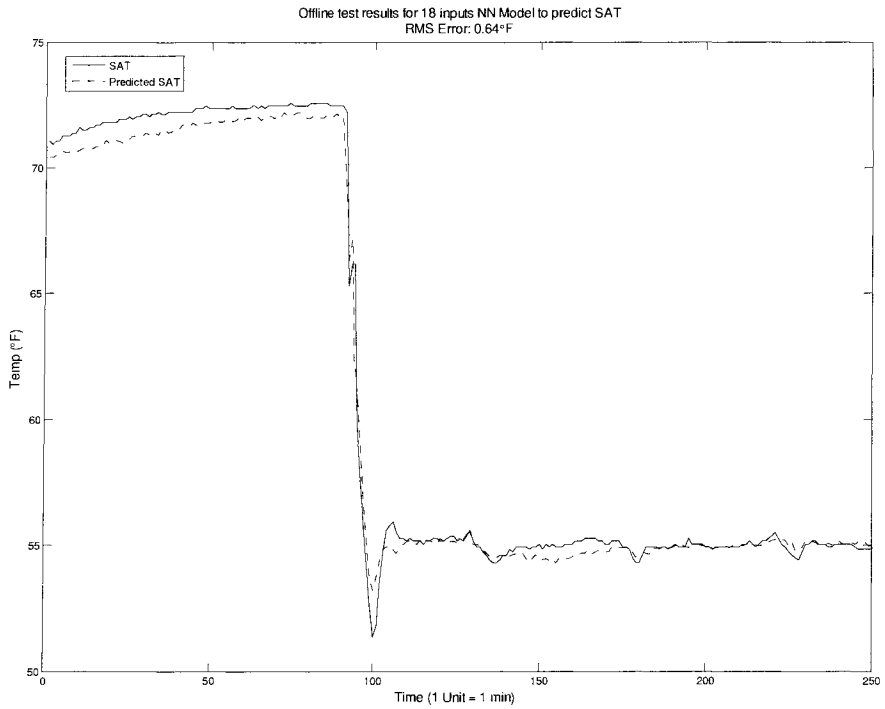

Figure 6.2-1: Offline test results for predicting SAT using NN model with 18 inputs

6.2.1.2 Neural Network model using derivative information

Another NN model using derivative information was studied with the following inputs:

1. Chilled Water Coil Entering Temperature - CHWC-EWT
2. Derivative of Chilled Water Entering Temperature - d(CHWC-EWT)
3. Mixed Air Temperature - MAT,
4. Derivative of Mixed Air Temperature - d(MAT),
5. Supply Air Flow Rate – SACFM
6. Derivative of Supply Air Flow Rate – d(SACFM)
7. Chilled Water Coil Valve Position – CHWC-VLV Position
8. Derivative of Chilled Water Coil Valve Position – d(CHWC-VLV Position)

Adding derivative inputs should help the NN model to map the time dependence of inputs to the output. Offline tests were conducted for above NN model and results are shown in Figure 6.2-2.

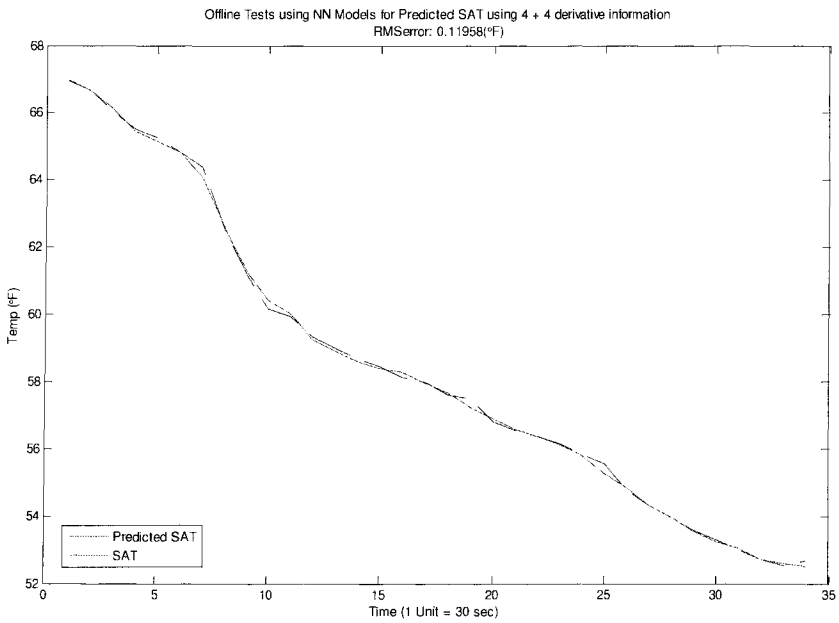

Figure 6.2-2: Offline test results for predicting SAT using NN model with 8 inputs having derivative information

Figure 6.2-2 shows that NN model using 8 inputs with derivative information has very good results but required more computation time. So another NN model was developed. While selecting inputs for this new NN model, basic knowledge of the heat and mass transfer involved in the cooling process and results obtained from sensitivity analysis were used (See Appendix E for details).

6.2.1.3 Neural Network model using 8 inputs

This NN model used the basic knowledge of heat and mass transfer for selection of different inputs. The 8 inputs selected for this NN model are:

1. Chilled Water Entering Temperature (CHWC-EWT);
2. Chilled Water Leaving Temperature (CHWC-LWT);
3. Chilled Water Mix Temperature (CHWC-MWT);
4. Chilled Water Flow Rate (CHWP-GPM)
5. Mixed Air Temperature (MAT);
6. Supply Air Flow Rate (SACFM);

7. Chilled Water Coil Valve Position (CHWC-VLV);
8. Supply Air Set Point (SATSPT);

Figure 6.2-3 shows the offline test results.

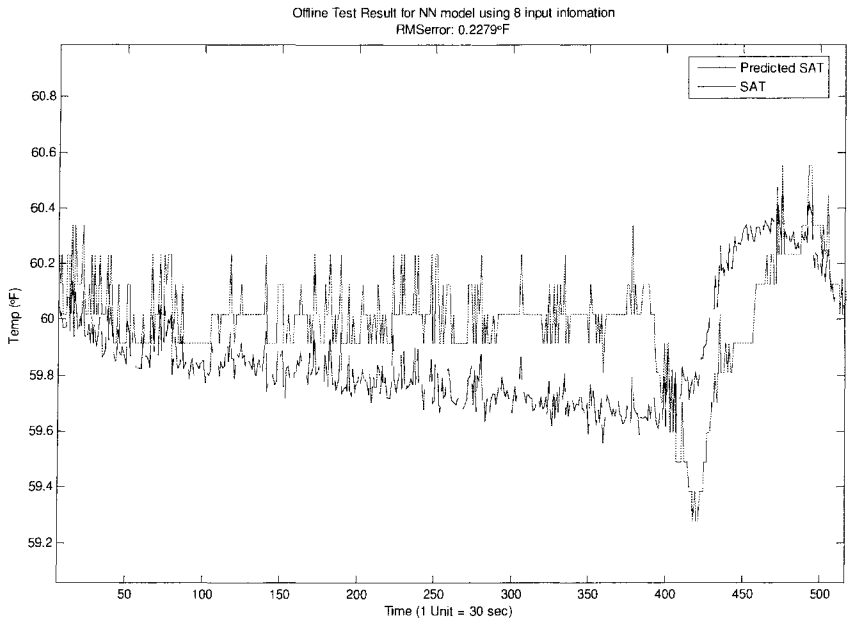

Figure 6.2-3: Offline test result for predicting SAT using NN model with 8 inputs for unseen data

It is observed that since the NN model was not trained with all conditions that system might experience; all of the time good results were not obtained. It is very difficult to have a training data set that will include all conditions that the system might experience. One method to solve this problem is to develop an adaptive NN model. In this method, the adaptive NN model will be updated periodically using the most current data which has been recorded.

6.2.1.4 Updating Trained Neural Network Model

Even though offline tests had acceptable performance for the above mentioned NN models, in real-time tests unacceptable errors between SAT and SAT_{PRED} were recorded. One of the main reasons for this behavior is the training data. In most of the cases studied, during real-time tests, the trained NN model was tested on unseen data that included values for variables

that were outside the training data set. Since NNs are highly nonlinear, for these unseen data inputs erratic results were obtained.

No information was found which discussed this problem and its solution in detail. A possible solution for this problem is to update the trained NN model in real time.
Two different approaches used to improve the NN model performance are:

1. Retraining the original NN model using old and new data (one big data set) or
2. Retraining the original NN model using new data only (using updating procedures)

6.2.1.5 Adaptive Neural Network model

For offline test of this adaptive NN model technique a good set of training data was created using the archives available from the ERS. Available data sets for all seasons and extreme limits were included in the training data set. Once a correct data set had been created, all of the 18 inputs discussed previous were considered as inputs to create a better NN model to predict SAT. Various combinations of NN architectures were studied to determine a good NN model.

The same test data set was used throughout the training phase to study different NN architectures. The best NN architecture used only 8 inputs (CHWC-EWT, CHWC-LWT, CHWC-MWT, MAT, CHWC-VLV, CHWP-GPM, SATSPT, SACFM), 8 nodes in the hidden layer and one output. Thus, it was decided to update the NN with just 8 input parameters and 8 nodes in the hidden layer.

This trained NN models were tested with completely unseen data. The retraining NN model showed a large error in the SAT_{PRED} whereas the adaptive NN model error in the SAT_{PRED} drops significantly and is within the acceptable limits (error is 0.5-1.0°F). The results are shown in Table 6.2-1 and Figure 6.2-4.

To study the NN model with retraining technique, 4 data sets were selected randomly from the archive of available data sets. Each data set was divided into 5 equal parts. For each case,

the NN model collects data for each part and predicts SAT, i.e. SAT_{PRED}. After prediction of SAT, the same NN model was retrained with the data for which it had just predicted SAT. This makes the NN model adapt to current data while retaining the information from the old data set. Thus, with this technique, valuable information accumulated in the original neural net is preserved and the NN is adapted to the new data. Computationally it is found that instead of retraining the neural net for the combined data set, i.e. old plus new data set, if the NN is trained on the new data set only; better results are obtained as indicated in Table 6.2-1 below. Also retraining on the combined set requires more computation time as compared with the retraining technique.

Table 6.2-1: Offline test results for retraining and adaptive trained neural network model

		Test Data1	Test Data2	Test Data3	Test Data4
# of Data Points used for retraining		2592	288	288	288
RMS error with old and new data sets (°F)	Part1	1.82	4.59	5.12	1.33
RMS error with new data set (°F)		0.25	0.16	0.26	0.33
RMS error with old and new data sets (°F)	Part2	1.34	0.43	5.10	1.25
RMS error with new data set (°F)		0.26	0.24	0.19	0.36
RMS error with old and new data sets (°F)	Part3	1.10	0.53	5.00	2.83
RMS error with new data set (°F)		0.25	0.17	0.29	0.33
RMS error with old and new data sets (°F)	Part4	1.86	0.87	0.54	0.99
RMS error with new data set (°F)		0.24	0.16	0.22	0.32
RMS error with old and new data sets (°F)	Part5	1.81	1.07	1.53	1.79
RMS error with new data set (°F)		0.25	0.18	0.21	0.33

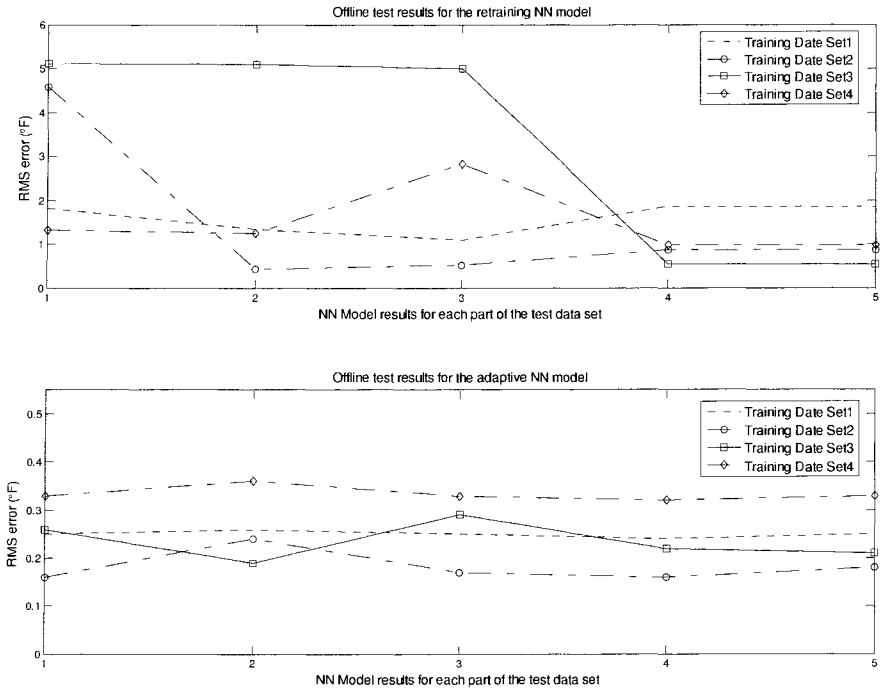

Figure 6.2-4: Offline test results for predicting SAT using retraining and adaptive NN model

Figure 6.2-5 and Figure 6.2-6 show the results for the retraining and the adaptive NN models respectively. The RMS errors are much smaller for the adaptive NN model calculated for entire data sets. Legends in Figure 6.2-5 and Figure 6.2-6 shows the RMS error for neural network model using complete data set, data set 1, data set 2, data set 3 and data set 4 respectively. From the Figure 6.2-5, for the offline test for retraining of NN model when the complete data set was used for training it had RMS error of 6.88°F, when only 1st data set was used it had RMS error of 8.28°F, for 2nd data set RMS error of 5.62°F, for 3rd data set RMS error of 5.63°F and for 4th data set RMS error of 4.96°F.

From the Figure 6.2-6, for offline test for the adaptive NN model when the complete data set was used for training it had RMS error of 1.32°F, when only 1st data set was used it had RMS error of 1.26°F, for 2nd data set RMS error of 1.07°F, for 3rd data set RMS error of 1.07°F and for 4th data set RMS error of 0.79°F.

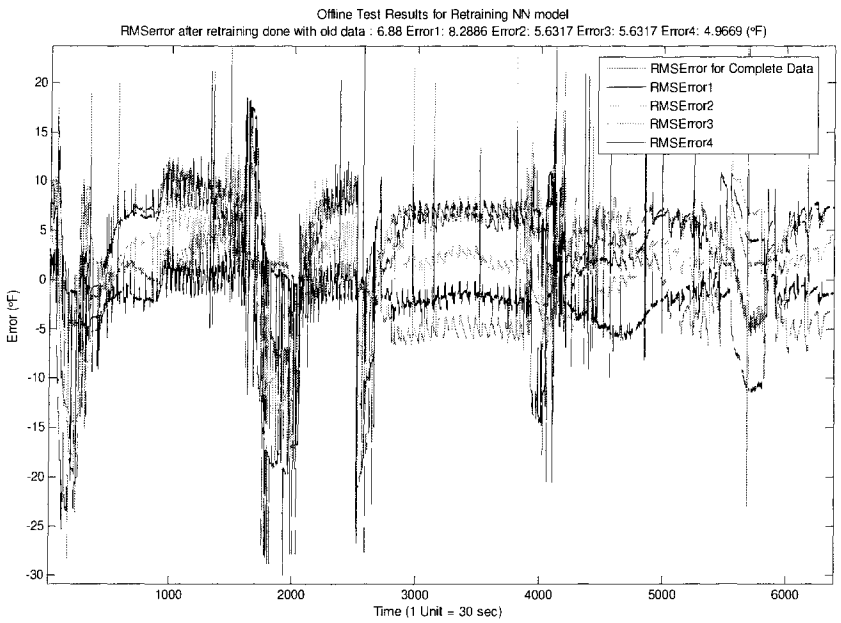

Figure 6.2-5: Offline test results for predicting SAT using retraining NN model

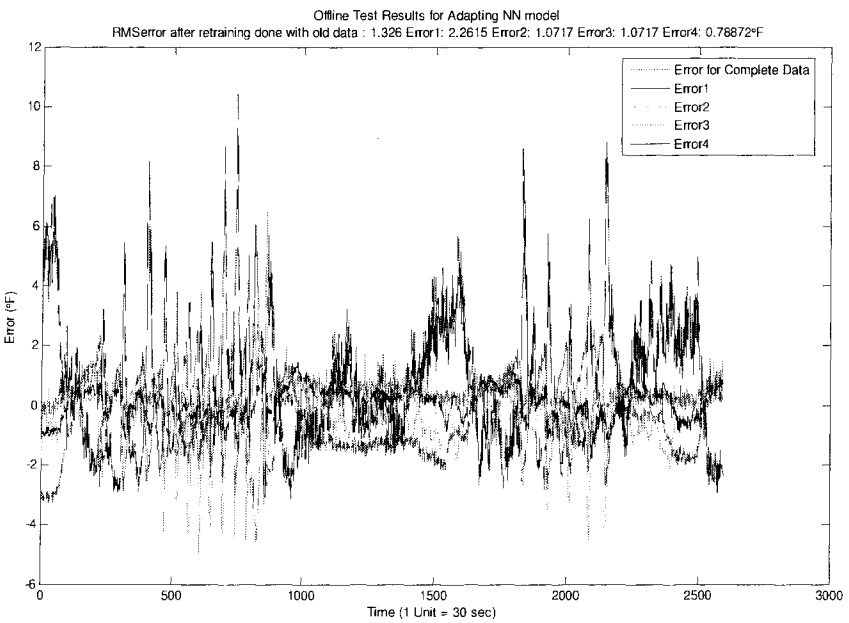

Figure 6.2-6: Offline test results for predicting SAT using adaptive NN model

6.2.1.6 Neural Network model output for erroneous and non erroneous data

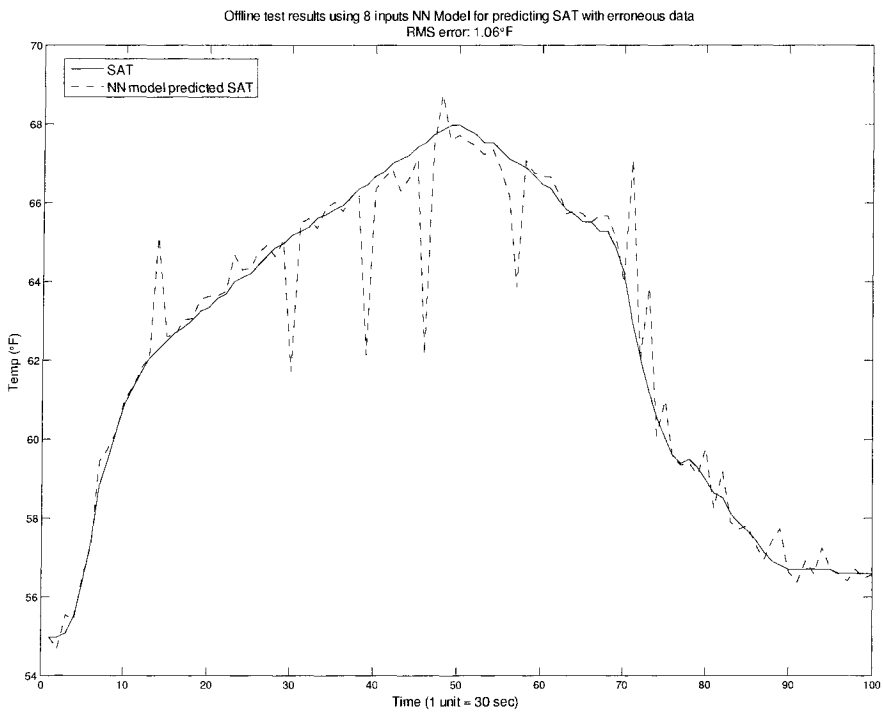

Figure 6.2-7: Offline test result for predicting SAT using NN model with 8 inputs having erroneous data

While conducting offline tests for the 8 input NN model, erroneous data was observed. Results for this test are shown in Figure 6.2-7. Figure 6.2-7 shows that sometimes large errors are observed in SAT_{PRED} because of erroneous data. This erroneous data problem was encountered during real-time tests. Instead of recording the correct value, *zero* or *-999* was recorded. This error is due to the communication interval between Metasys and Matlab. Metasys is updated every 20 seconds and if Matlab requests a value within those 20 seconds, either a zero value or a garbage value gets recorded. To solve this data reading problem during real-time tests, *zero* or *-999* values were replaced with the previous correct value.

Erroneous data has a significant effect on NN model performance. The NN model needs correct data for training and a trained network needs correct input data for calculating output values. The normalization process also gets affected due to this erroneous data. For example, if data for MAT is read as 'zero' instead of actual value (which should be within the range of

60 to 80 °F), then minimum value of MAT used for normalization is zero and the whole data is probably being normalized using zero as a minimum value.

In the original normalization process, minimum and maximum values were found from the data set used for training. So with the erroneous data, the training data was normalized incorrectly and hence further resulting error in SAT_{PRED}. So one method used to eliminate the error in the normalization is using predefined minimum and maximum values for each input. These predefined minimum and maximum values are calculated using previous available data. After correcting erroneous data, results for a NN model were improved and are shown in Figure 6.2-8.

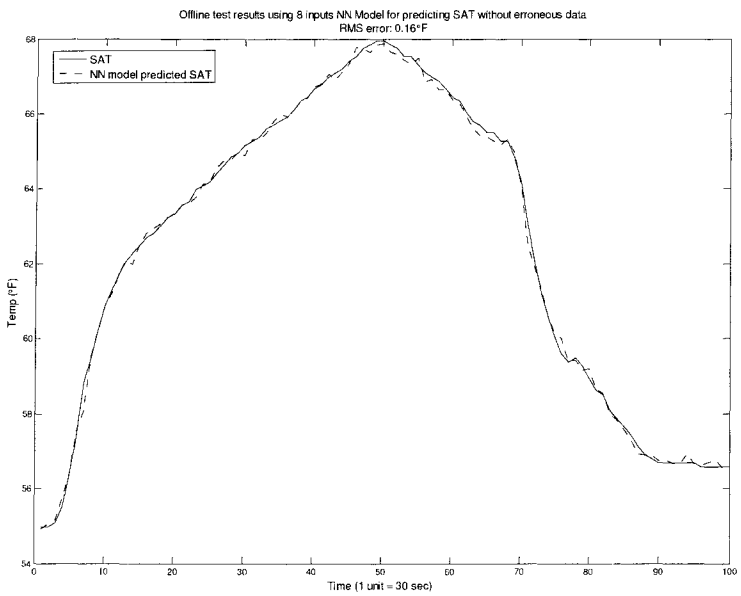

Figure 6.2-8: Offline test result for predicting SAT using NN model with 8 inputs without erroneous data

6.2.2 Real-time Test Results for NN Models and Adaptive Neural Network Model

6.2.2.1 Real-time results for NN model using 18 inputs

The real-time tests were conducted using the same NN model that was developed for the offline tests. The output from the NN model was recorded for every data point. Figure 6.2-9 shows the results for SAT, SAT_{PRED}, and SATSPT. It is observed that this NN model using

18 inputs has poor performance in real-time tests. Also it requires more computation time and fails to show output repeatability. Additional computation time causes problems in executing change in the chilled water valve position signal calculated by the FLC which results in delay for corrective action. Hence, another NN model with fewer inputs was studied.

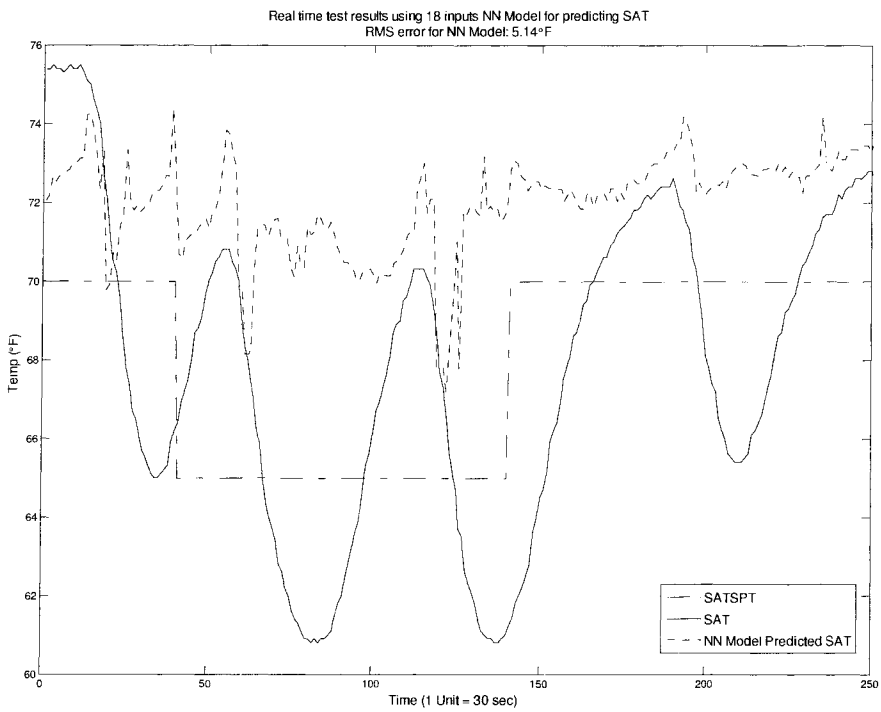

Figure 6.2-9: Real-time test results for predicting SAT using NN model with 18 inputs

6.2.2.2 Real-time results for NN model using derivative information

Since the offline results for this NN model were good, real-time tests were conducted using 8 inputs with derivative information. Figure 6.2-10.shows poor performance of this NN model.

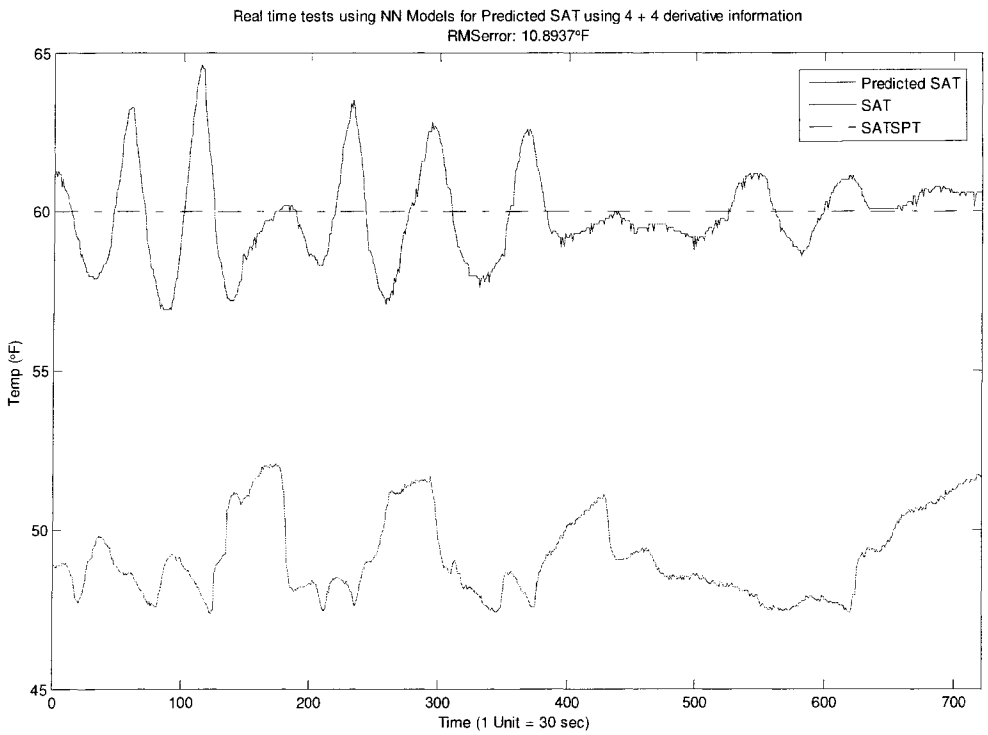

Figure 6.2-10: Real-time test results for predicting SAT using NN model using 8 inputs with derivative information

6.2.2.3 Real-time results for NN model using 8 inputs

Offline testing of NN model using 8 inputs for predicting SAT has shown better results hence real-time tests were conducted. During the real-time tests, larger RMS error was observed as shown in Figure 6.2-11 so this NN model was retrained using additional data points. After retraining, based on RMS error value, no improvement in NN model performance is observed as shown in Figure 6.2-12.

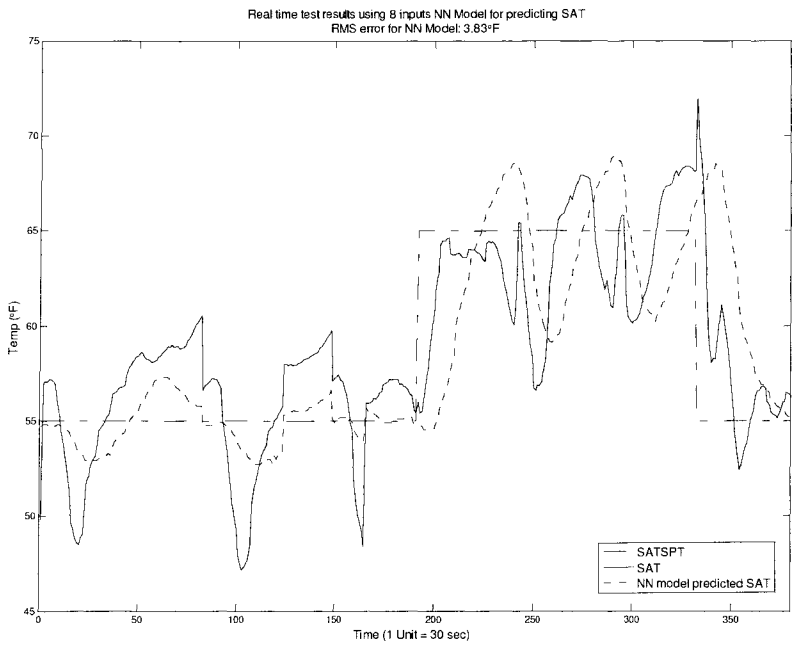

Figure 6.2-11: Case I - Real-time test results for predicting SAT using NN model using 8 inputs

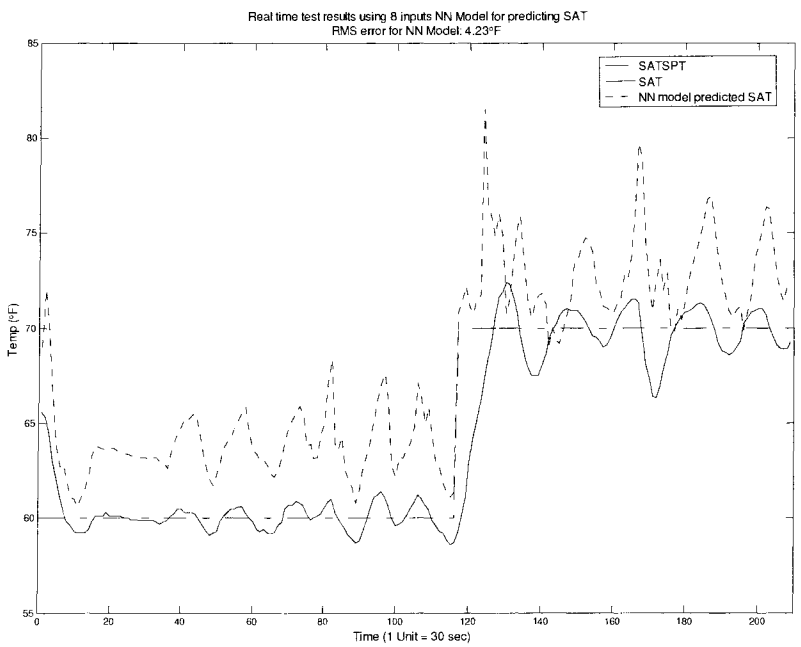

Figure 6.2-12: Case II - Real-time test results for predicting SAT using NN model using 8 inputs

6.2.2.4 Real-time results for adaptive NN model

Figure 6.2-13 shows results for real time tests using the adaptive NN model. Most of the time adaptive NN model was able to the predict SAT within 1.5°F with RMS error of 1.18°F. Additional real-time tests were conducting using this adaptive NN model and results are shown in Figure 6.2-14 along with the test conditions.

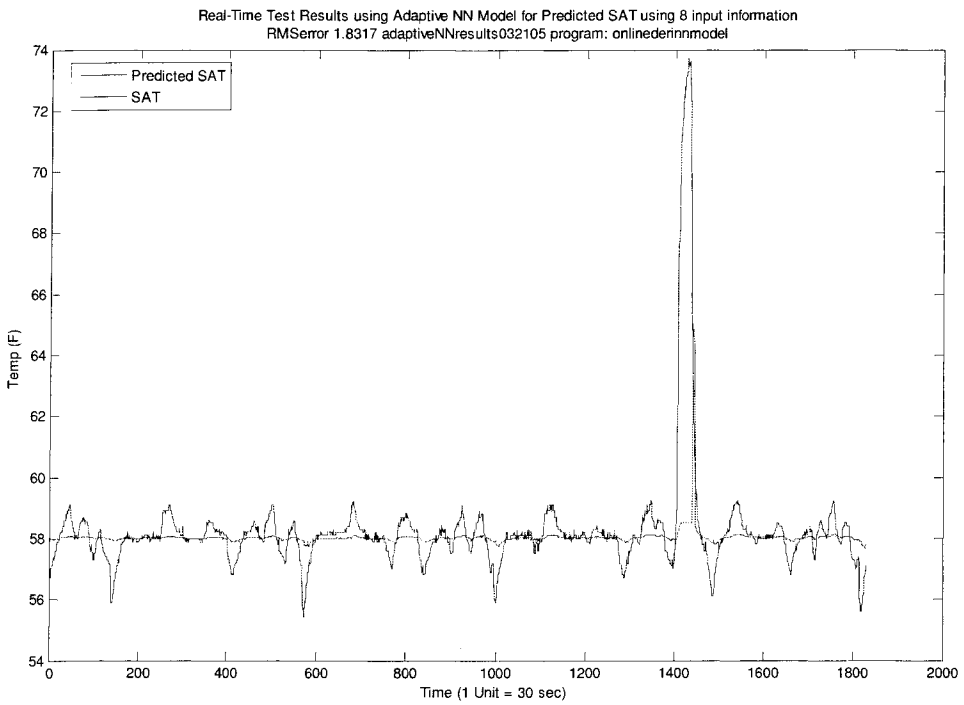

Figure 6.2-13: Real-time test results for predicting SAT using adaptive NN model

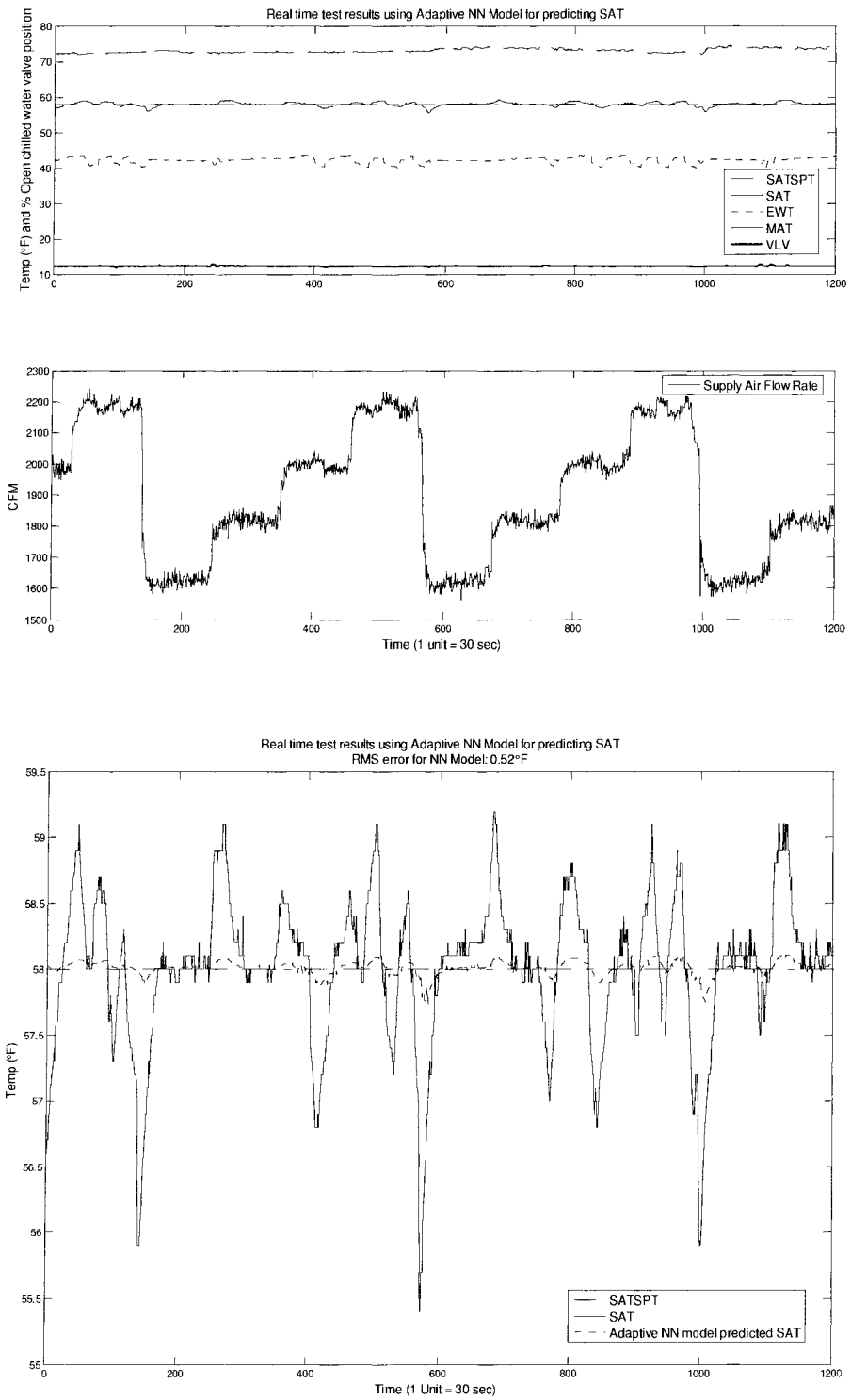

Figure 6.2-14: Real-time test results for predicting SAT using adaptive NN model

6.2.3 Summary

This section discussed various NN models that were developed. The purpose of developing an offline NN model was to study more cases for variations in the inputs and their effects on SAT. NN models with different numbers of inputs, namely 18, 8 and with derivative information were studied. An erroneous data problem was encountered during training the NN models. During real-time tests sometimes *zero* or *-999* values were recorded instead of correct input data. This problem was solved by replacing the erroneous data by using the previous correct reading and/or increasing the sampling periods to 20 seconds. Results for NN output using erroneous data and without erroneous data were studied. RMS error with erroneous data was 1.0809°F and without erroneous data was 0.1416°F.

Offline test results for a NN model using 8 inputs, along with derivative information, had RMS error of 0.14°F. But in real time experiments, poor performance of NN model was observed with RMS error of 10.89°F. Offline test results for a NN model using 8 inputs had RMS error of 0.23°F. For real time experiments the RMS error was 4.34°F.

For all the NN models studied, higher RMS errors were observed, so two updating techniques for NN models were studied. In first technique the NN model was retrained using old and new data sets and in the second NN model was retrained using only the new data set. Offline results show that an RMS error less than 1°F was observed for the second adaptive NN model. Also real-time test had 1.8 °F RMS errors for second adaptive NN model.

6.3 GRNN Model Results for predicting SAT

Another cooling coil model was developed using the GRNN method. Generalized Regression Neural Network (GRNN) has been proposed as an alternative to statistical regression equations and conventional artificial neural-networks. GRNN are memory-based feedforward networks based on the estimation of probability density functions. Offline and real-time test results are discussed below.

6.3.1 Offline Test

To predicted the supply air temperature (SAT) for the next time step, an offline GRNN model was developed using 8 inputs, namely;

1. Chilled Water Entering Temperature (CHWC-EWT);
2. Chilled Water Leaving Temperature (CHWC-LWT);
3. Chilled Water Mix Temperature (CHWC-MWT);
4. Chilled Water Flow Rate (CHWP-GPM)
5. Mixed Air Temperature (CHWC-MAT);
6. Supply Air Flow Rate (SACFM);
7. Chilled Water Coil Valve Position (CHWC-VLV);
8. Supply Air Set Point (SATSPT);

The offline GRNN was developed and tested using Matlab's Neural Network Toolkit. This GRNN model (GRNN model I) used the same training data that was used for adaptive NN models. Offline testing of this trained GRNN model was done on the unseen data. Results show that this GRNN model was able to predict SAT with higher RMS error. So, another GRNN model (GRNN model II) using past output information for SAT was studied. This GRNN model used 18 inputs, namely the 8 inputs used in GRNN Model I and 10 time steps of past information for the SAT. Results for both of these models are shown in Figure 6.3-1 and Table 6.3-1.

Table 6.3-1: Results for Offline Tests using GRNN Models for Predicted SAT

	Model Description	RMS error (°F)
GRNN model I	8 inputs	1.64
GRNN model II	8 inputs plus 10 time steps past information for the output	0.10

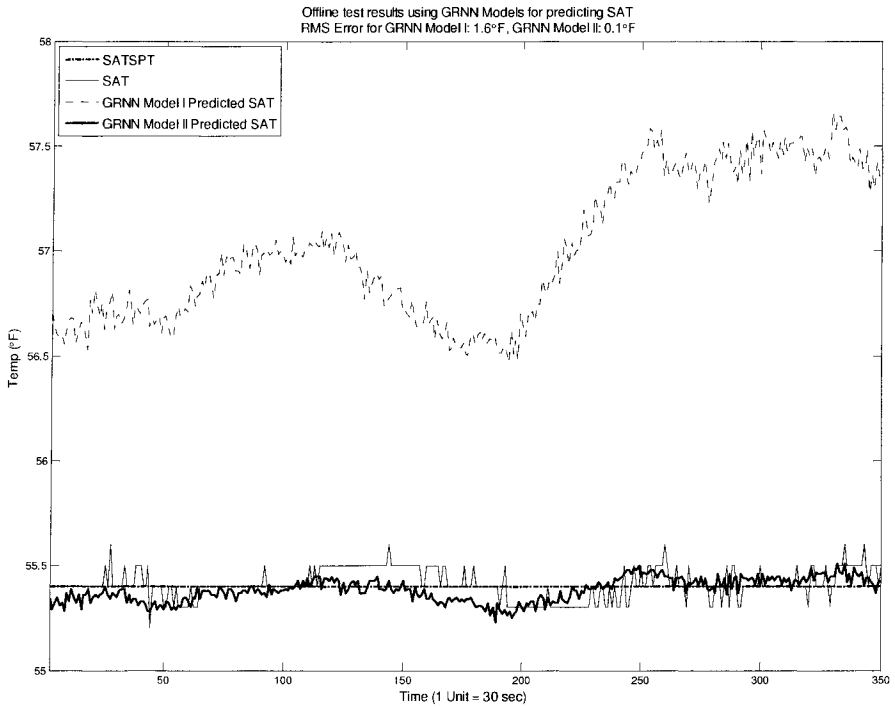

Figure 6.3-1: Offline test results for predicting SAT using GRNN models I and II

In addition to the above two models, another GRNN model (GRNN model III) was studied which used derivative information for the previous outputs in addition to the 8 inputs used in GRNN model I. Offline test results show that GRNN model II had the least RMS error among the three models studied. Table 6.3-2 and Figure 6.3-2 shows results for offline test.

Table 6.3-2: Offline tests results for predicting SAT using GRNN Models

	GRNN Model Type	RMS error (°F)
GRNN Model I	8 inputs	16.08
GRNN model II	8 inputs with 10 time steps of past information for the output	0.97
GRNN model III	8 inputs with 10 time steps of past derivative information for the output	2.91

Offline test results shows that GRNN model II and III are better and hence were considered for the real-time studies.

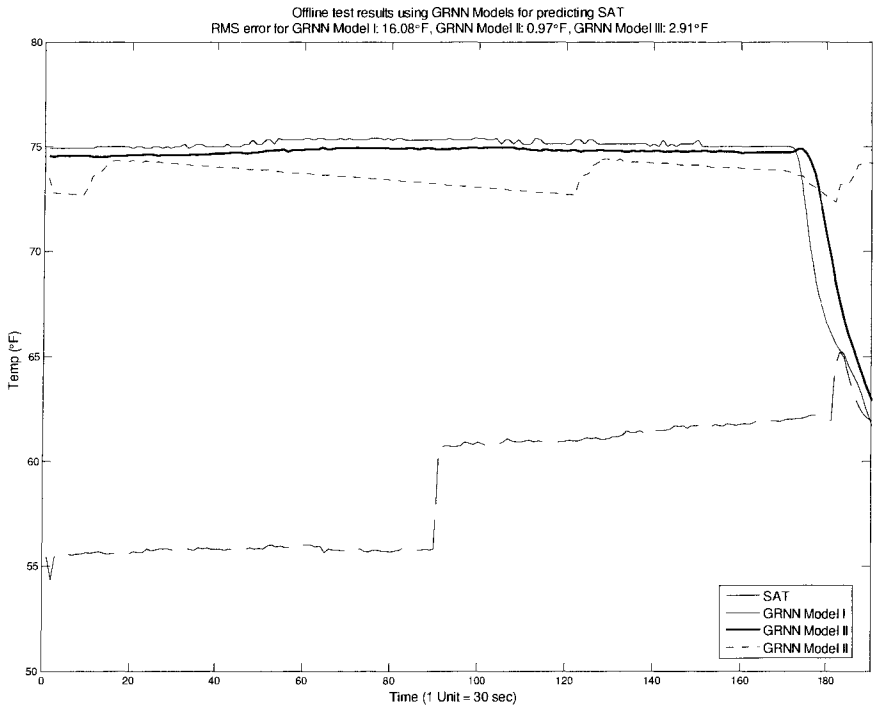

Figure 6.3-2: Offline test results for predicting SAT using GRNN model I, II and III

6.3.2 Real-time Test

Real-time tests were conducted with GRNN models II and III. The GRNN models II and III, which were trained offline, were recalled for real-time tests. Using the GRNN models, predicted SAT values were calculated and recorded for each time step. For real-time tests, SATSPT was changed from 60 to 70°F every 45 minutes. Figure 6.3-3 shows the variations in the input variables and Figure 6.3-4 shows the results for SAT and predicted SAT using GRNN models. The RMS errors for each GRNN model are shown in Table 6.3-3. Results show that GRNN models II and III both are able to predict SAT with lower error. Due to less computational time and better performance, GRNN model II was further tested for constant SATSPT with variation in the supply air flow rate from 1600 – 2200 cfm in steps of 200 cfm every hour. For real-time tests GRNN model II was able to predict SAT with low RMS error.

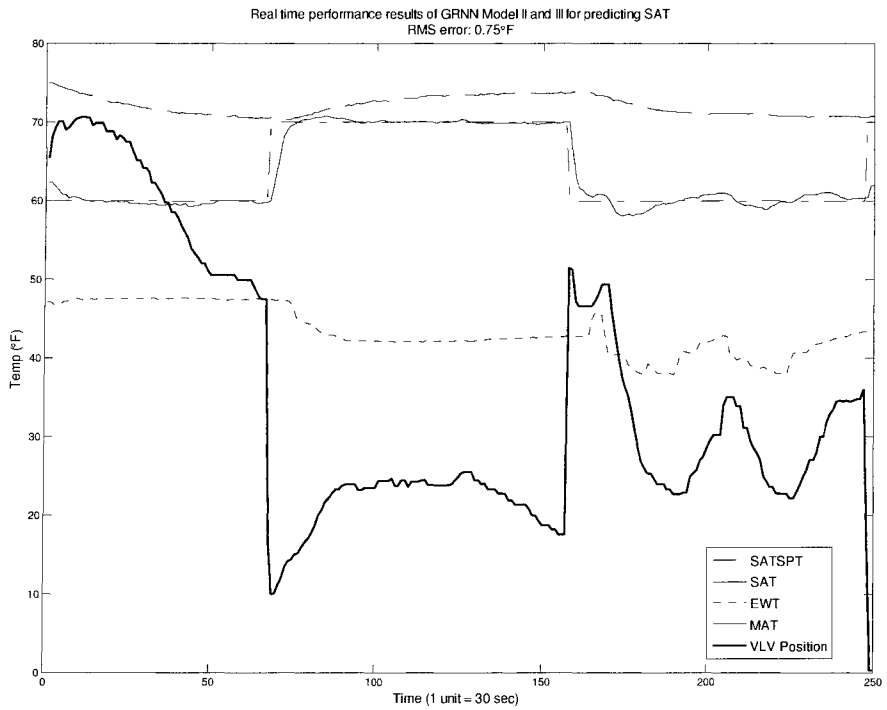

Figure 6.3-3: Variation in the inputs for real-time testing of GRNN model II and III

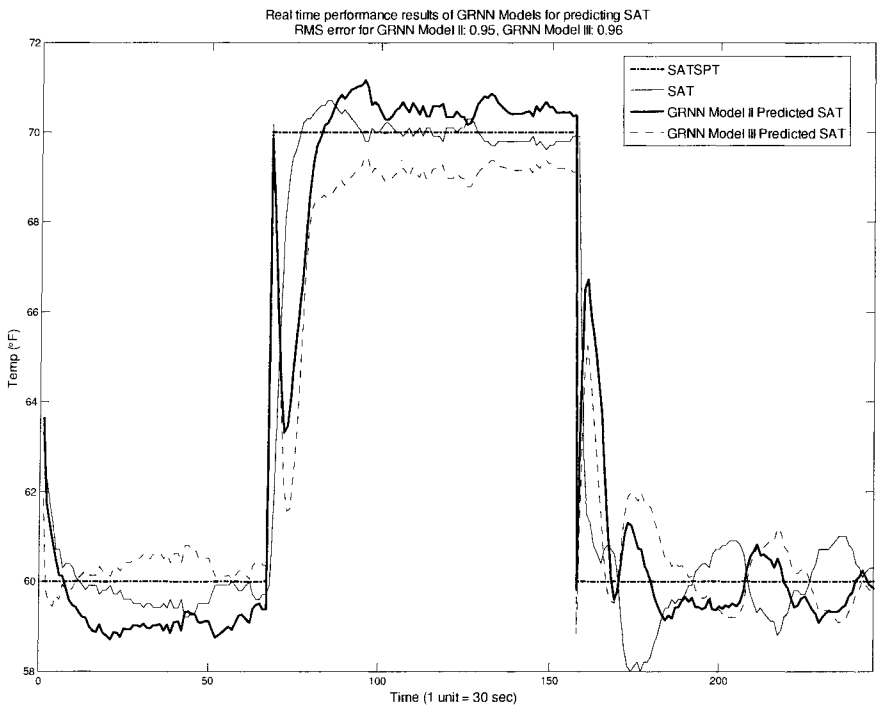

Figure 6.3-4: Real-time test results for predicting SAT using GRNN models II and III

Table 6.3-3: RMS error for predicting SAT using GRNN models II and III

	GRNN Model Type	RMS error (°F)
GRNN model II	8 inputs with 10 time steps of past information	0.95
GRNN model III	8 inputs with 10 time steps of past derivative information	0.96

6.3.3 Summary

The development of offline and real-time GRNN models to predict SAT for the next time step was discussed in this section. All the GRNN models were developed and tested using Matlab's Neural Network Toolbox. Three different GRNN models were studied. The first model used 8 inputs, the second model used 8 inputs along with 10 time steps of past information for the output, and the third model used 8 inputs along with 10 time steps of past derivative information for the outputs. It was observed that for the offline tests GRNN model I had poor performance with RMS error of 1.64°F for the first case and 16.08°F for the second case. GRNN model II was the best model with RMS errors of 0.10°F and 0.97°F respectively for the two cases studied. GRNN model III had an RMS error of 2.91°F for the second case. GRNN models II and III were tested in real-time. GRNN model II had an RMS error of 0.95°F and GRNN model III had an RMS error of 0.96°F. Overall, the performance of GRNN model II was better than the other GRNN models studied.

6.4 Lump Capacitance Models Results for predicting SAT

For offline testing of different models, open loop data was used. For the open loop test of the system, no control action was executed on the chilled water valve position. The chilled water valve position was incremented from 0 – 100% open position in steps of 10% and then decremented from 100 – 0% open position in steps of 10% after holding it constant for 45 minutes in every position. CHWC-EWT, CHWC-LWT, EAT, and chilled water valve position were recorded. For the LCM development, all these recorded values were used to

predict SAT in the offline test of each model. All the LCMs were evaluated based on RMS error value calculated as:

$$\text{RMS error} = \sqrt{\sum_{i=1}^{n} (\text{SAT} - \text{SAT}_{\text{PRED}})^2}; \qquad \ldots 6.4\text{-}1$$

where,

n = number of data points

6.4.1 Offline Test

Offline tests were conducted using fours different lump capacitance models as discussed in section 4.5. As mentioned in the above section, open loop test data was used for this study. SAT predicted by each LCM is shown in Figure 6.4-1.

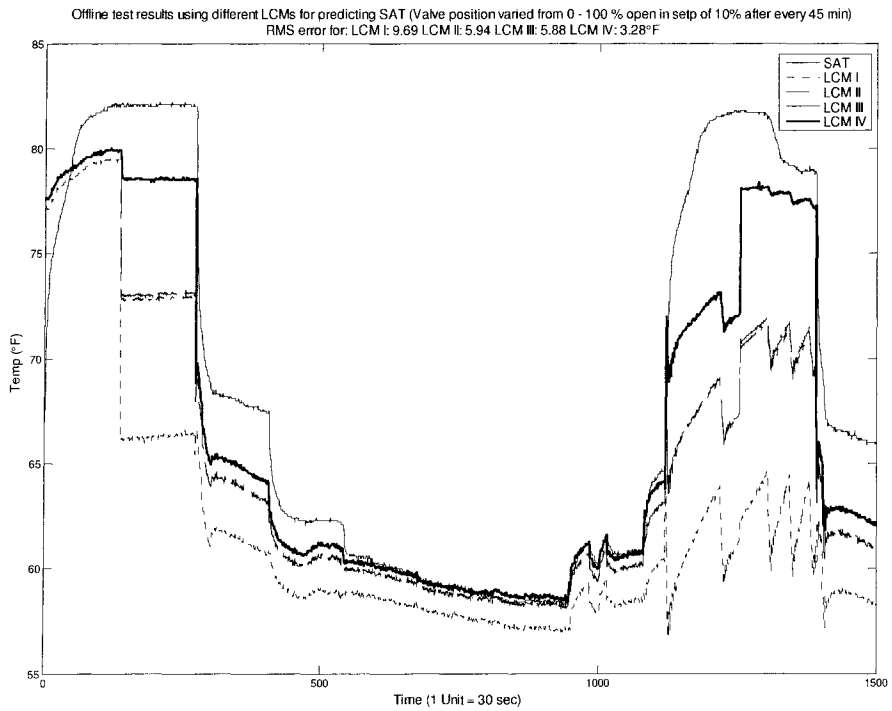

Figure 6.4-1: Difference between SAT_{PRED} and SAT for different LCMs

Table 6.4-1: RMS error for offline test in predicting SAT using different LCMs

	RMS error (°F)
LCM I	9.69
LCM II	5.94
LCM III	5.88
LCM IV	3.28

From Figure 6.4-1 and Table 6.4-1, LCM IV has minimum RMS error. For this model, the GA technique was used to determine values for VLVca and VLVcb along with UA, AFc, Tcw, Tca. LCM III was next best. For LCM II and LCM III, based on RMS error values, no significant changes in the results were observed even though LCM III accounted for data reading errors. LCM I have maximum RMS error since it did not account for the correction in the chilled water flow through the valve.

6.4.2 Real-time Test

The purpose of this real-time test was to verify LCM performance. As mentioned before, the purpose of the model is to use it in the development of a control system. Real-time test results may also suggest if some modifications in the model are required. Real-time validation is accomplished by recording SAT for the step change in the supply air set point temperature and recording calculated SAT_{PRED} values along with CHWC-EWT, CHWC-LWT, CHWC-MWT, EAT, CHWC-GPM and chilled water valve position.

Based on the offline test results, real-time testing was done using LCM IV. A step change in supply air set point temperature was commanded. SAT_{PRED} was calculated for each time step using LCM IV. Figure 6.4-2 shows the CHWC-EWT, MAT, SAT, SATSPT and VLV position. Real-time test results are shown in Figure 6.4-3 and Table 6.4-2. LCM IV was able to predict SAT within 2°F for most of the test time.

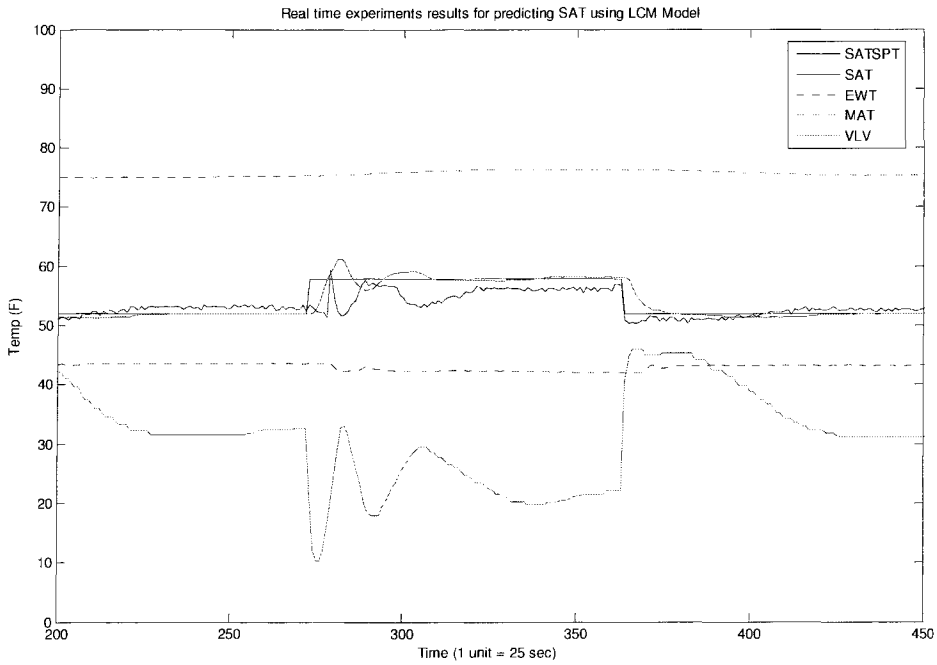

Figure 6.4-2: Variation in the inputs for real-time test using LCM IV

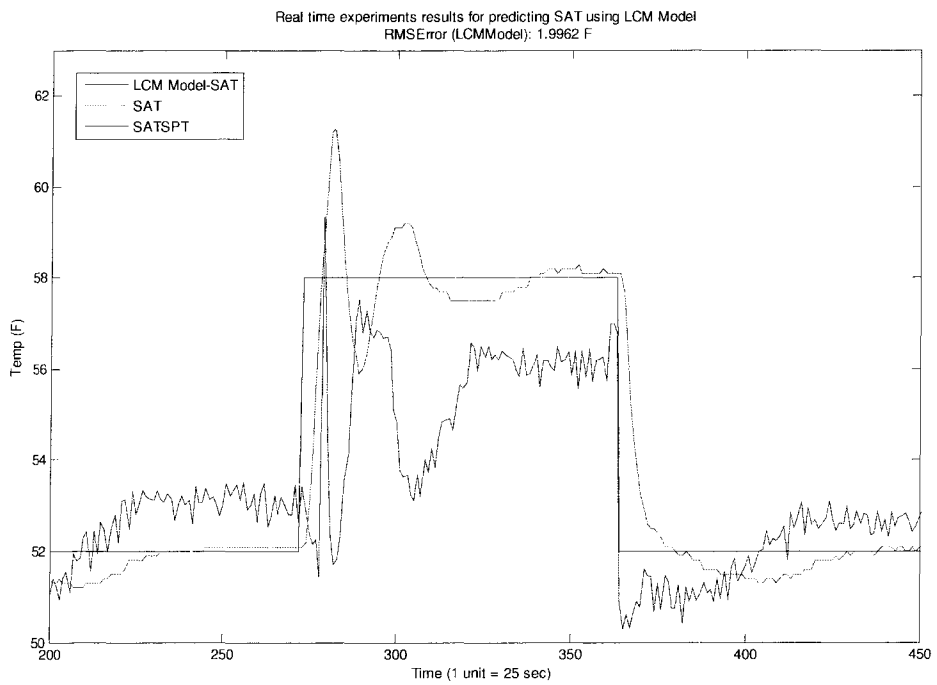

Figure 6.4-3: Predicted SAT using LCM IV and actual SAT for real time test

Table 6.4-2: RMS error for real time study using LCM IV

Results obtained for Real-time Study	
	RMS error (°F)
LCM IV	1.99

6.4.3 Summary

A simple lumped capacitance model was developed based on fundamental heat and mass transfer principles. The real-time test results showed that the lumped capacitance model can predict the dynamic behavior of a cooling coil without requiring geometric specifications of the cooling coil. LCM IV was able to predict SAT with an RMS error of 2°F over the entire range for the study. The performance of the model was improved when chilled water valve characteristics were considered. A Genetic Algorithm technique was able to find values for the constants in the model. In other studies, the model was applied in real-time tests to determine a better control system.

6.5 Real-time Test Result Comparison for Predicting SAT using Different Models

As discussed before, different models were developed for predicting the supply air temperature (SAT). To compare the real-time performance of the different models developed for predicting SAT, tests were conducted under the same conditions. Figure 6.5-1 shows the variation in the inputs for the real-time test. Figure 6.5-2 shows the variation in predicted SAT for GRNN model I and GRNN model II. Table 6.5-1 show the RMS error in predicting SAT for the different models studied.

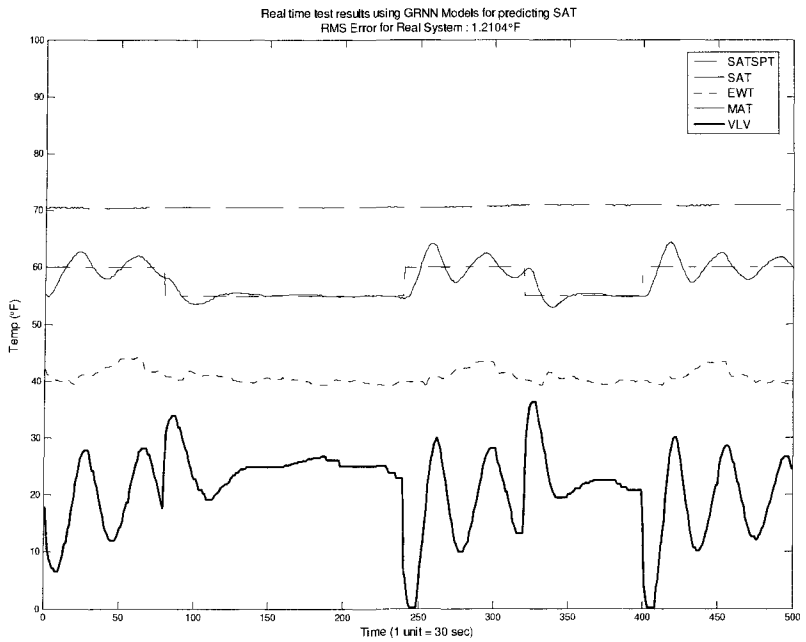

Figure 6.5-1: Variation in the inputs for SAT predicting models study

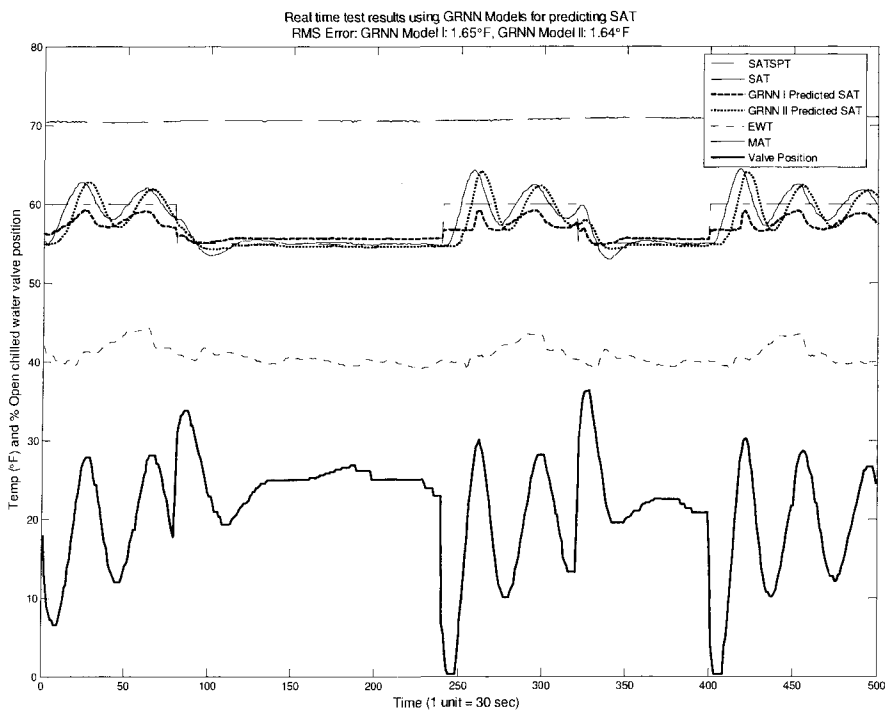

Figure 6.5-2: Real-time test results for predicted SAT using GRNN I and GRNN II

Table 6.5-1: RMS error for real-time test in predicting SAT using different models

Real-time study results for predicted SAT using different models	
Model	RMS error (°F)
LCM IV	5.30
NN	1.86
GRNN I	1.65
GRNN II	1.64

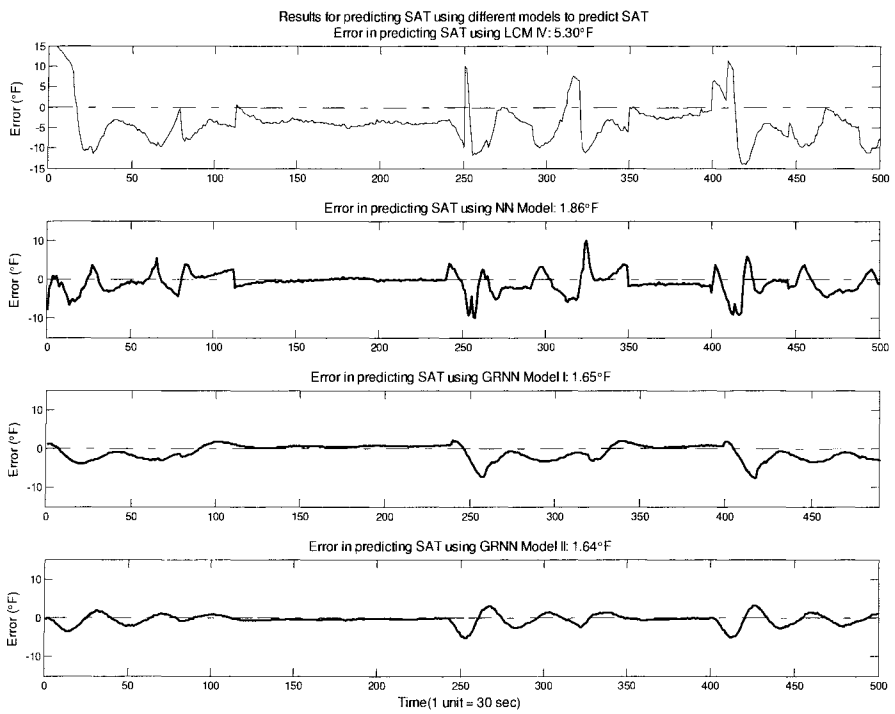

Figure 6.5-3: Errors for real-time tests in predicting SAT using different models

Table 6.5-1 and Figure 6.5-3 show that the best results were obtained for GRNN model II and the worst for LCM IV of all the four models studied. The LCM IV predicted SAT lower than the actual. Results from the LCM IV and NN models were oscillatory compared to the GRNN models.

Several different models developed for predicting SAT are discussed in Chapter 4. Based on the offline test results, only two models, the LCM IV and GRNN II models were studied further in real-time. Real-time tests were conducted and results are shown in Figure 6.5-4, Figure 6.5-5 and Figure 6.5-6. The GRNN model II was able to predict the SAT with slightly lower RMS error compared with the LCM IV.

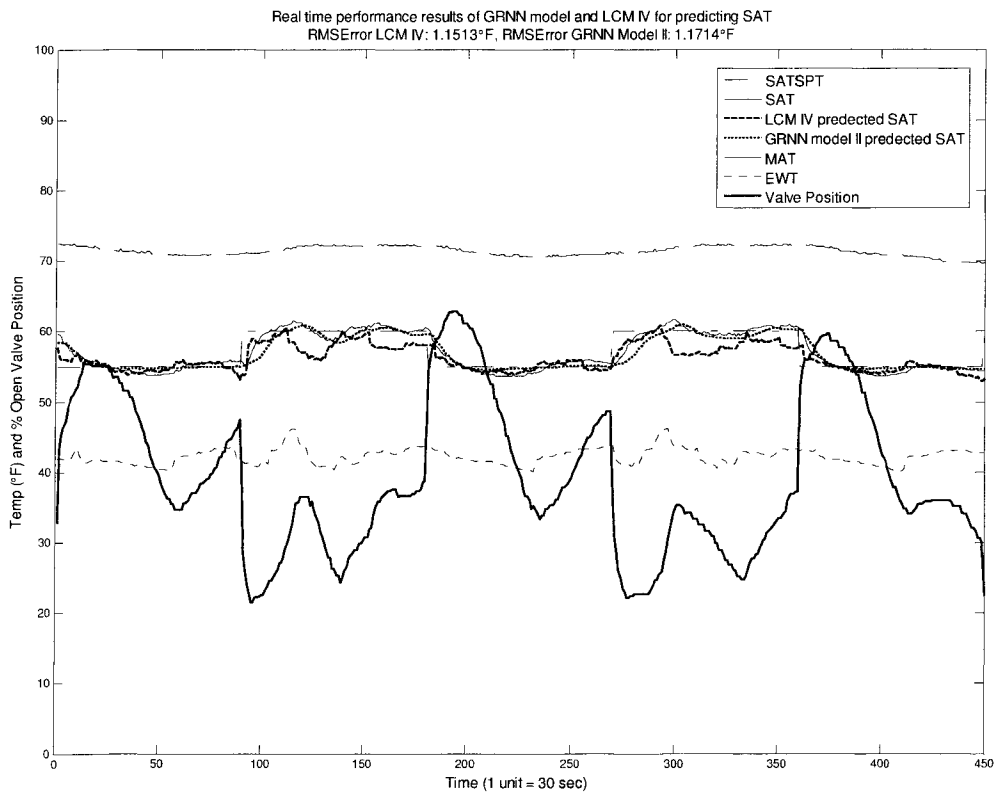

Figure 6.5-4: Real-time study results for predicting SAT using LCM IV and GRNN model II

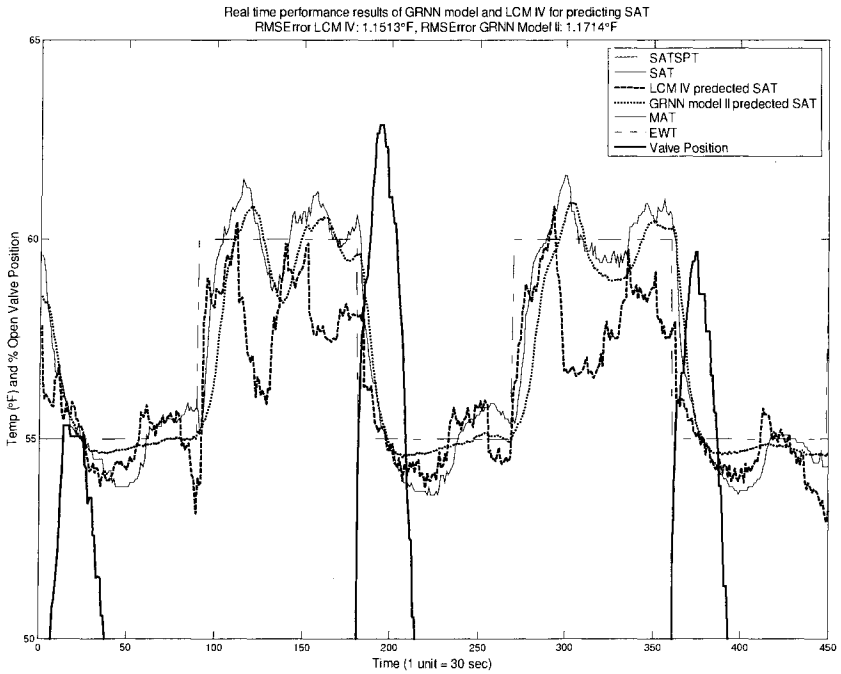

Figure 6.5-5: Zoom-In Real time study results for predicting SAT using LCM IV and GRNN model II

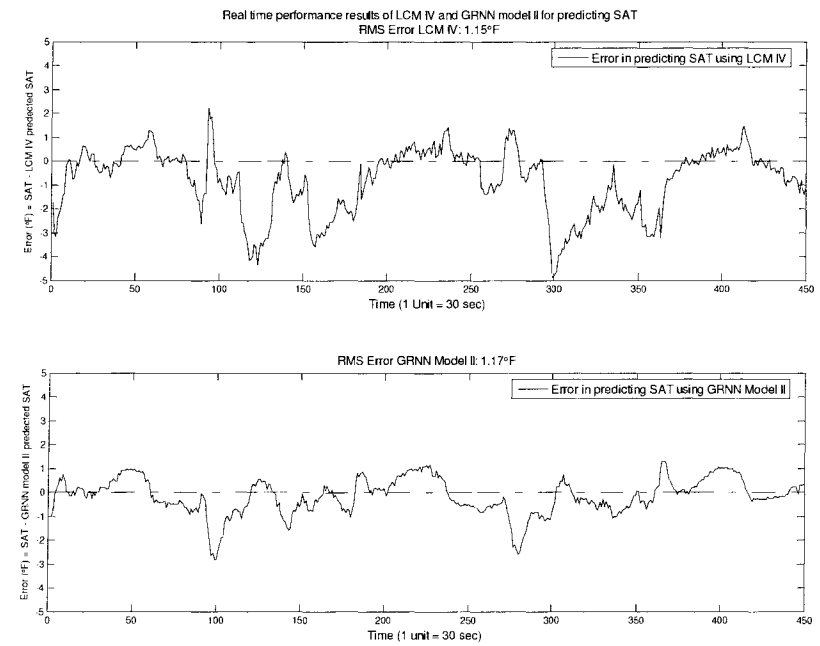

Figure 6.5-6: Real-time study results for RMS error in predicting SAT using LCM IV and GRNN model II

6.6 Summary

Different models, namely, Lump Capacitance models, GRNN models and Neural Networks models were studied offline and in real time for predicting SAT. For the offline study, the best results were obtained with GRNN models II and III which had RMS errors less than 1.5°F. Further all these models were studied in real time. For initial real-time tests, GRNN II and III, NN models had better results with RMS errors within 1.8°F. But LCM IV had unacceptable results. More real-time tests were conducted using GRNN model II and LCM IV. The performance of these models was good and RMS error was less than 1.2°F for both the models.

Chapter 7 Results for Adaptive Fuzzy Logic Controller (AFLC)

7.1 AFLC using Genetic Algorithms (AFLC-GA)

As discussed in the adaptive FLC section, Genetic Algorithms (GAs) were used to study different Fuzzy Rule Matrices (FRMs) generated by the following methods:

- FRM obtained by random numbers generated between 1 and 11
- Changing the Human FRM by random numbers generated amongst –1, 0, and 1 without changing specific elements
- Changing the Human FRM by random numbers generated amongst –1, 0, and 1 without changing specific elements in the outer loop of the Human FRM

For adapting FRMs using the GA technique, the linguistic representation of fuzzy membership functions (FMFs) for control actions are changed to a numeric representation as shown in Table 7.1-1. Last row in Table 7.1-1 shows the changes in the chilled water valve position for the numerical representation given in the second row.

Table 7.1-1: Numeric representations of control action (u) for each FMF

Control Action (u)											
Fuzzy Membership Function	*NL*	*NM*	*NS*	*NVS*	*NT*	*ZE*	*PT*	*PVS*	*PS*	*PM*	*PL*
Numerical Representation	1	2	3	4	5	6	7	8	9	10	11
Control Action (change in chilled water valve position)	-5	-4	-3	-2	-1	0	1	2	3	4	5

7.1.1 FRM obtained by Random Numbers Generated between 1 through 11

In this study, FRMs are initialized by generating random numbers between 1 through 11 for different possible fuzzy rule control actions. This type of initialization method provides much different FRMs than those generated based on human intuition.

7.1.1.1 Offline Test

An example FRM generated for the offline study is shown in Table 7.1-2.

Table 7.1-2: Best FRM generated by random numbers between 1 through 11 for offline test

Error	*Derivative (d)*										
(e)	*NL*	*NM*	*NS*	*NVS*	*NT*	*ZE*	*PT*	*PVS*	*PS*	*PM*	*PL*
NL	3	10	7	2	11	3	6	3	7	8	8
NM	8	7	3	1	5	6	2	1	6	9	7
NS	8	8	11	5	1	8	8	3	2	5	6
NVS	7	5	6	7	5	4	9	10	1	9	3
NT	2	1	5	10	11	6	5	7	1	1	1
ZE	11	8	11	8	11	9	2	3	4	8	11
PT	3	9	1	4	8	9	9	2	3	10	2
PVS	4	2	2	9	4	6	8	11	6	8	1
PS	3	9	3	2	11	10	8	9	5	4	9
PM	10	10	5	5	8	3	3	2	1	11	4
PL	7	11	7	2	7	4	4	10	1	9	3

Each FRM was generated from the random numbers between 1 through 11 using the GA technique. FRMs equal to the population size were generated. Using required parameters from the previous data except SAT, control actions were generated using FRMs for every data point. GRNN Model II was used to predict SAT for every control action. The RMS error for all the data points and for every FRM was calculated. Based on RMS error values, FRMs are sorted. The FRM having minimum RMS error was at the top and the FRM with maximum RMS error at the bottom. The two best FRMs, having the least RMS error, are reproduced to obtain two other FRMs. Reproduction was done by crossover and mutation

process (See Appendix F and Goldberg 1987 for more information) for predefined number of generations or until a termination criteria was reached. The termination criteria used in this project was zero RMS error. The number of generations defined the number of cycles for which reproduction process will be continued. For offline tests, the best generated FRM, shown in Table 7.1-2, having minimum RMS error resulted in poor FLC performance. Figure 7.1-1 shows results for the offline test.

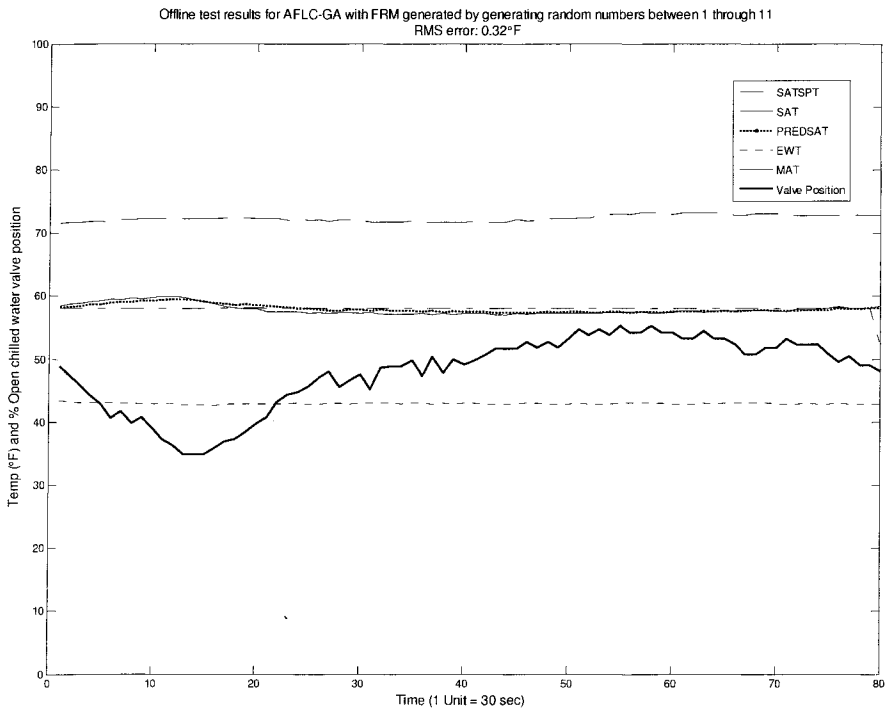

Figure 7.1-1: Offline test results for the best generated FRM from the random numbers between 1 through 11

7.1.1.2 Real-time Test

For the real-time test, the same technique used for the offline test was used to generate FRMs. Table 7.1-3 shows that the FRM developed was not logical. For example, when the fuzzy membership in error is ZE and the fuzzy membership in derivative of error is ZE, there should have been *zero* control action.

For real-time tests, after every 60 minutes SATSPT was cyclically changed between 55°F and 65°F. After every 120 minutes, a new FRM was generated. This new FRM was generated during the real time study by using the GA technique. In this study FRMs equal to the population size are generated, sorted based on the RMS error after calculating the RMS error for each. The crossover and mutation operations are performed for the predefined number of generations on the two FRMs having minimum RMS error to obtain the best FRM. This best FRM had the least RMS error of all the FRMs for the data used in the study. This best FRM is then used for real-time test. After completion of one cycle, again a new best FRM was obtained as mentioned in the above process and used for real-time study.

Table 7.1-3: Best FRM generated by random numbers between 1 through 11 for real-time test

Error	*Derivative (d)*										
(e)	*NL*	*NM*	*NS*	*NVS*	*NT*	*ZE*	*PT*	*PVS*	*PS*	*PM*	*PL*
NL	1	1	7	3	4	1	1	1	11	6	4
NM	10	11	11	8	8	1	1	11	2	5	11
NS	3	10	1	11	6	6	11	9	3	9	2
NVS	11	10	11	7	1	6	11	2	7	1	4
NT	6	7	1	1	1	7	1	11	11	2	9
ZE	10	2	1	2	8	10	4	6	8	11	2
PT	3	11	6	2	9	1	7	8	4	4	6
PVS	7	3	9	7	5	5	6	1	9	1	10
PS	1	6	1	9	1	10	11	7	11	9	6
PM	10	1	11	2	4	7	1	11	8	1	2
PL	1	5	8	7	6	11	5	3	7	1	11

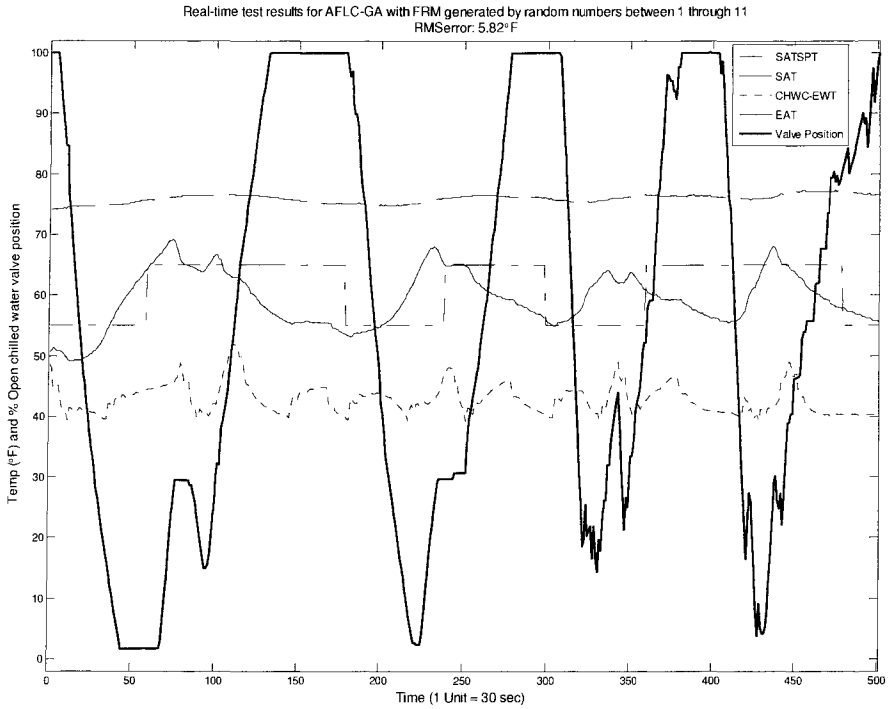

Figure 7.1-2: Real-time test results using FRM generated by random numbers between 1 through 11

From Figure 7.1-2, it is observed that results obtained for FRMs generated by random numbers using the GA technique resulted in poor FLC performance. Careful study of these FRMs showed that actions taken by the FLC were not correct, i.e. when the error was negative and decreasing, the chilled water valve should have been closed rapidly, but the opposite action was executed by opening the valve rapidly. Due to unacceptable performance of this FLC, another method for generating FRMs by changing the human FRM was studied.

7.1.2 Changing Human FRM by Random Numbers Generated amongst –1, 0, and 1 without changing Specific Elements

Using the GA technique, different FRMs were generated by modifying the Human FRM which is given in Table 5.2-2. The elements in the human FRM were modified by a number generated randomly amongst -1, 0 and 1 without changing five elements. These five unchanged elements are the four corners and the center element from the human FRM. This

assured that when the error was zero and derivative of error was zero, no control action was generated. Also, fixing the four corners assured correct control action in extreme situations. The use of the human FRM as a reference helped to avoid generating very erratic FRMs.

7.1.2.1 Offline Test

For offline study, different possible FRMs were generated as mentioned in the previous section. Using data from previous experiments and GRNN model II for predicting SAT, for every data point control signals were generated for the FRM being studied. The RMS error for each FRM was calculated. Using the GA technique, the best FRM was found for the case studied. For the offline test, the best generated FRM, shown in Table 7.1-4, having minimum RMS error resulted in good FLC performance. Figure 7.1-3 shows results for the offline study.

Table 7.1-4:FRM generated by changing Human FRM by random numbers generated amongst –1, 0, and 1 without changing specific elements
(Unchanged elements highlighted)

Error	*Derivative (d)*										
(e)	*NL*	*NM*	*NS*	*NVS*	*NT*	*ZE*	*PT*	*PVS*	*PS*	*PM*	*PL*
NL	1	1	2	1	4	2	5	5	4	6	6
NM	2	3	3	3	4	4	4	4	6	5	6
NS	2	3	4	4	4	4	4	4	6	6	8
NVS	3	3	2	5	3	4	4	5	8	7	7
NT	4	4	4	3	5	4	5	6	6	8	7
ZE	3	3	3	6	6	6	6	6	9	9	8
PT	3	3	5	4	6	7	6	8	7	8	10
PVS	3	4	5	5	8	7	9	7	10	8	9
PS	6	6	5	6	6	9	9	8	10	11	9
PM	6	7	7	6	8	8	10	9	11	10	11
PL	6	8	7	9	7	8	8	10	10	11	11

The FRMs in Table 7.1-4 look more logical than those generated in Table 7.1-2. Figure 7.1-3 shows results for the offline test.

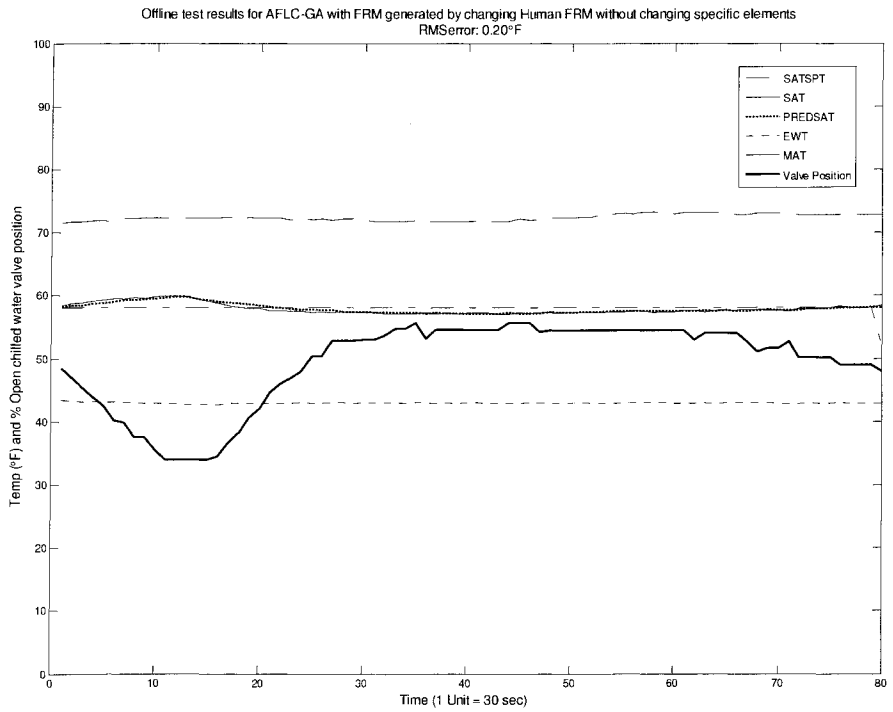

Figure 7.1-3: Offline test results for FRM generated by changing Human FRM by random numbers generated amongst –1, 0, and 1 without changing specific elements

7.1.2.2 Real-time Test

The same GA technique used in the offline test to generate FRMs was used for real time experiments. For the real-time test, after every 60 minutes SATSPT was cyclically changed between 55°F and 65°F. After every 120 minutes, a new FRM was generated. New FRMs were generated for each cycle during the real-time study by using the GA technique. Using the initial population and performing crossover and mutation operations for the predefined generations, the best FRM was obtained. This best FRM had the least RMS error for the data used in the study and was used in real-time test. Table 7.1-5 shows one of the FRM generated for the real time test.

Table 7.1-5: FRM generated by changing Human FRM by random numbers generated amongst –1, 0, and 1 without changing specific elements for real-time test (unchanged elements highlighted)

Error	*Derivative (d)*										
(e)	*NL*	*NM*	*NS*	*NVS*	*NT*	*ZE*	*PT*	*PVS*	*PS*	*PM*	*PL*
NL	1	1	1	1	2	2	4	4	4	4	6
NM	1	2	2	4	2	2	3	4	5	7	5
NS	3	1	2	3	4	4	4	3	7	8	7
NVS	2	3	3	5	3	4	5	6	7	8	7
NT	3	4	4	4	6	6	7	6	8	9	8
ZE	4	3	4	4	5	6	7	8	9	7	8
PT	3	5	4	4	7	7	7	9	8	8	9
PVS	4	4	5	6	6	7	7	9	10	9	9
PS	5	5	7	6	8	9	8	9	9	9	11
PM	6	7	8	6	8	9	9	8	9	10	10
PL	6	7	8	9	7	10	9	10	11	11	11

Figure 7.1-4 shows the results for real-time test for this FRM. Though improved results were obtained for FLC, the SAT response was oscillatory and more valve action was observed. Similar results were obtained for other real-time studies. So, different method for generating better FRM was tried.

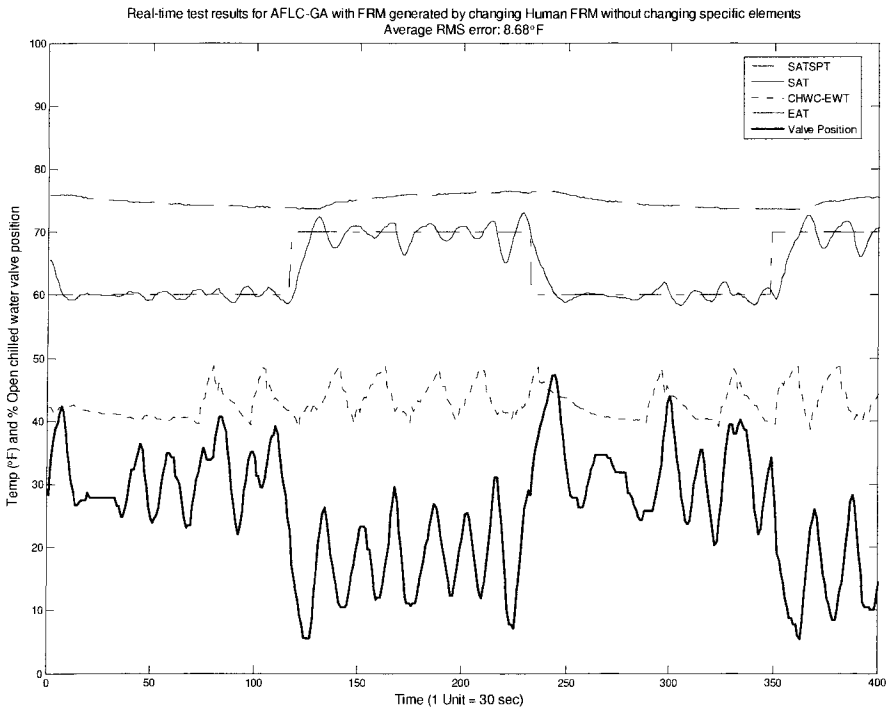

Figure 7.1-4: Real-time test results using FRM generated by changing Human FRM using random numbers generated amongst –1, 0, and 1 without changing specific elements

7.1.3 Changing Human FRM by Random Number Generated amongst –1, 0, and 1 without changing Specific Elements in the Outer Loop of the Human FRM

Another method that could improve FLC performance was to generate FRMs by restricting elements in the outer loop of the FRM. This method assured that when the system was out of control, correct control action will be taken.

7.1.3.1 Offline Test

Table 7.1-6 shows the FRM for minimum RMS error and Figure 7.1-5 shows the offline test results. Figure 7.1-5 shows that the FRM generated by changing the human FRM without changing elements in the outer loop had better control action than the previous methods studied. The RMS error for the offline study using this FRM was 0.19°F.

Table 7.1-6: FRM generated by modifying elements in the Human FRM without changing elements in the outer loop (unchanged elements highlighted)

Error	*Derivative (d)*										
(e)	*NL*	*NM*	*NS*	*NVS*	*NT*	*ZE*	*PT*	*PVS*	*PS*	*PM*	*PL*
NL	1	1	2	2	3	4	4	5	5	6	6
NM	1	2	2	3	4	4	5	5	6	6	6
NS	2	2	3	4	4	5	5	6	6	6	7
NVS	2	3	4	4	5	5	6	6	6	7	7
NT	3	4	4	5	5	6	6	6	7	7	8
ZE	4	4	5	5	6	6	6	7	7	8	8
PT	4	5	5	6	6	6	7	7	8	8	9
PVS	5	5	6	6	6	7	7	8	8	9	10
PS	5	6	6	6	7	7	8	8	9	10	10
PM	6	6	6	7	7	8	8	9	10	10	11
PL	6	6	7	7	8	8	9	10	10	11	11

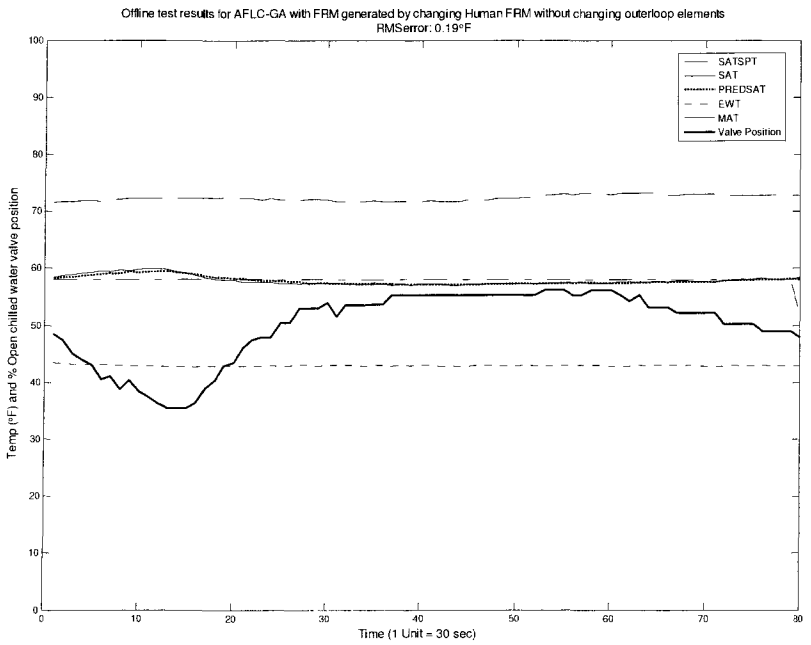

Figure 7.1-5: Offline test results using FRM generated by modifying elements in the Human FRM without changing elements in the outer loop

7.1.3.2 Real-time Test

Table 7.1 7 shows the FRM without changing the elements in the outer loop generated for the real time study.

Table 7.1-7: Fuzzy Logic Rule Matrix (11 by 11) generated without changing elements in the outer loop of the Human FRM (unchanged elements highlighted)

Error	*Derivative (d)*										
(e)	*NL*	*NM*	*NS*	*NVS*	*NT*	*ZE*	*PT*	*PVS*	*PS*	*PM*	*PL*
NL	1	1	2	2	3	4	4	5	5	6	6
NM	1	2	2	3	4	4	5	5	6	6	6
NS	2	2	3	4	4	5	5	6	6	6	7
NVS	2	3	4	4	5	5	6	6	6	7	7
NT	3	4	4	4	5	6	6	6	7	7	8
ZE	4	4	5	5	6	6	6	7	7	8	8
PT	4	5	5	6	6	6	7	7	8	8	9
PVS	5	5	6	6	6	7	7	8	8	9	10
PS	5	6	6	6	7	7	8	8	8	10	10
PM	6	6	6	7	7	8	8	9	10	11	11
PL	6	6	7	7	8	8	9	10	10	11	11

Figure 7.1 6 show the results for the real-time test. It was observed that when SAT was more than SATSPT, correct control actions were generated which causes a decrease in SAT and visa-versa. Also decreased overshoot and faster rise times was observed. The RMS error for the real time study was 1.16°F.

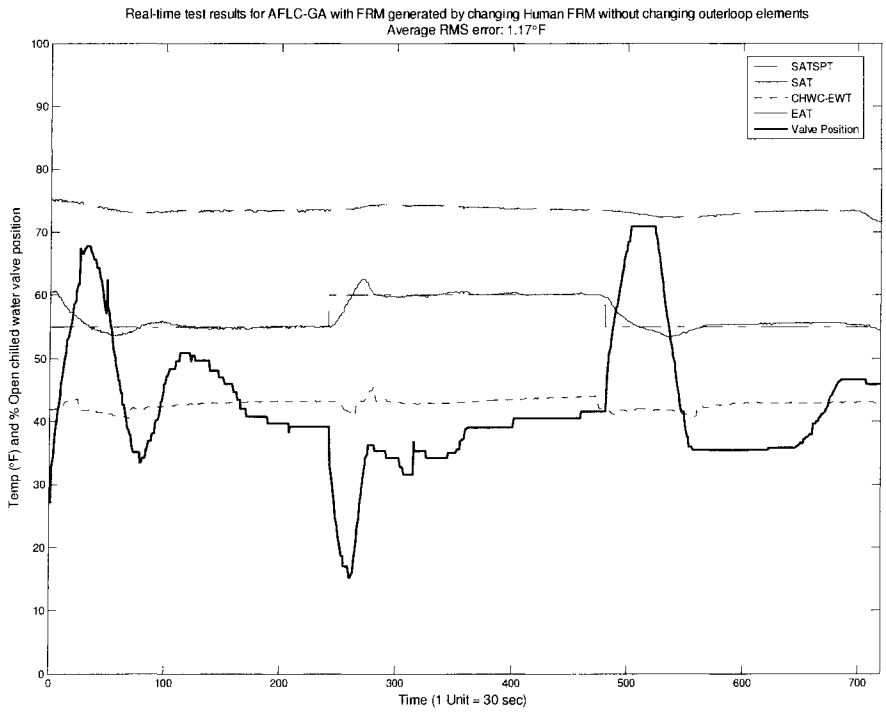

Figure 7.1-6: Real time test results for AFLC-GA with FRM generated by modifying the Human FRM without changing elements in the outer loop of the Human FRM

7.2 Summary

In this section, a method for developing an AFLC controller using Genetic Algorithms is discussed. Adaptive FLC controllers were developed offline and in real time by using Genetic Algorithms technique which generated different fuzzy rule matrix (FRM).

For development of adaptive FRMs, the linguistic representation of fuzzy membership functions for control actions was changed to a numeric representation.

Three different methods for generating FRMs are studied. In the first method, FRMs are initialized by generating random numbers between 1 through 11 for each element in the FRM. This method generated FRMs different than those generated by human intuition. The best FRM generated during the offline study had an RMS error of 0.316°F. But during real

time experiments, the same FRM resulted in poor performance of the FLC with an RMS error of 5.82°F.

In the second method, FRMs were generated by changing every element in the Human FRM by a random numbers amongst –1, 0, and 1 without changing the four corner and center elements of the Human FRM. This method of generating FRMs takes advantage of human intuition, thus avoiding generating absurd actions that cause large RMS errors. The best FRM generated during the offline study had an RMS error of 0.20°F. During the real time experiments, FRMs were evolved and the performance of the FLC improved and RMS error of 2.2°F was observed

In the third method, FRMs were generated by changing the Human FRM by random numbers generated amongst –1, 0, and 1 without changing the elements in the outer loop and the center element of the Human FRM. Using this method assured that FRMs generated would take correct control action when the system was out of control. Offline results showed an RMS error of 0.197°F, which was the least of the three methods studied, and real time experiments had an RMS error of 1.168°F.

7.3 AFLC using Evolutionary Strategies (AFLC-ES)

Two techniques used for the development of an AFLC using ES were evolving Scaling Factors and Mapping Factors.

Scaling Factor

A Scaling Factor is a number which is used to scale the input and/or output variables before the fuzzification and/or after defuzzification process respectively. One of the methods used to develop an adaptive FLC controller was to modify the scaling factors used for error (e), derivative of error (d) and control signal (u). Details for evolving scaling factors are given in section 5.3.3

For the offline test, data from the previous experiments and GRNN Model II for predicting SAT were used. This data set includes values for CHWC-EWT, MAT, SATSPT, SACFM

and CHWP-GPM. Initially, the reference gene [0 0 0 0 0 0 0 0 0 0 0 0] was used for the offline test. After every 240 sampling periods, a new gene was evolved. Offline test data included a step change in SATSPT. This allowed checking the response of new evolved scaling factors for a step change in SATSPT.

Up to 3 bits of the total 12 bits from the parent gene were modified by randomly generated binary numbers to evolve new gene. Using this new gene, new scaling factors were calculated for e, d and u as explained in section 5.3.3 and then new fuzzy membership functions for the FLC. For the cycle of 240 data points, using GRNN model II to predict SAT for every data point, the RMS error between predicted SAT and SATSPT was calculated for every gene evolved. Smaller values of total RMS error indicate better FLC performance for that configuration of gene. If the RMS error using this new gene was less than the error for the parent gene, then this new gene replaces the parent gene and is evolved further. Otherwise, the new gene is discarded. This process of generating a new gene was continued for a predefined number of generations or a termination criterion is achieved. Termination criteria used for this study was zero RMS error.

Mapping Factor

A Mapping Factor is a number which modifies uniform fuzzy membership functions to non-uniform fuzzy membership functions as shown in Figure 5.3-4.

To develop an AFLC, another technique was to evolve powere, powerd, and poweru. Section 5.3.4 gives details for application of mapping factors in development of an AFLC. For offline tests the reference gene was [0 0 0 0 0 0 0 0 0 0 0 0]. Maximum and minimum values used for each factor are given in Table 7.3-1.

Table 7.3-1: Maximum and Minimum values for each mapping factor

Mapping Factor	Minimum Value	Maximum Value
Error	1	3
Derivative of error	1	2
Control signal	1	3

Values of SFerror, SFderivative of error and SFcontrol signal were held constant at 10°F, 5°F/s and 20% open/sampling period respectively while evolving mapping factors.

Starting with the above reference gene, the offline test was initiated. After every 240 sampling periods, a new gene was evolved by modifying part of the parent gene. Data used for this offline study had a step change in SATSPT. This allowed checking the response of the new evolved fuzzy membership functions for a step change in SATSPT.

A new gene was evolved by modifying up to 3 bits, at a time, of the total 12 bits parent gene by randomly generated binary numbers. Using this new gene, new mapping factors are calculated for e, d and u as explained in section 5.3.4 and then new fuzzy membership functions are calculated. For the cycle of 240 data points, using the GRNN model II to predict SAT for every data point, the RMS error between predicted SAT and SATSPT was calculated for every evolved gene. Smaller values of total RMS error indicate better FLC performance of that gene. If the RMS error using this new gene is less than that of parent gene, then this new gene replaces parent gene and is evolved further. Otherwise, the new gene is discarded. This process of generating a new gene was continued for a predefined number of generations.

7.3.1 Offline Test Results for AFLC-ES with evolving Scaling Factor

Figure 7.3-1 shows the variation in the SAT for the best evolved genes obtained during the offline test. It also shows that FLC performance was good for SATSPT of 65°F but poor of 55°F. Longer rise time was observed when SATSPT was 55°F. Real-time tests were conducting using the best gene generated for the offline test.

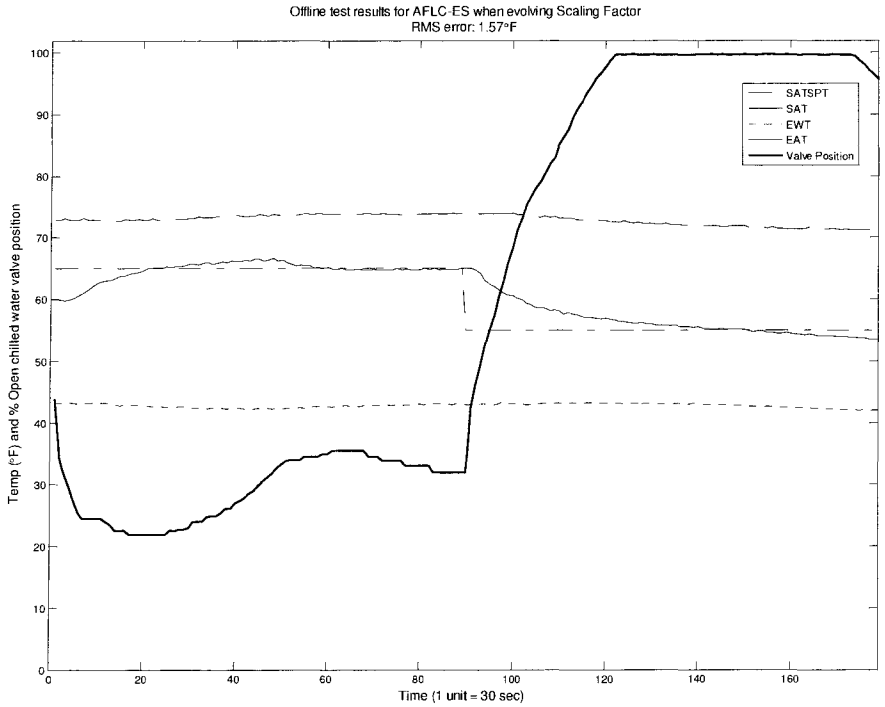

Figure 7.3-1: Offline test results using scaling factor

7.3.2 Real-time Test Results for AFLC-ES with evolving Scaling Factor

For the real-time study, the best gene, [1 0 1 0 0 0 1 0 1 1 1 1], evolved in the offline test was used as a parent gene. Figure 7.3-2 shows results for AFLC when evolving scaling factors for three sets. It was observed that even though several genes were evolved, the FLC performance was deteriorating (results not shown in Figure 7.3-2). More oscillatory responses and more chilled water valve actions were observed. No improvement in the FLC performance was observed during real-time experiments. Thus, no further studies were conducted for evolving scaling factors. The RMS error for each cycle for real-time experiments is shown in the Table 7.3-2.

Table 7.3-2: RMS error for real-time test with ES evolving scaling factor

Cycle	RMS error (°F)
1	2.56
2	2.22
3	2.59

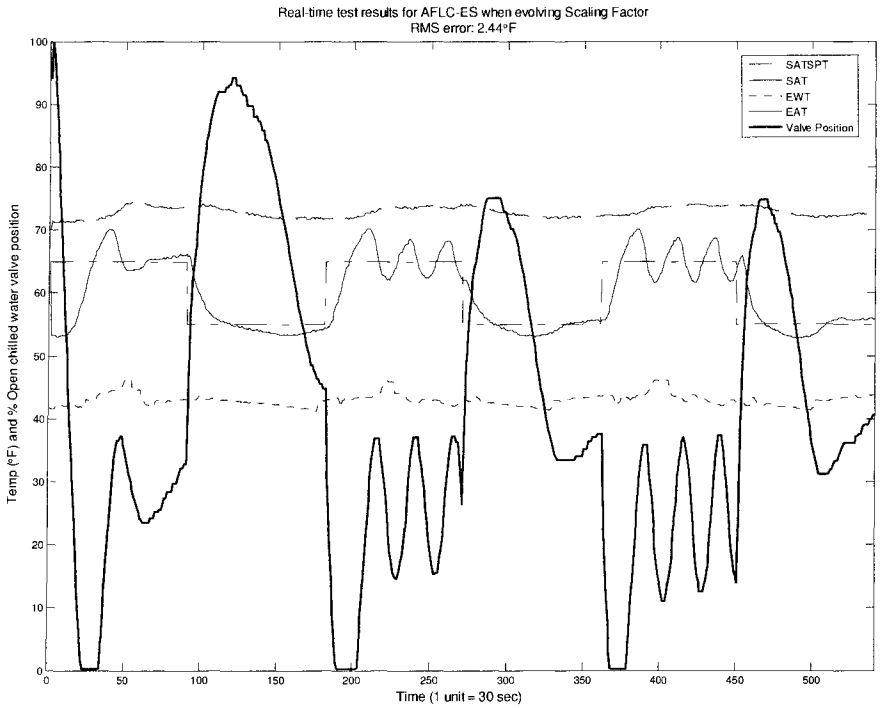

Figure 7.3-2: Real time test results with varying scaling factor.
(new scaling factor is evolved after every 90 minutes)

7.3.3 Offline Test Results for AFLC-ES with evolving Mapping Factor

As discussed in section 5.3.4, another technique to develop AFLC using ES was to evolve mapping factors. For the offline study, mapping factors were evolved as mentioned in section 5.3-3. Figure 7.3-3 shows the variation in the SAT for better evolved genes during the offline test. It also shows that using this technique for evolving mapping factors works better than evolving scaling factors.

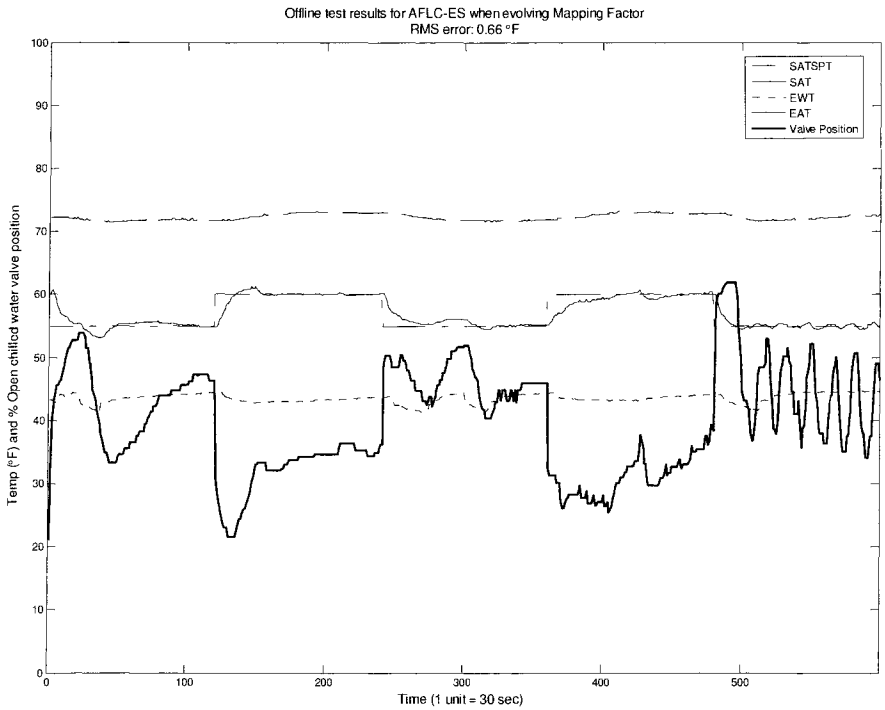

Figure 7.3-3: Offline test results for AFLC with evolving mapping factor.
(new mapping factor was evolved after every 120 minutes)

7.3.4 Real-time Test Results for AFLC-ES with evolving Mapping Factor

For the real-time test, the same parent gene used in the offline test was used as the initial gene and then evolved as tests proceeded. The same evolutionary technique explained in section 5.3.4 was used. Figure 7.3-4 shows the results obtained for the real-time test. Table 7.3-3 shows total RMS error for each cycle of 120 minutes. It can be clearly observed that total RMS error was decreasing indicating improvement in the AFLC-ES performance. This shows the potential application of ES for evolve mapping factors to develop an Adaptive Fuzzy Logic Controller.

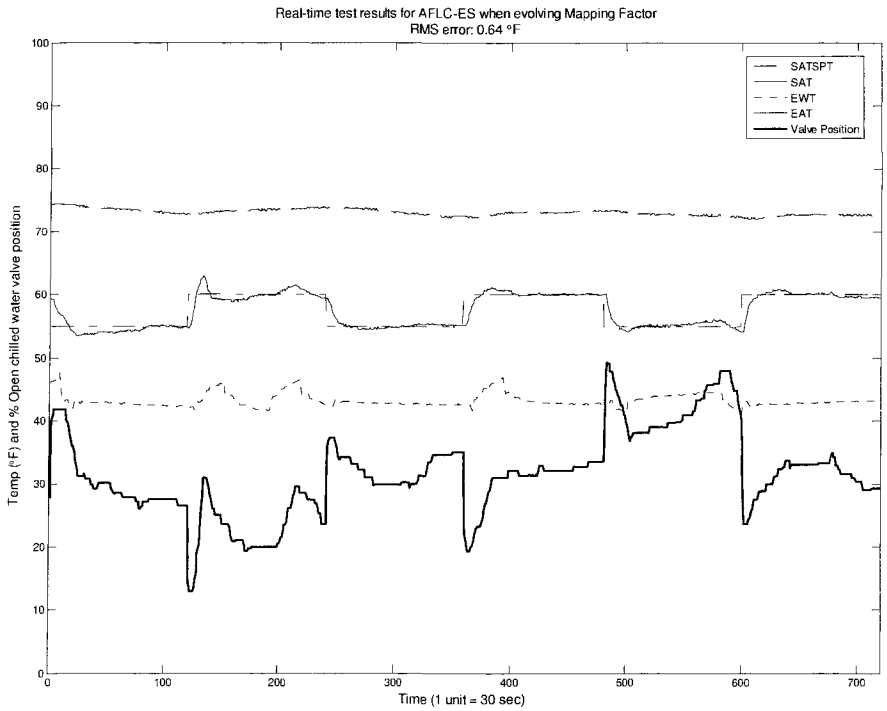

Figure 7.3-4: Real-time test results for AFLC-ES with evolving mapping factor (new mapping factor was evolved after every 120 minutes)

Table 7.3-3: RMS error resulted for real-time study of mapping factor

Cycle	RMS error (°F)
1	0.99
2	0.5
3	0.5

7.4 Summary

In this chapter, methods to develop an adaptive FLC controller are discussed. Adaptive FLC controllers were developed offline and in real-time by using Genetic Algorithms to alter FRMs and Evolutionary Strategies to alter scaling factors and mapping factors.

Three different techniques using Genetic Algorithm techniques were studied for generating FRMs in order to improve the performance of the AFLC. The best results using GA

technique were obtained for generating FRMs by changing elements from the human FRM without changing elements in the outer loop. This method of generating FRMs ensured correct control action when the system is out-of-control while utilizing the advantage of human expertise.

Using the Evolutionary Strategies technique, scaling factors for error (e), derivative of error (d) and control signal (u) were altered simultaneously. The offline tests showed improvements in FLC performance. Real-time experiments were conducted, but no significant improvement in the FLC performance was observed.

In another offline study using the Evolutionary Strategies technique, mapping factors were evolved. A mapping factor modified the fuzzy membership functions and hence the action taken by the FLC. These modifications were based on the uniform fuzzy membership functions and were bounded. Also fuzzy membership functions in error, derivative of error and control signal were modified simultaneously. The offline tests showed good results and demonstrated its usefulness for developing an adaptive FLC. Real-time results for AFLC-ES when evolving mapping factor were better and thus showed potential application for development of Adaptive Fuzzy Logic Controllers.

Chapter 8 Comparison test results for AFLC-GA/ES and PIDL

8.1 Introduction

An Adaptive Fuzzy Logic Controller (AFLC) was developed using the Genetic Algorithm (GA) and Evolutionary Strategy (ES) techniques to modify parameters in the AFLC. Experiments were conducted to compare the performance of the AFLC with that of the Proportional Integral Derivative Loop (PIDL) controller.

As mentioned in the experimental facility description, the Energy Resource Station (ERS) combines laboratory testing capabilities with real building characteristics and is capable of simultaneously testing two side-by-side full-scale commercial HVAC systems with identical thermal loads. The side-by-side HVAC systems allow System A to be controlled by an AFLC, while System B is controlled by the EMCS using a PIDL controller. The controllers can be switched between the systems to remove any bias. Because each test system has the same loads (external and internal) and construction, the only difference in the comparison tests would be the operation and control methodologies. The performance of both the controllers in terms of chilled water valve travel, reduced energy usage, faster response time, reduced overshoot, and reduced settling time was compared and the benefits of the proposed operation and control methodology were quantified.

Two different comparison tests were conducted, namely:

- To study and compare the response of the controllers for changes in the supply air temperature set point (SATSPT).
- To study and compare the response of the controllers for cyclic changes in the supply air volume flow rate (SA-CFM).

After discussion with ERS staff, it was decided to vary SAT cyclically between 52°F and 58°F. After conducting several real time experiments and discussing results of changing SATSPT tests with the ERS staff, the ERS manager (Klaassen, 2004) suggested conducting a second set of comparison tests that would have cyclic variations in SACFM with constant SATSPT of 58°F. These tests would be more realistic, since in commercial HVAC

applications, changes in the SATSPT are rarely observed, whereas sudden changes in SACFM are common to meet varying loads.

The test conditions for the AHU settings for conducting the first comparison tests are given below:

1. The system operation was scheduled to be in an occupied mode for 24 hours.
2. The outdoor air damper position was set to be 0% open, exhaust air damper was set to be 0% open and return air damper was set to be 100% open.
3. The economizer control was disabled.
4. The AHU supply air pressure set point was 1.4 in. W.G., because two diffusers were employed for each zone.
5. The zone temperatures were maintained between 70°F and 72°F.
6. The supply air static pressure set point was at 2.0 W.G
7. The minimum entering air flow rates were 400 cfm for the exterior zones and 100 cfm for the interior zones for the A- and B-Test Systems.
8. The supply air temperature set point was varied periodically between 52°F and 58°F.

A test set up sheet that provides the details of real-time test conditions is provided in Appendix H.

Figure 8.1-1 and Figure 8.1-2 show the variations in the Lighting and Baseboard load heat schedules respectively in the zones. Figure 8.1-3 shows the total load from lighting and baseboard. The same load schedule was maintained throughout these comparison tests conducted at the ERS.

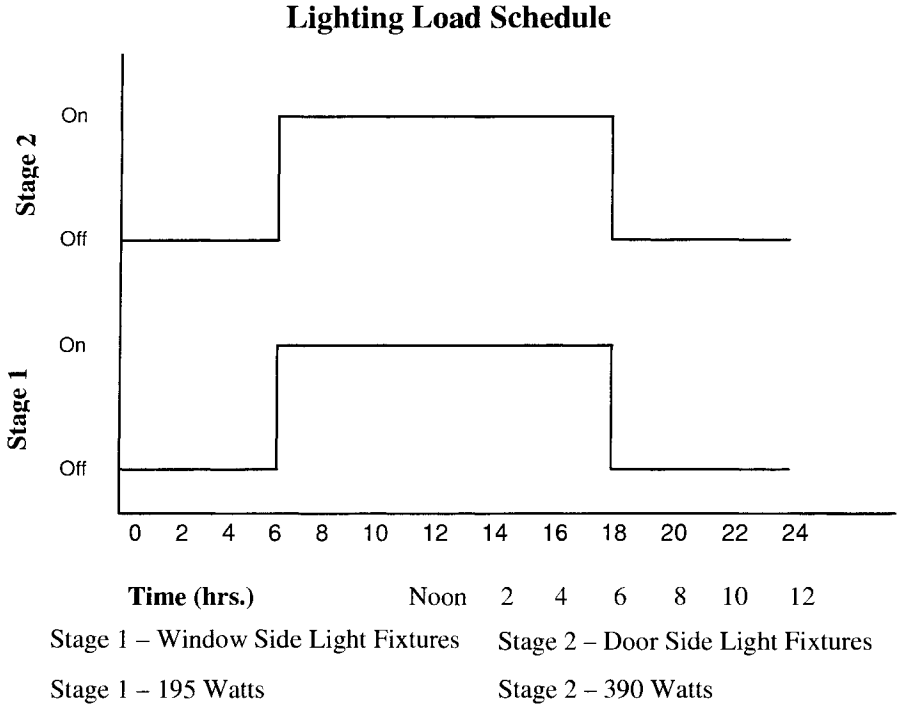

Figure 8.1-1: Lighting load schedule in the zones

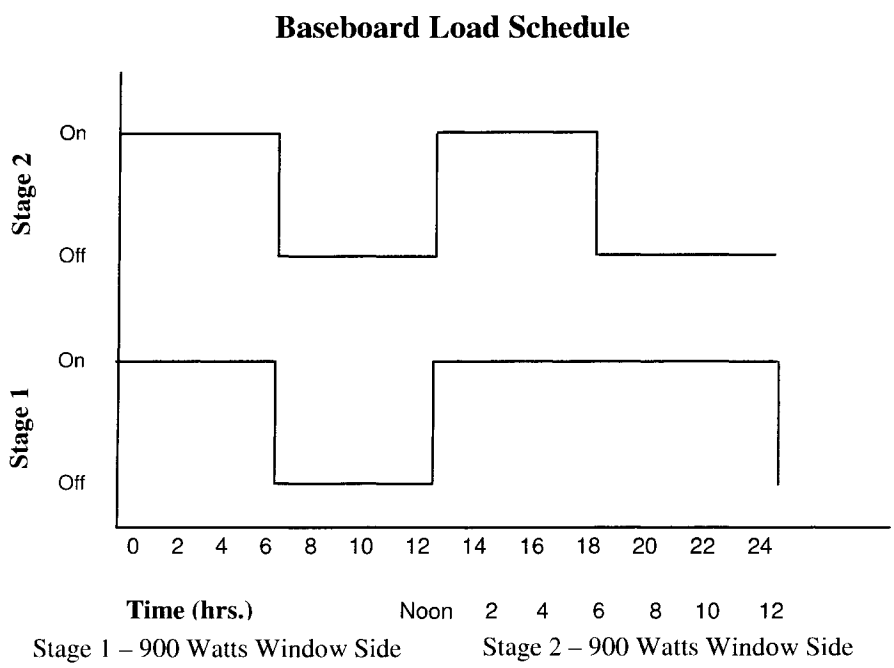

Figure 8.1-2: Baseboard load schedule in the zones

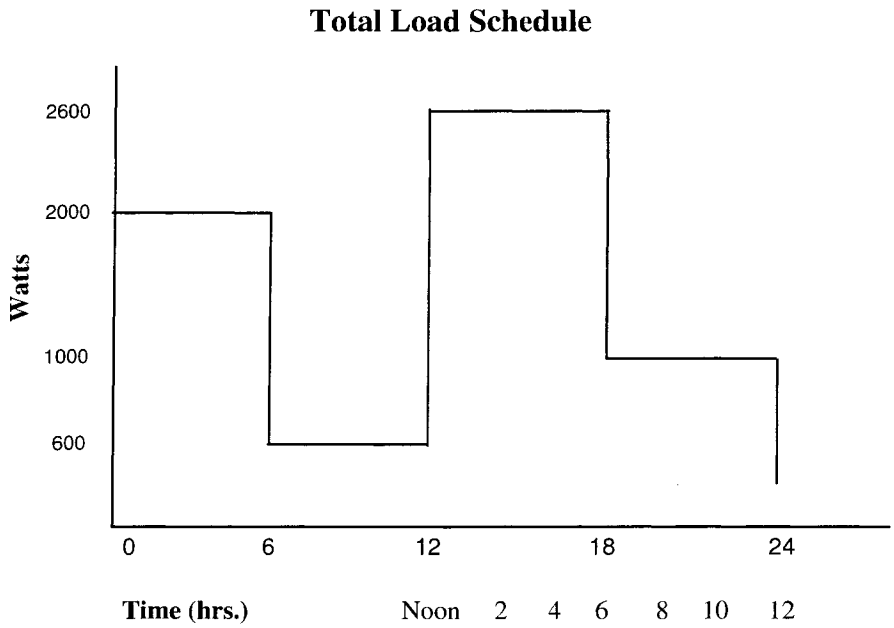

Figure 8.1-3: Total load schedule in the zones

8.2 Comparison Test Results for Variation in SATSPT with AFLC-GA on AHUA

In this comparison test, experiments were conducted with the AFLC controlling AHUA and PIDL controller controlling AHUB for variation in SATSPT. In this experiment, GAs were used to develop the AFLC by evolving better FRMs. Since GAs were used to develop an Adaptive Fuzzy Logic Controller it was named AFLC-GA. The SATSPT was cyclically changed between 52°F and 58°F after every 45 minutes.

Figure 8.2-1 shows the overall real-time results. It was observed that the AFLC-GA was able to control the chilled water valve position to maintain the SAT within +/- 3°F. Figure 8.2-2 and Figure 8.2-3 show a zoom-in for the sections when the AFLC-GA had better performance. It is noticed from Figure 8.2-1, Figure 8.2-2 and Figure 8.2-3 that:

1. The CHWC-EWT's were similar for both the AHU's and varied between 41°F and 44°F.
2. The MAT's were similar for both of the AHU's and varied between 69°F and 74°F.

3. The chilled water valve movement was more for AHUA with the AFLC.
4. The SATSPT was simultaneously changed for both the AHU's.

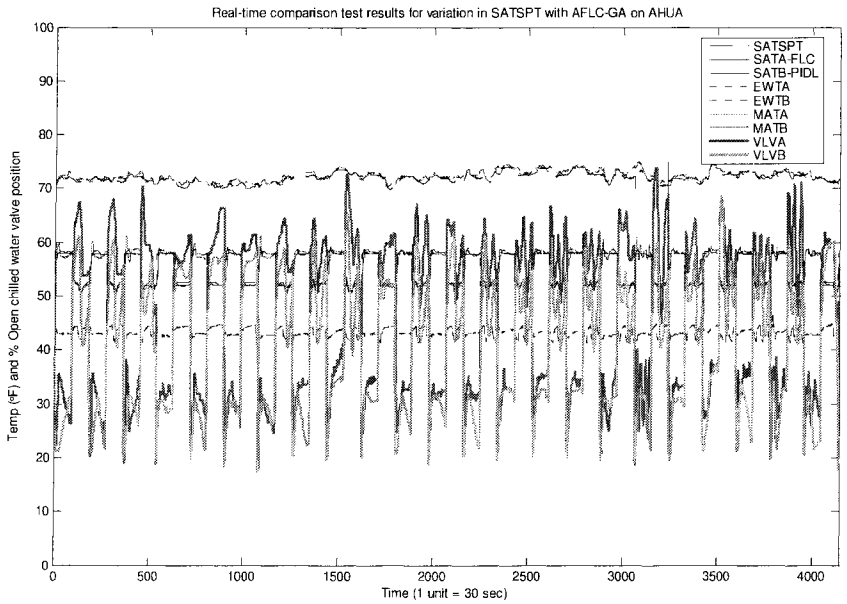

Figure 8.2-1: Comparison test results for variation in SATSPT

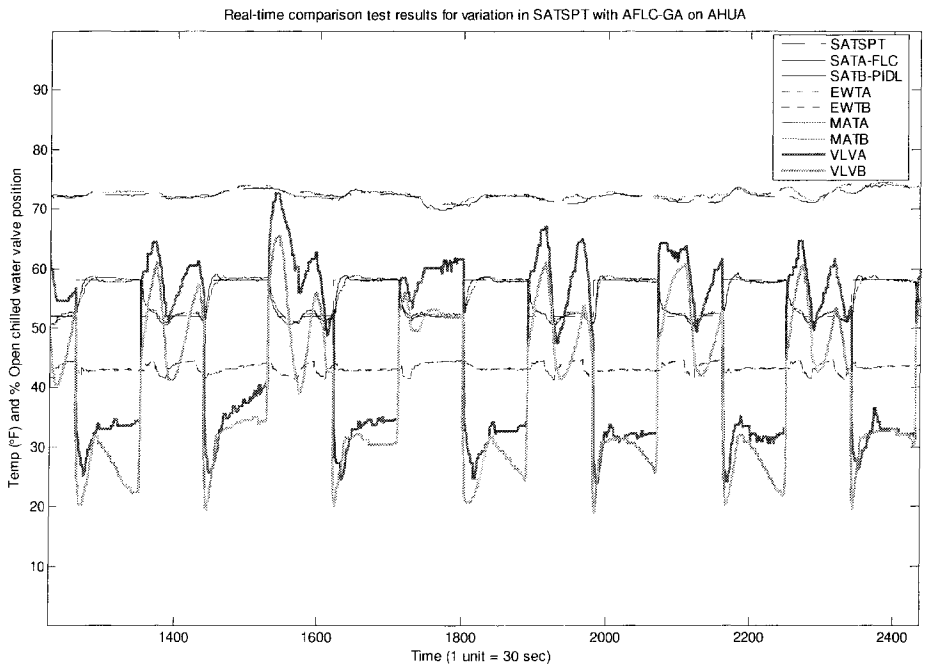

Figure 8.2-2: Zoom-in comparison test results for variation in SATSPT

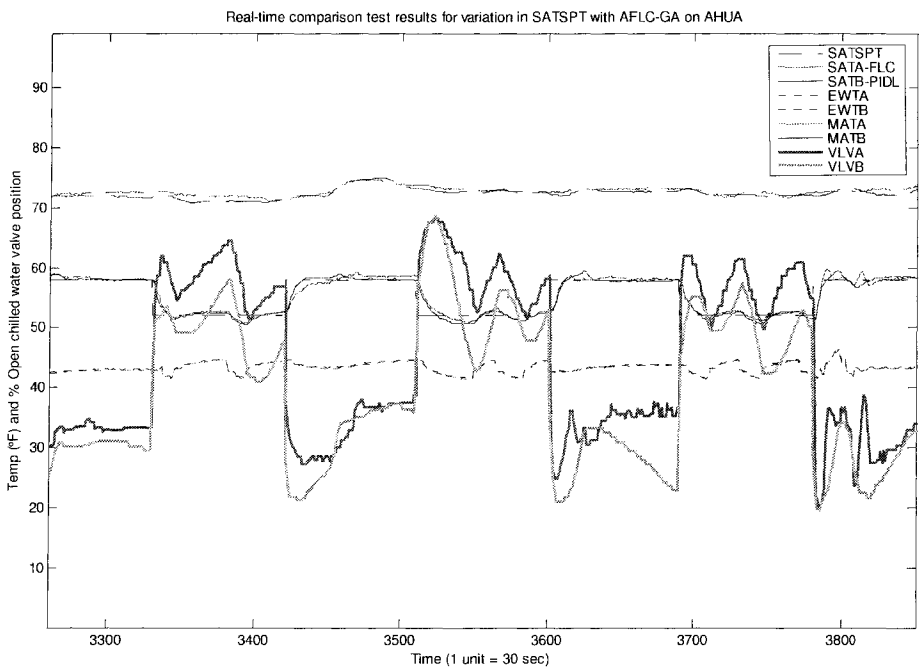

Figure 8.2-3: Zoom-in comparison test results for variation in SATSPT

8.2.1 RMS Error

As mentioned earlier, the performance of the AFLC-GA was evaluated using the standard statistical measures of root mean square (RMS) error, which is defined as

$$\text{RMSError} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\text{SAT-SATSPT})^2} \qquad \text{... 8.2-1}$$

where,

SAT = Supply Air Temperature (°F)

SATSPT = Supply Air Temperature Set-Point (°F)

n = number of data patterns

Table 8.2-1 and Figure 8.2-4 shows RMS error/cycle. One cycle is defined as changing SATSPT from 52°F to 58°F and then back to 52°F. It was noticed that for most of the cycles, the AHUA controlled by the AFLC-GA had higher RMS error compared to AHUB controlled by the PIDL controller, but it was decreasing with the number of cycles as the AFLC-GA adapted to the existing conditions.

Table 8.2-1: Comparison of total RMS error/cycle between SATA (AFLC-GA) and SATB (PIDL) for variation in SATSPT test

Comparison Test Results for variation in SATSPT AFLC-GA on AHUA			
Average RMS error/cycle (°F/cycle)			
	Controller		
Cycle No.	AFLC-GA (AHUA) A	PIDL (AHUB) B	% Difference (A -B)*100/B
1	0.94333	0.69722	35.30
2	0.93278	0.74056	25.96
3	1.0267	0.62333	64.71
4	0.66889	0.35056	90.81
5	0.87389	0.52333	66.99
6	0.89778	0.50722	77.00
7	0.86944	0.53333	63.02
8	0.79833	0.72611	9.95
9	0.91	0.71778	26.78
10	0.69611	0.36222	92.18
11	0.81	0.66556	21.70
12	0.745	0.59167	25.91
13	0.72944	0.76111	-4.34
14	0.67611	0.54944	23.05
15	0.755	0.63111	19.63
16	0.78056	0.62389	25.11
17	0.83444	0.56167	48.56
18	0.93833	0.67389	39.24
19	0.73389	0.58778	24.86
20	0.94778	0.705	34.44
21	0.59556	0.66722	-12.03
22	0.68833	0.69556	-1.05
23	0.74389	0.64889	14.64

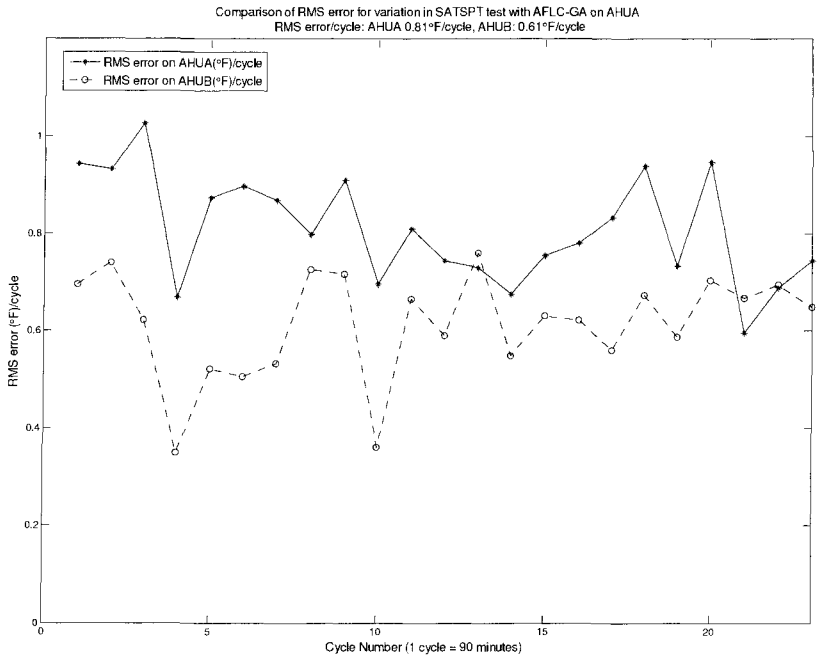

Figure 8.2-4: Comparison of RMS error/cycle between AHUA (AFLC-GA) and AHUB (PIDL) for variation in SATSPT test

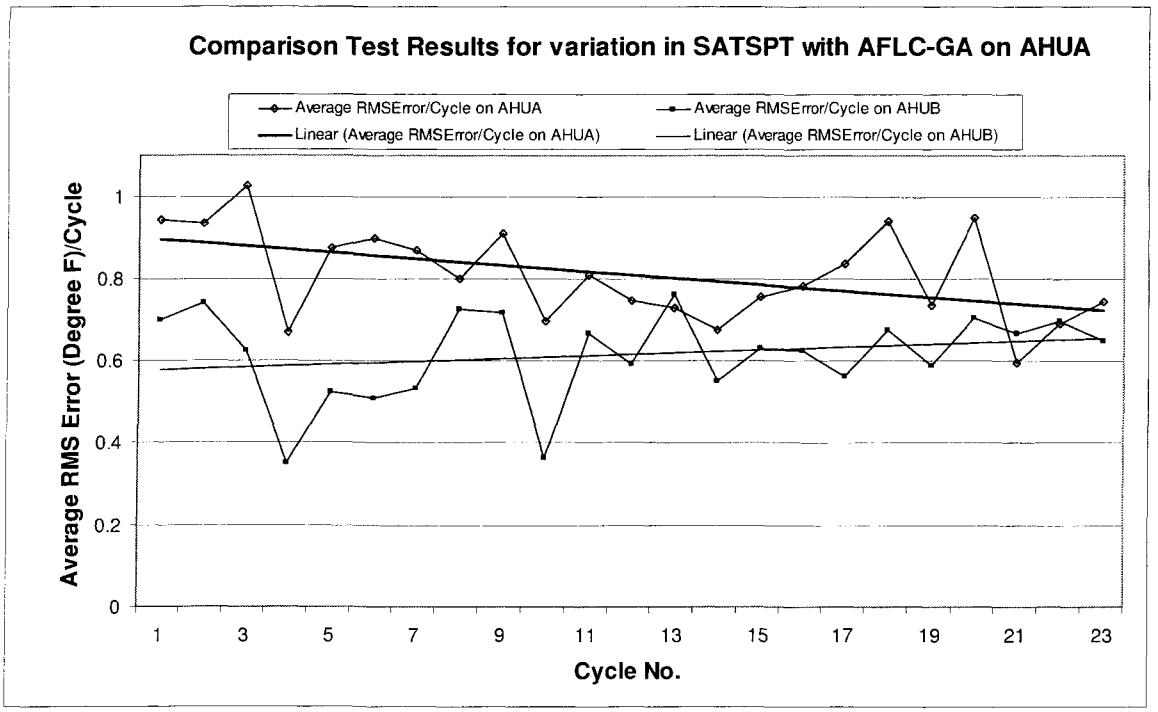

Figure 8.2-5: Comparison of RMS error/cycle with trend line between SATA (AFLC-GA) and SATB (PIDL) for variation in SATSPT test

Figure 8.2-5 shows RMS error/cycle with a trend line. It is observed that the trend line for AHUA controlled by the FLC-GA has a downward slope whereas for AHUB controlled by the PIDL it has an upward slope. Thus it was expected that if this experiment would have been continued, RMS error/cycle for AHUA would have decreased further as the AFLC-GA adapted to the operating conditions.

8.2.2 Hydronic Energy

Using the methodology and equations explained in the performance indices section, the hydronic energy consumption for each cycle was calculated and is shown in Table 8.2-2.

Table 8.2-2: Comparison of total hydronic energy/cycle consumption between AHUA (AFLC-GA) and AHUB (PIDL) for variation in SATSPT test

Comparison Test Results for variation in SATSPT AFLC-GA on AHUA			
Average Hydronic Energy/Cycle (MBTU/Cycle)			
Cycle No.	AFLC-GA (AHUA) A	PIDL (AHUB) B	% Difference (A -B)*100/B
1	0.0835	0.0879	-5.09
2	0.0830	0.0881	-5.86
3	0.0869	0.0908	-4.31
4	0.0876	0.0879	-0.33
5	0.0820	0.0860	-4.66
6	0.0834	0.0861	-3.06
7	0.0855	0.0896	-4.53
8	0.0866	0.0904	-4.23
9	0.0841	0.0901	-6.62
10	0.0862	0.0884	-2.54
11	0.0841	0.0882	-4.68
12	0.0872	0.0892	-2.24
13	0.0866	0.0900	-3.77
14	0.0864	0.0910	-5.03
15	0.0877	0.0918	-4.49

Table 8.2-2: continued

16	0.0878	0.0923	-4.89
17	0.0881	0.0926	-4.92
18	0.0851	0.0904	-5.83
19	0.0851	0.0892	-4.53
20	0.0834	0.0872	-4.34
21	0.0847	0.0891	-4.90
22	0.0849	0.0901	-5.75
23	0.0872	0.0884	-1.32

From the results in Table 8.2-2 and Figure 8.2-6, it is observed that the average hydronic energy/cycle for AHUA controlled by an AFLC-GA is approximately 4.3% less than for AHUB controlled by PIDL controller.

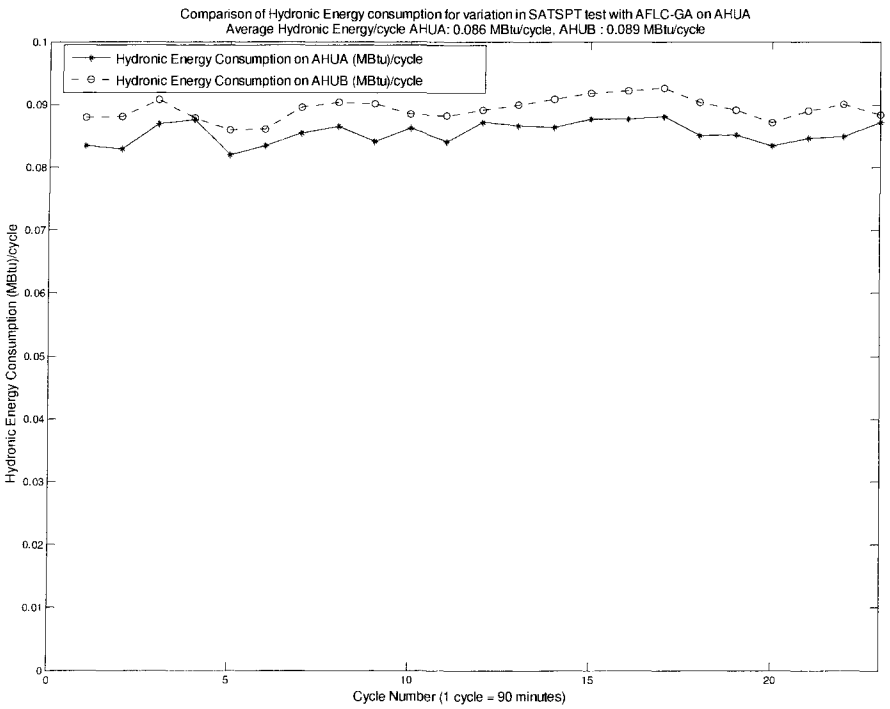

Figure 8.2-6: Comparison of hydronic energy consumption /cycle on AHUA (AFLC-GA) and AHUB (PIDL) for variation in SATSPT test

8.2.3 Actuator Travel Distance

Table 8.2-3: Comparison of total actuator travel distance/cycle between chilled water valve on AHUA (AFLC-GA) and AHUB (PIDL) for variation in SATSPT test

Comparison test results for variation in SATSPT AFLC-GA on AHUA Average Chilled Water Valve Movement/Cycle			
	Controller		
Cycle No.	AFLC-GA (AHUA) A	PIDL (AHUB) B	% Difference (A -B)*100/B
1	174.41	128.43	35.80
2	118.93	114.59	3.79
3	140.19	93.22	50.39
4	89.46	53.96	65.79
5	104.37	78.89	32.30
6	118.15	80.74	46.33
7	107.14	94.36	13.54
8	104.16	118.76	-12.29
9	130.13	116.62	11.58
10	93.89	60.97	53.99
11	138.05	108.68	27.02
12	111	89.65	23.81
13	138.56	132.77	4.18
14	123.56	95.65	29.18
15	142.1	102.08	39.20
16	120.86	107.53	12.40
17	128.16	87.82	45.93
18	253.48	105.32	140.68
19	126.91	93.1	36.32
20	124.03	123.72	0.25
21	163.84	115.21	29.68
22	269.18	119.45	55.62
23	142.86	95.21	50.05

Table 8.2-3 and Figure 8.2-7 show the total actuator travel distance/cycle (ATD/cycle). ATD/cycle for AHUA, controlled by an AFLC-GA, was approximately 35.0% more than for AHUB, controlled by PIDL controller.

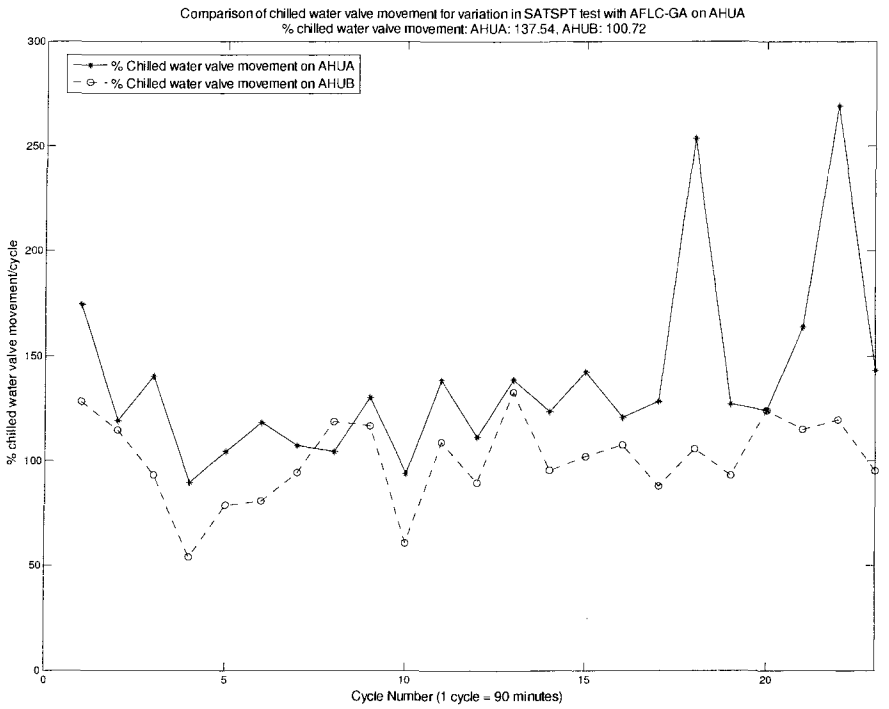

Figure 8.2-7: Comparison of actuator travel distance/cycle between chilled water valve on AHUA (AFLC-GA) and AHUB (PIDL) for variation in SATSPT test

8.2.4 Rise Time, Overshoot and Setting Time

Figure 8.2-8 shows the variation in SAT for AHUA and AHUB when AHUA was under FLC-GA and AHUB under PIDL control. CHWC-EWT was same for both the AHU's and had very little variation. Less than 2°F variation was observed in MAT.

Both controllers were able to control the process within the required limits. Also controllers acted immediately to the step change in the SATSPT by changing the chilled water valve position to achieve the new SATSPT sooner.

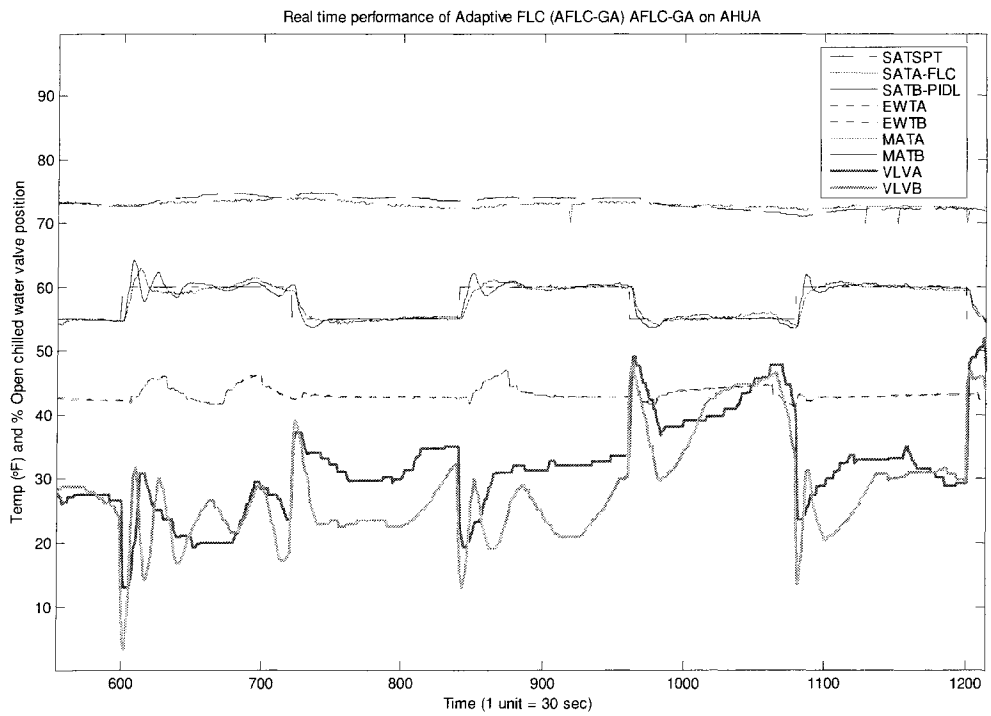

Figure 8.2-8: Variation in SAT for AHUA (AFLC-GA) and AHUB (PIDL) along with the other variables

Figure 8.2-9 shows detail for the above process. From Table 8.2-4 it is observed that even though the controlled variable for PIDL had less rise time, it also had more overshoot as compared to an AFLC-GA. Also, the settling time for the PIDL was more than that of the AFLC-GA.

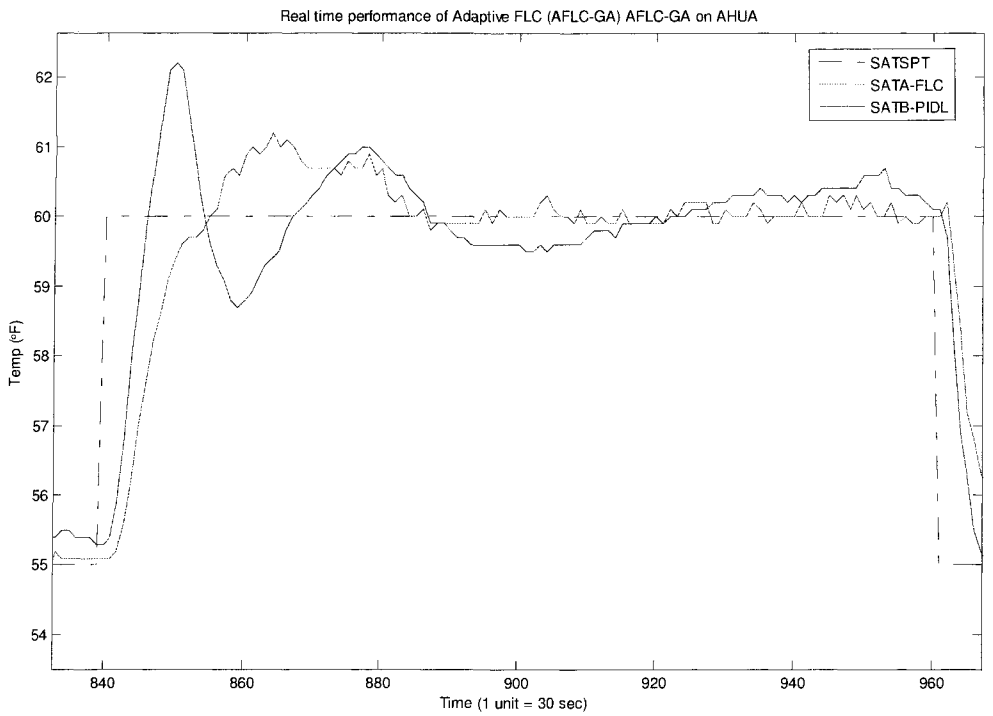

Figure 8.2-9: Case I - Comparison of rise time, overshoot and settling time between SATA (AFLC-GA) and SATB (PIDL) for variation in SATSPT test

Table 8.2-4: Case I - Comparison of rise time, overshoot and settling time between SATA (AFLC-GA) and SATB (PIDL) for variation in SATSPT test

	AFLC-GA (AHUA)	PIDL (AHUB)
Rise Time (sec)	15	7
Overshoot (°F)	0.7	2.2
Settling Time (Time Steps)	40	45

Figure 8.2-10 shows another case. From Figure 8.2-10 and Table 8.2-5: Case II - it is observed that the controlled variable for the AFLC-GA had longer rise time but much less overshoot as compared to PIDL. Also, the settling time was less for AFLC-GA than for the PIDL.

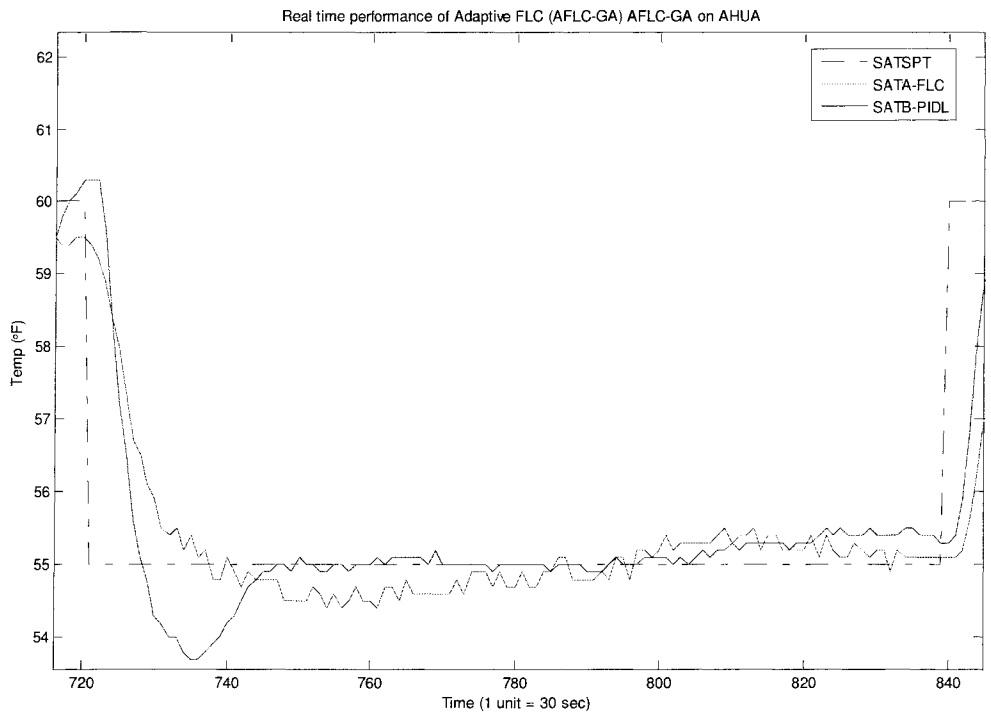

Figure 8.2-10: Case II - Comparison of rise time, overshoot and settling time between SATA (AFLC-GA) and SATB (PIDL) for variation in SATSPT test

Table 8.2-5: Case II - Comparison of rise time, overshoot and settling time between SATA (AFLC-GA) and SATB (PIDL) for variation in SATSPT test

	AFLC-GA (AHUA)	PIDL (AHUB)
Rise Time (sec)	12	6
Overshoot (°F)	0.6	1.3
Settling Time (Time Steps)	12	22

Figure 8.2-11 shows another case. From Table 8.2-6 and Figure 8.2-11 it is observed that even though the controlled variable for the AFLC-GA and PIDL had the same rise times, the AFLC-GA had less overshoot and a shorter settling time than the PIDL.

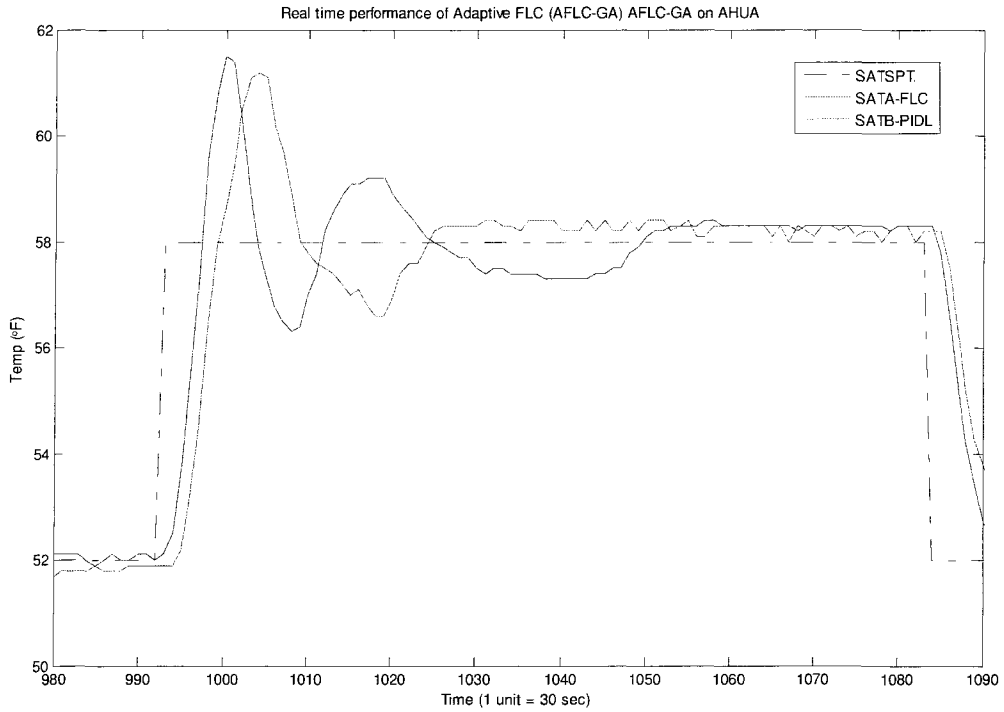

Figure 8.2-11: Case III - Comparison of rise time, overshoot and settling time between SATA (AFLC-GA) and SATB (PIDL) for variation in SATSPT test

Table 8.2-6: Case III – Comparison of rise time, overshoot and settling time between SATA (AFLC-GA) and SATB (PIDL) for variation in SATSPT test

	AFLC-GA (AHUA)	PIDL (AHUB)
Rise Time (sec)	3	3
Overshoot (°F)	3.2	3.5
Settling Time (Time Steps)	30	38

8.3 Comparison Tests Results for Variations in SACFM with AFLC-ES on AHUA

For this comparison test, an ES technique was used to develop the AFLC-ES by evolving better fuzzy membership functions. The SATSPT was held constant at 58°F and the SACFM was varied from 1600 to 2200 in steps of 200 cfm every hour. The test conditions for the second comparison test were similar to those used in the comparison test for variation in SATSPT. Initially, experiments were conducted with the AFLC-ES controlling AHUA and the PIDL controller controlling AHUB and then with AFLC-ES on AHUB and PIDL on AHUA. A test set up sheet that provides the details of the experimental conditions is in Appendix H. Figure 8.3-1 shows the scheduled CFM for both of the AHU's.

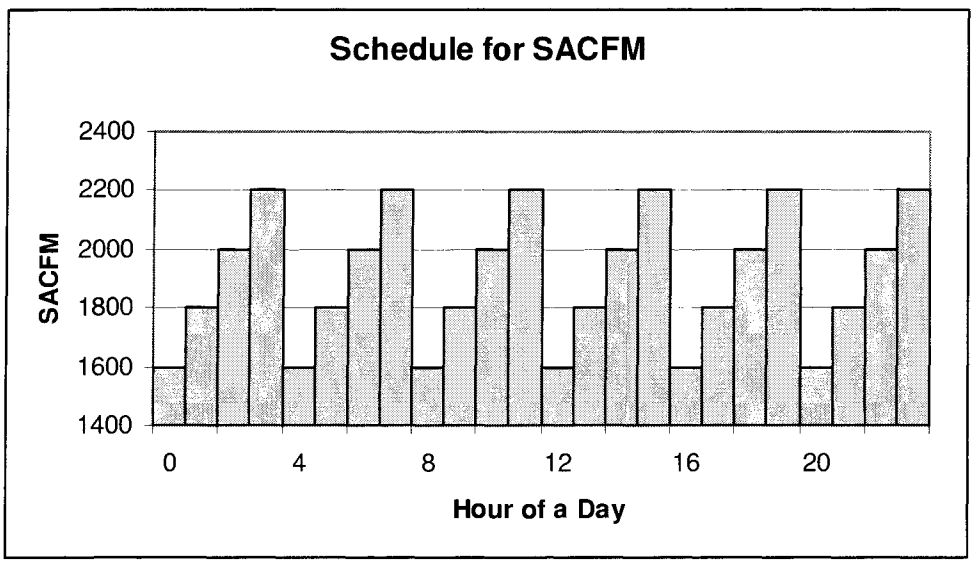

Figure 8.3-1: SACFM schedule

Figure 8.3-2 shows the overall real-time results. Figure 8.3-3 and Figure 8.3-4 show details for the sections when the AFLC-ES performance is better. It is noticed that:

1. CHWC-EWT's were similar for both the AHU's and varied between 42°F and 45°F.
2. MAT's were similar for both the AHU's and varied between 69°F and 74°F.
3. The SATSPT was held constant at 58°F for both the AHU's.

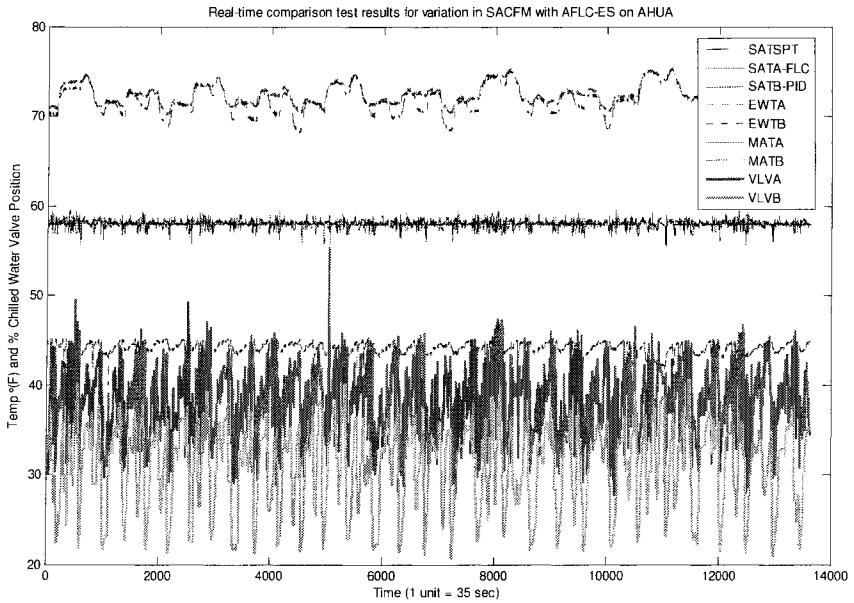

Figure 8.3-2: Comparison test results for variation in SACFM

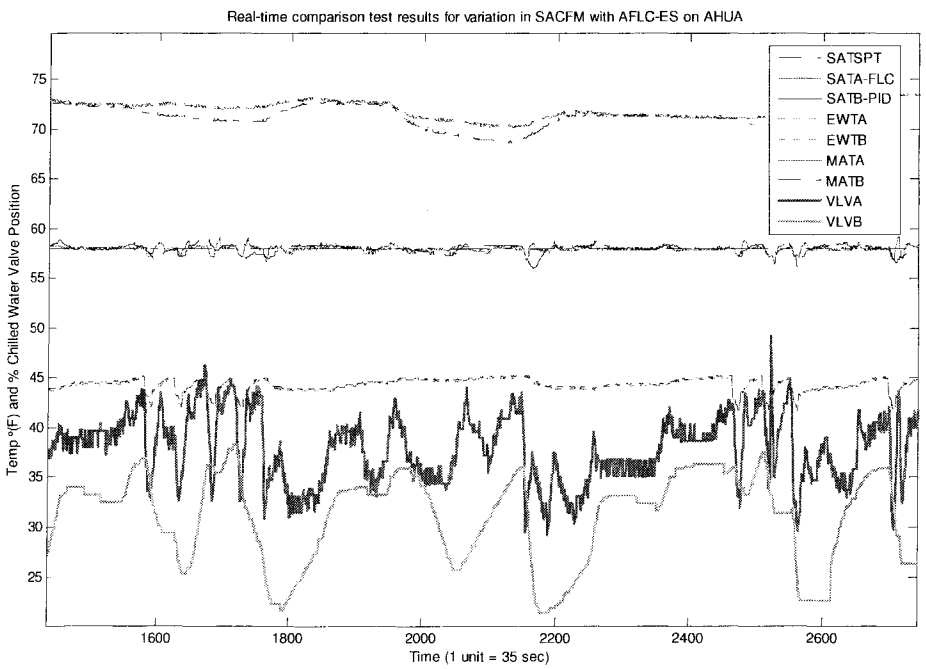

Figure 8.3-3: Zoom-in comparison test results for variation in SACFM

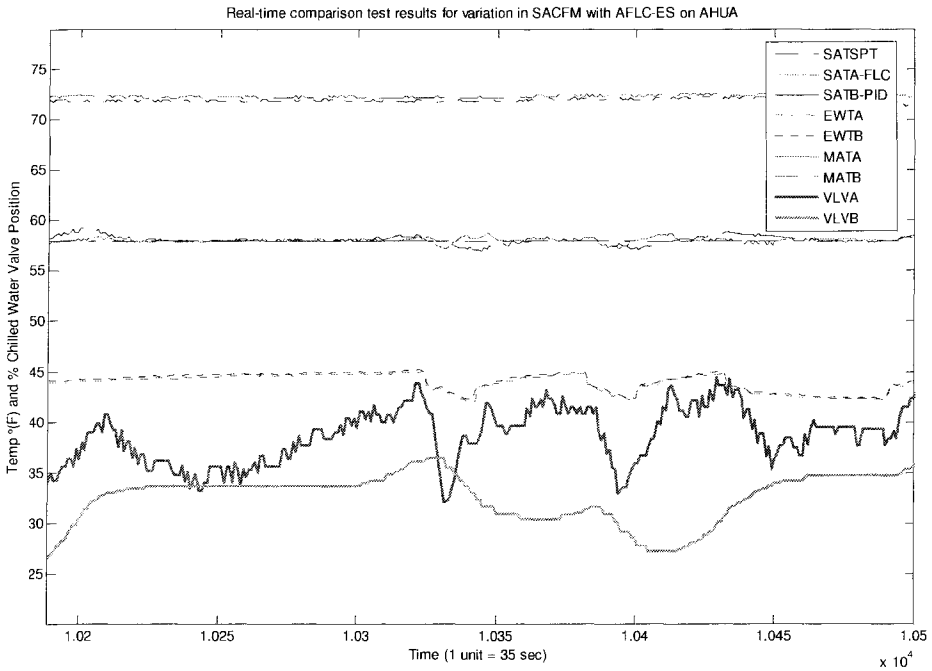

Figure 8.3-4: Zoom-in comparison test results for variation in SACFM

Figure 8.3-5 shows the variation in the SATA and SATB with scheduled change in SACFM. It was observed that even though the SACFM was varied, the SATA and SATB were maintained almost within ±1°F by both controllers.

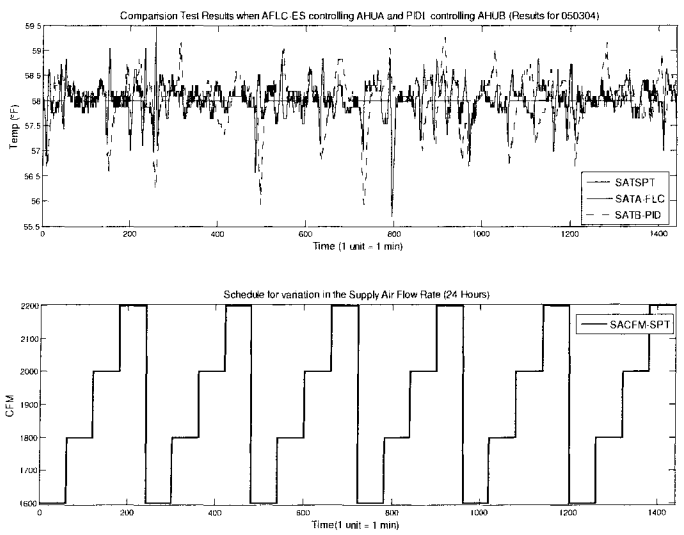

Figure 8.3-5: Variation in SATA (AFLC-ES) and SATB (PIDL) and SACFM schedule

Figure 8.3-6 through Figure 8.3-9 show the resulting SATA for the AFLC-ES and SATB for the PIDL controller along with the variation in the SACFMs in both AHU's.

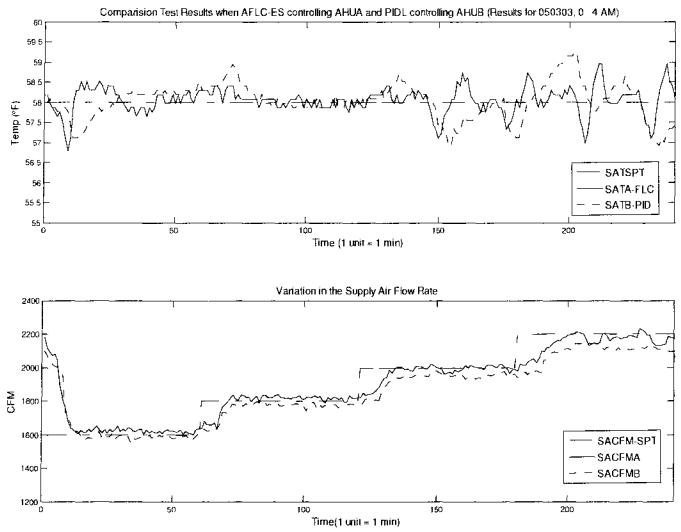

Figure 8.3-6: Zoom-in (0 AM – 4 AM) for SATA and SATB along with variation in the SACFM for 2nd test day

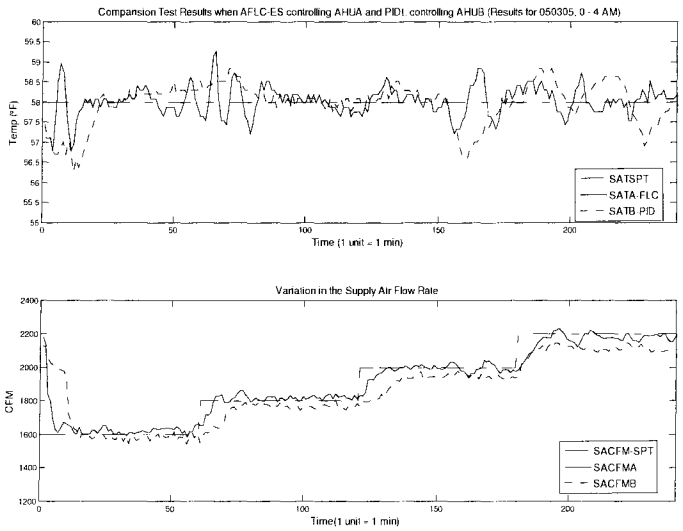

Figure 8.3-7: Zoom-in results (0 AM – 4 AM) for SATA and SATB along with variation in the SACFM for 4th test day

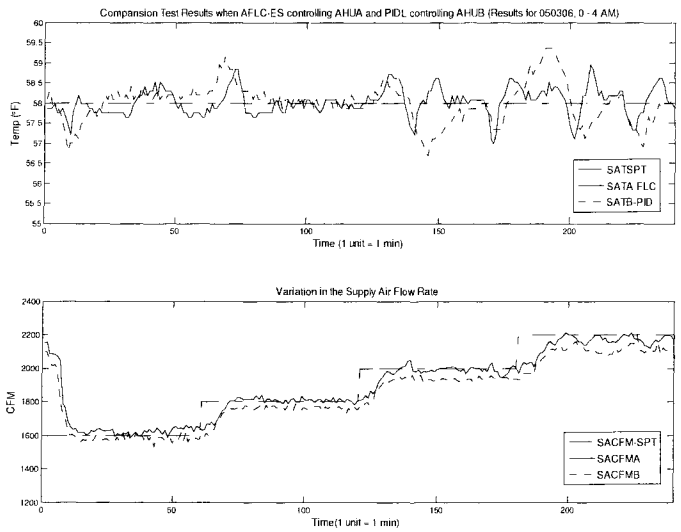

Figure 8.3-8: Zoom-in results (0 AM – 4 AM) for SATA and SATB along with variation in the SACFM for 5th test day

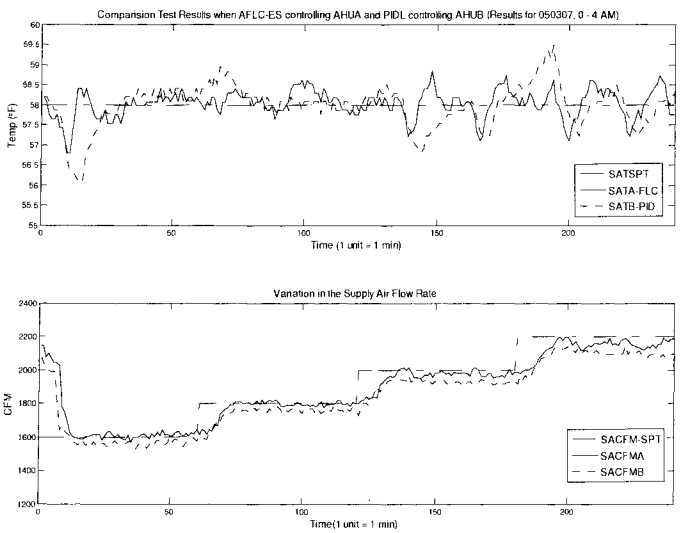

Figure 8.3-9: Zoom-in results (0 AM – 4 AM) for SATA and SATB along with variation in the SACFM for 6th test day

Thus it was observed that the AFLC-ES was able to control the chilled water valve position so as to maintain the SAT within 1°F of the SATSPT.

8.3.1 RMS Error

Table 8.3-1 and Figure 8.3-10 show hourly RMS error for the 7th day of the comparison test for variation in SACFM. Most of the hours with AHUA controlled by the AFLC-ES had almost 13% lower hourly RMS error as compared to that for AHUB controlled by the PIDL controller. The average hourly RMS error for the 7th day for the AFLC-ES on AHUA was 0.24°F and 0.30°F for the PIDL on AHUB.

Table 8.3-1: Comparison of average hourly RMS error between SATA (AFLC-ES) and SATB (PIDL) for variation in SACFM test

Comparison test results for variation in SACFM AFLC-ES on AHUA Test Results for 3/8/05			
Average Hourly RMS error (°F)			
	Controller		
Hour of a Day	AFLC-ES (AHUA) A	PIDL (AHUB) B	% Difference (A -B)*100/B
1	0.38935	0.50565	-23.00
2	0.1985	0.14655	35.45
3	0.1479	0.20522	-27.93
4	0.29838	0.26793	11.36
5	0.14137	0.36803	-61.59
6	0.1655	0.10705	54.60
7	0.2481	0.34435	-27.95
8	0.1473	0.2726	-45.96
9	0.4557	0.42648	6.85
10	0.19833	0.19827	0.03
11	0.1767	0.2088	-15.37
12	0.20652	0.4377	-52.82
13	0.32433	0.37288	-13.02
14	0.27878	0.32337	-13.79
15	0.3614	0.47737	-24.29

Table 8.3-1: continued

16	0.3676	0.39925	-7.93
17	0.25678	0.33595	-23.57
18	0.1855	0.21982	-15.61
19	0.1799	0.1889	-4.76
20	0.21518	0.30587	-29.65
21	0.23458	0.41298	-43.20
22	0.1469	0.2146	-31.55
23	0.1203	0.20837	-42.27

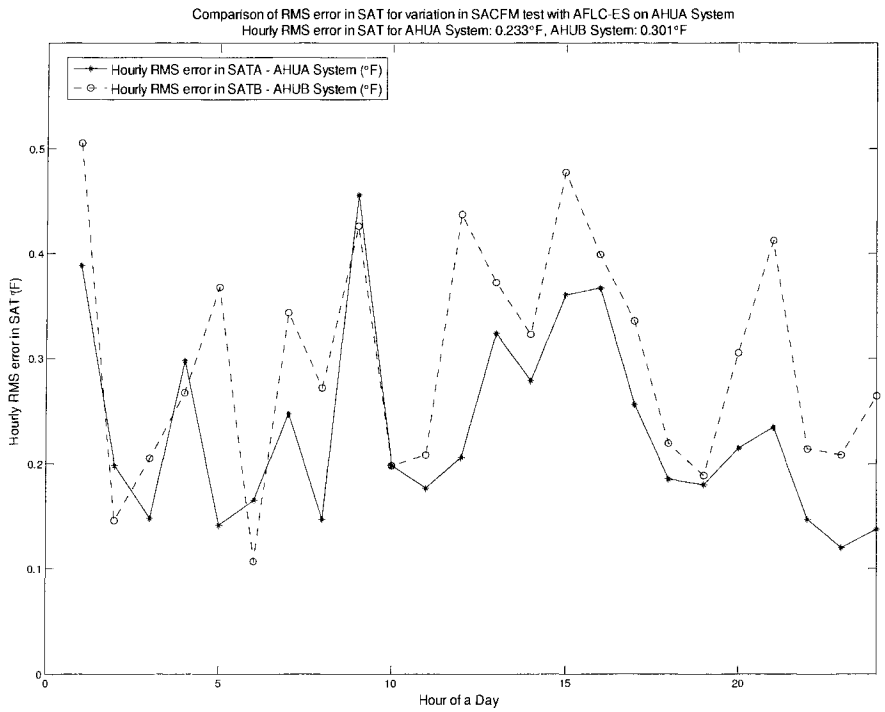

Figure 8.3-10: Comparison of average hourly RMS error between SATA (AFLC-ES) and SATB (PIDL) for variation in SACFM test

8.3.2 Hydronic Energy

Table 8.3 2 and Figure 8.3 11 show the average hydronic energy/hr consumption. Most of the hours with AHUA controlled by the AFLC-ES had almost 6.2% lower hydronic energy/hr as compared to that for AHUB controlled by the PIDL controller. The average hydronic energy for the 7th day for the AFLC-ES on AHUA was 0.100 MBTU/hr and 0.115 MBTU/hr for the PIDL on AHUB.

Table 8.3-2: Comparison of average hydronic energy/hr consumption between AHUA (AFLC-ES) and AHUB (PIDL) for variation in SACFM test

Comparison test results for variation in SACFM AFLC-ES on AHUA Test Results for 3/8/05			
Average Hydronic Energy/hr (MBTU/hr)			
	Controller		
Hour of a Day	AFLC-ES (AHUA) A	PIDL (AHUB) B	% Difference (A -B)*100/B
1	0.109	0.1136	-4.04
2	0.1028	0.1106	-7.08
3	0.1008	0.1067	-5.57
4	0.0975	0.1055	-7.54
5	0.1104	0.1160	-4.82
6	0.1063	0.1130	-5.96
7	0.0972	0.1036	-6.18
8	0.0895	0.0980	-8.63
9	0.1045	0.1062	-1.59
10	0.1029	0.1109	-7.22
11	0.0938	0.1027	-8.64
12	0.0954	0.0999	-4.50
13	0.1044	0.1116	-6.41
14	0.0989	0.1076	-8.07
15	0.0995	0.1061	-6.22

Table 8.3-2: continued

16	0.0975	0.1042	-6.47
17	0.1091	0.1166	-6.47
18	0.1009	0.1092	-7.64
19	0.0935	0.1022	-8.52
20	0.0927	0.0965	-3.92
21	0.1046	0.1083	-3.43
22	0.1047	0.1126	-6.99
23	0.0966	0.1028	-6.00

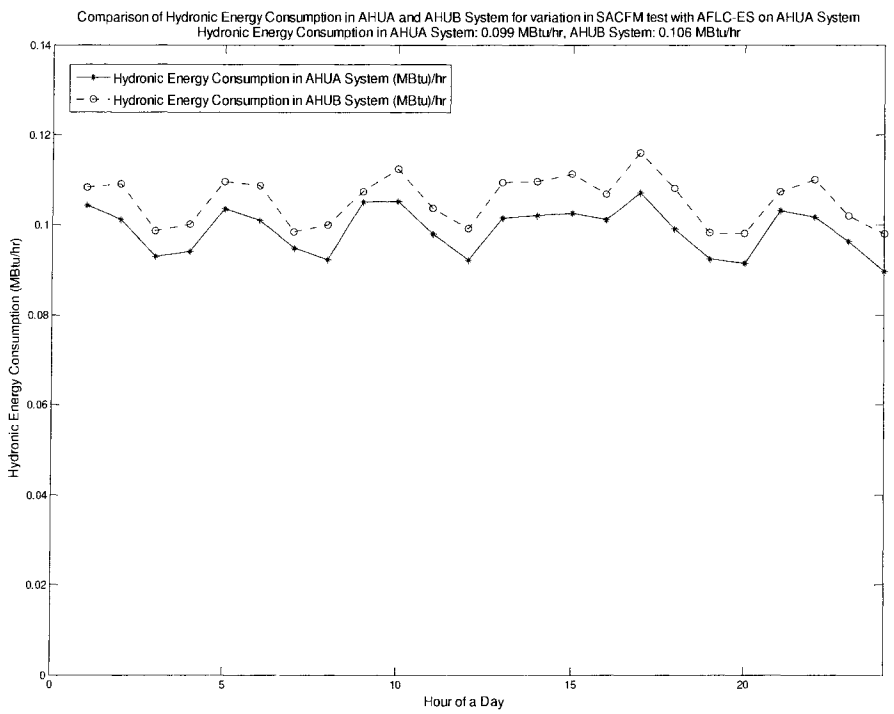

Figure 8.3-11: Comparison of average hydronic energy/hr consumption between AHUA (AFLC-ES) and AHUB (PIDL) for variation in SACFM test

8.3.3 Actuator Travel Distance

Table 8.3-3 and Figure 8.3-12 show the total actuator travel distance/hr. AHUA controlled by the AFLC-ES had approximately 435.0% more valve movement than that for AHUB controlled by the PIDL controller. The average of total actuator travel distance for the 7th day

Table 8.3-3: Comparison of total actuator travel distance/hr between chilled water value on AHUA (AFLC-ES) and AHUB (PIDL) for variation in SACFM test

Second Comparison Test Results AFLC-ES on AHUA Test Results for 3/8/05			
Average % Chilled Water Valve Movement/Hr			
	Controller		
Hour of a Day	AFLC-ES (AHUA) A	PIDL (AHUB) B	% Difference (A -B)*100/B
1	96.58	56.10	72.17
2	42.56	11.63	265.84
3	75.56	14.92	406.52
4	71.60	17.94	299.22
5	120.91	19.74	512.67
6	57.53	10.85	430.07
7	47.87	14.20	237.03
8	56.97	11.27	405.43
9	63.47	17.69	258.71
10	47.02	7.56	521.93
11	54.73	6.21	780.94
12	69.96	13.48	419.01
13	57.04	14.85	284.11
14	53.88	4.62	1065.98
15	78.29	15.06	419.77
16	73.56	16.85	336.63
17	85.88	16.74	412.92
18	34.55	7.13	384.33
19	33.05	8.32	297.26
20	51.59	11.59	345.27
21	50.78	7.97	537.55
22	83.80	9.19	812.32
23	52.75	8.96	488.59

for the AFLC-ES on AHUA was 63.47% valve movement/hr and 14.04% valve movement/hr for the PIDL on AHUB.

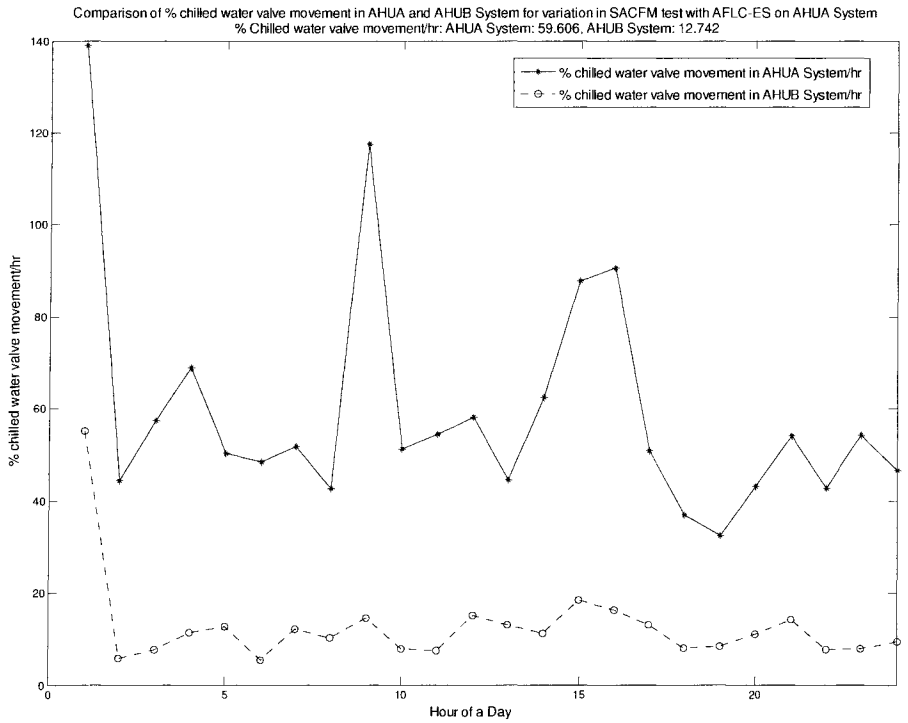

Figure 8.3-12: Comparison of total actuator travel distance/hr between chilled water value on AHUA (AFLC-ES) and AHUB (PIDL) for variation in SACFM test

Similarly the calculations for all six days when the FLC was controlling AHUA and AHUB were done. Table 8.3-4 and Figure 8.3-13 show the results for the performance indices obtained for the second comparison test when AHUA was controlled by the AFLC-ES and the PIDL was used on AHUB.

Table 8.3-5 shows the percentage difference in the values for each of the performance indices for the AFLC-ES results as compared to the PIDL results. Negative percentage value of the performance indices means AFLC-ES performance was better than that of PIDL controller. The first column shows the percentage difference RMS error, the second column shows the % difference in the chilled water valve movement/hr and the third shows the percentage difference in hydronic energy/hr. The RMS error for the AFLC-ES was 24% less than for the PIDL controller; hydronic energy/hr was 5.9% less than that for the PIDL controller, whereas the AFLC-ES had 300% more valve movement/hr than the PIDL controller.

Table 8.3-4: Comparison of performance indices between AHUA (AFLC-ES) and AHUB (PIDL) for variation in SACFM test

Comparison Test Results for variation in SACFM						
AFLC-ES on AHUA						
Test Date	RMS error (°F)		% Chilled water valve movement/hr		Hydronic Energy (MBtu/hr)	
YMMDD	AHUA	AHUB	AHUA	AHUB	AHUA	AHUB
50303	0.25	0.31	55.9	14.17	0.1008	0.1073
50304	0.25	0.34	53.5	15.26	0.1015	0.1075
50305	0.26	0.35	62.86	15.72	0.1029	0.1089
50306	0.24	0.34	50.46	15.64	0.1028	0.1093
50307	0.24	0.30	63.47	14.04	0.1005	0.1071
50308	0.23	0.30	59.61	12.74	0.0989	0.1055

Table 8.3-5: Percentage difference in performance indices between AHUA (AFLC-ES) and AHUB (PIDL) for variation in SACFM test

Comparison test results for variation in SACFM AFLC-ES on AHUA		
Percentage (AFLC-ES - PIDL)/PIDL		
RMS error	% Chilled water valve movement	Hydronic Energy
-19.03	294.38	-6.06
-28.42	250.69	-5.57
-25.58	299.96	-5.49
-29.10	222.54	-5.98
-20.02	352.17	-6.18
-21.65	366.91	-6.24

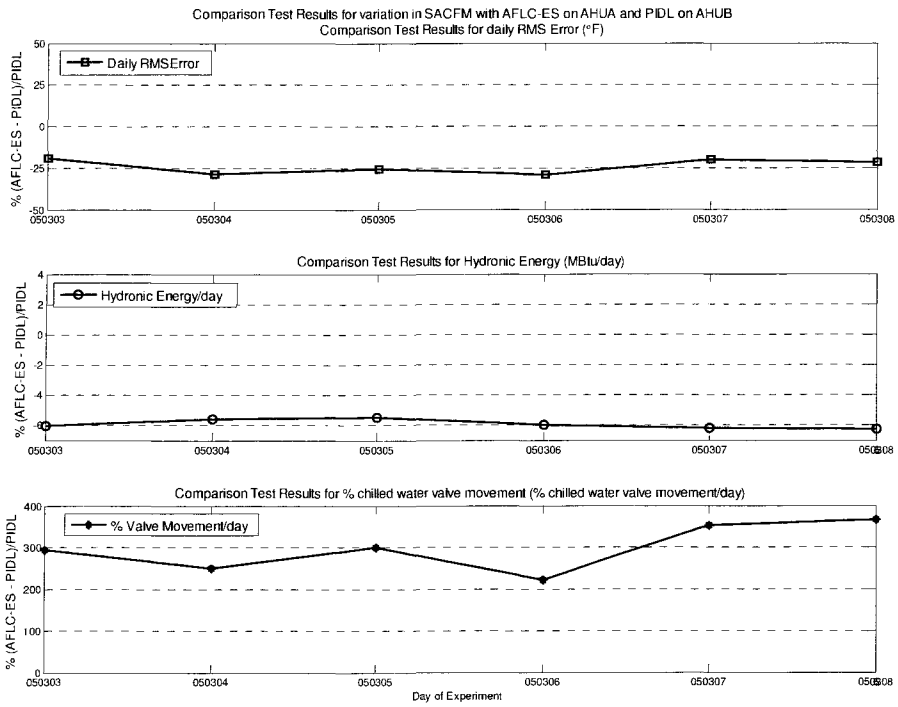

Figure 8.3-13: Percentage difference in performance indices between AHUA (AFLC-ES) and AHUB (PIDL) for variation in SACFM test

8.3.4 Rise Time, Overshoot and Settling Time

Figure 8.3-14 through Figure 8.3-16 shows the SAT response for both the systems. Figure 8.3-14 shows that SATA for system A, controlled by the AFLC-ES, had 0.3°F overshoot whereas SATB for system B, controlled by PIDL, had 0.9°F overshoot. Also, the SAT settling time required for system A controlled by the AFLC-ES controller was approximately 4 time steps less compared to system B controlled by the PIDL controller.

Figure 8.3-15 shows another response of both the systems for a change in SACFM set point from 1800 to 2000 CFM. It was observed that SATA, controlled by AFLC-ES, had 0.3°F overshoot whereas SATB, controlled by PIDL, had 0.9°F. Also, the SAT settling time required for AFLC-ES controller was approximately 21 time steps less compared to the PIDL controller.

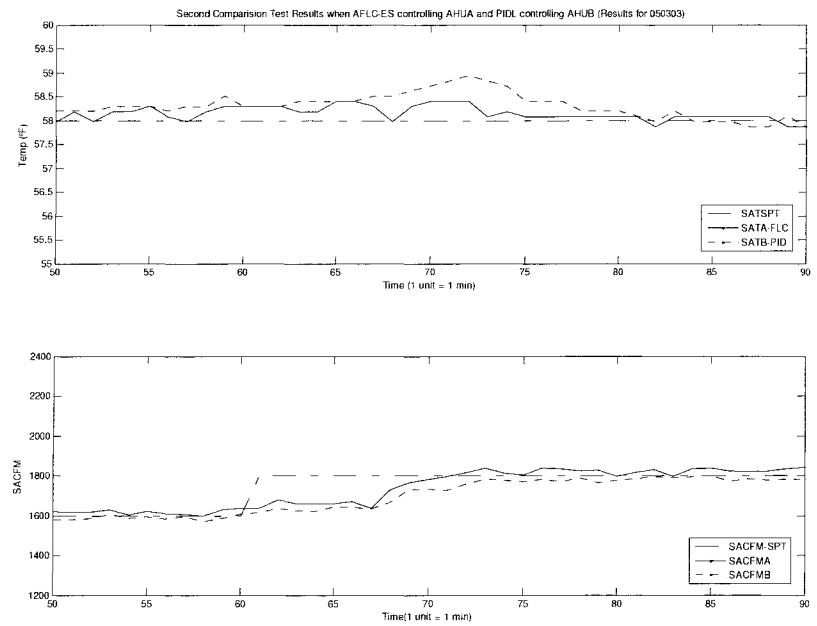

Figure 8.3-14: Case I – Comparison of rise time, overshoot and settling time in SATA (AFLC-ES) and SATB (PIDL) for variation in SACFM test

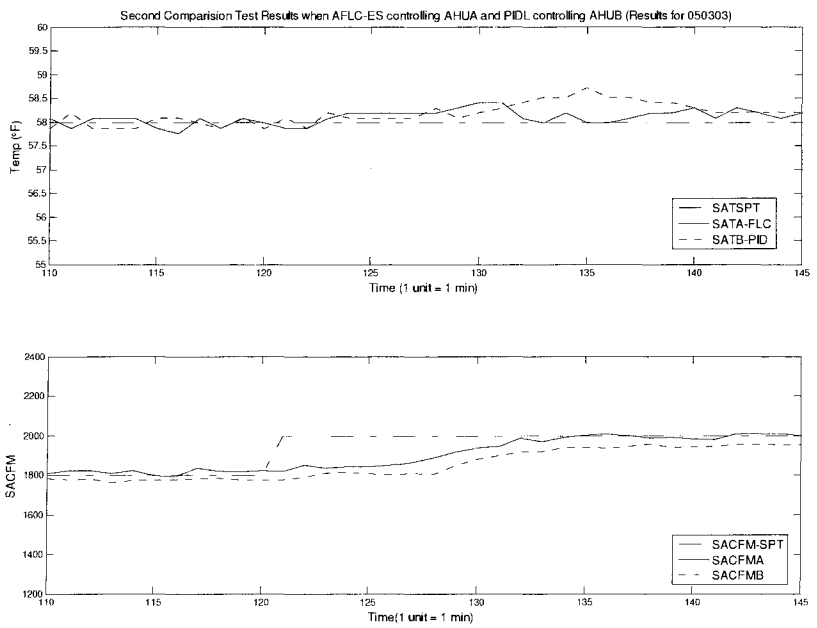

Figure 8.3-15: Case II - Comparison of rise time, overshoot and settling time in SATA (AFLC-ES) and SATB (PIDL) for variation in SACFM test

Figure 8.3-16 shows the response of SATA and SATB for SACFM set point changed from 1600 to 2200 CFM in steps of 200 CFM. It was observed that for changes in the SACFM set point, for all the cases studied, overshoot in SATA for system A was less than 1°F and was always less than that of SATB for system B. Also SAT settling time was always less for the system A, controlled by AFLC-ES, compared to system B, controlled by PIDL.

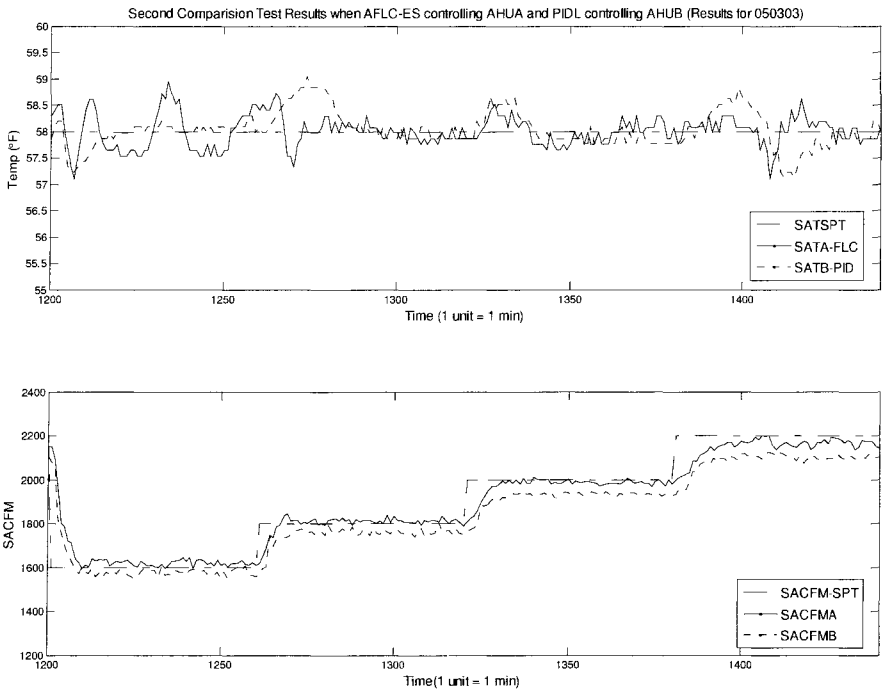

Figure 8.3-16: Case II – Comparison of rise time, overshoot and settling time in SATA (AFLC-ES) and SATB (PIDL) for variation in SACFM test

8.4 Comparison test results for variation in SACFM using AFLC-ES on AHUB

As mentioned in the introduction, one of the various tasks for this study was to periodically switch the AFLC from one air handling unit to the other to eliminate any bias in one unit over the other. In the previous section, results when the AFLC-ES was controlling AHUA were discussed. In this section results when the AFLC-ES is controlling AHUB are discussed.

8.4.1 RMS Error

Table 8.4-1 and Figure 8.4-1 show hourly RMS error for the variation in SACFM comparison test. Most of the hours with AHUB controlled by the AFLC-ES had almost 11% lower RMS error, excluding outliers, as compared to that for AHUA controlled by the PIDL controller. The average hourly RMS error for the 3rd day for AHUB (AFLC-ES) was 0.28°F and for AHUA (PIDL) was 0.31°F

Table 8.4-1: Comparison of average hourly RMS error between SATA (PIDL) and SATB (AFLC-ES) for variation in SACFM test

Comparison test results for variation in SACFM AFLC-ES on AHUB Test Results for 4/2/05			
Average hourly RMS error (°F)			
	Controller		
Hour of a Day	PIDL (AHUA) A	AFLC-ES (AHUB) B	% Difference (B –A)*100/A
1	0.3019	0.2121	-29.75
2	0.1221	0.3434	181.23
3	0.3394	0.3541	4.31
4	0.5770	0.4182	-27.51
5	0.3744	0.2933	-21.66
6	0.1669	0.1795	7.52
7	0.1897	0.1186	-37.50
8	0.4809	0.3496	-27.31
9	0.2456	0.1763	-28.22
10	0.1125	0.1170	3.98
11	0.3184	0.3515	10.38
12	0.5314	0.4103	-22.79

Table 8.4-1: continued

13	0.3915	0.5358	36.84
14	0.2179	0.2036	-6.59
15	0.4389	0.3723	-15.17
16	0.4639	0.4347	-6.30
17	0.3770	0.2855	-24.27
18	0.3164	0.2814	-11.08
19	0.3507	0.3707	5.72
20	0.2109	0.1407	-33.28
21	0.3554	0.3750	5.50
22	0.2045	0.1528	-25.27
23	0.1463	0.1275	-12.87

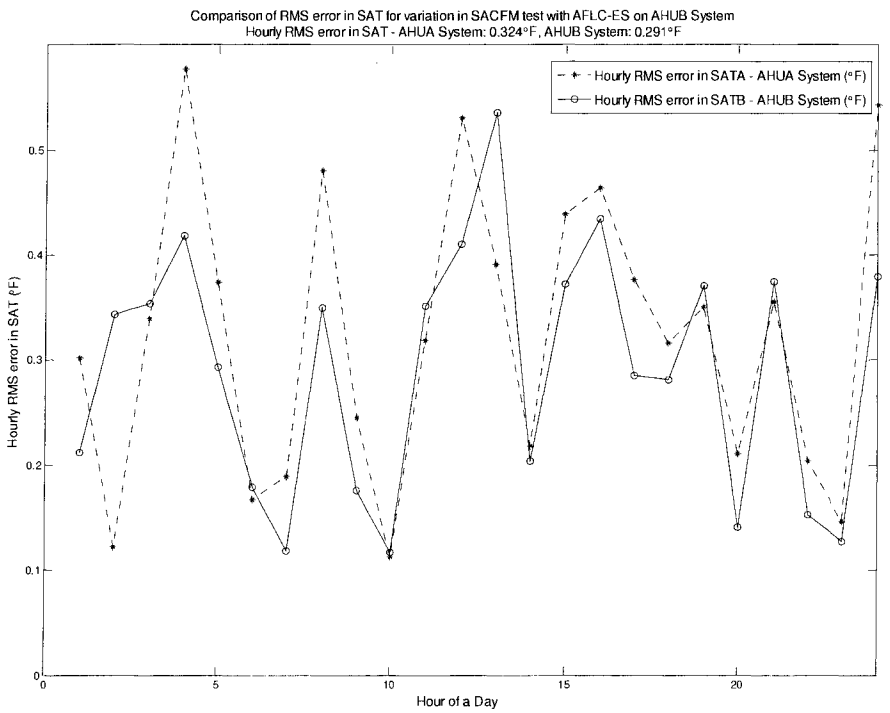

Figure 8.4-1: Comparison of average hourly RMS error between SATA (PIDL) and SATB (AFLC-ES) for variation in SACFM test

8.4.2 Hydronic Energy

Table 8.4-2 and Figure 8.4-2 show the average hydronic energy/hr consumption. Most of the hours with AHUB controlled by the AFLC-ES had almost 0.8% more hydronic energy/hr consumption as compared to that for AHUA controlled by the PIDL controller. The average hydronic energy for the 3rd day for the AHUA (PIDL) was 0.096 MBTU/hr and 0.097 MBTU/hr for AHUB (AFLC-ES).

Table 8.4-2: Comparison of average hydronic energy/hr consumption between AHUA (PIDL) and AHUB (AFLC-ES) for variation in SACFM test

Comparison test results for variation in SACFM AFLC-ES on AHUB Test Results for 4/2/05			
Average Hydronic Energy/hr (MBTU/hr)			
	Controller		
Hour of a Day	PIDL (AHUA) A	AFLC-ES (AHUB) B	% Difference (B -A)*100/A
1	0.1036	0.1024	-1.16
2	0.0973	0.0966	-0.63
3	0.0933	0.0955	2.38
4	0.0930	0.0949	2.05
5	0.1030	0.1007	-2.20
6	0.0984	0.0997	1.32
7	0.0904	0.0917	1.44
8	0.0902	0.0919	1.94
9	0.1023	0.1003	-1.94
10	0.0969	0.0974	0.42
11	0.0916	0.0942	2.87
12	0.0913	0.0938	2.69
13	0.0963	0.0958	-0.47
14	0.0959	0.0976	1.75
15	0.0985	0.0995	0.98

Table 8.4-2: continued

16	0.0996	0.0998	0.18
17	0.1031	0.1022	-0.87
18	0.0985	0.1006	2.14
19	0.0952	0.0987	3.67
20	0.0887	0.0902	1.67
21	0.1021	0.1000	-2.10
22	0.0984	0.0986	0.16
23	0.0899	0.0915	1.79

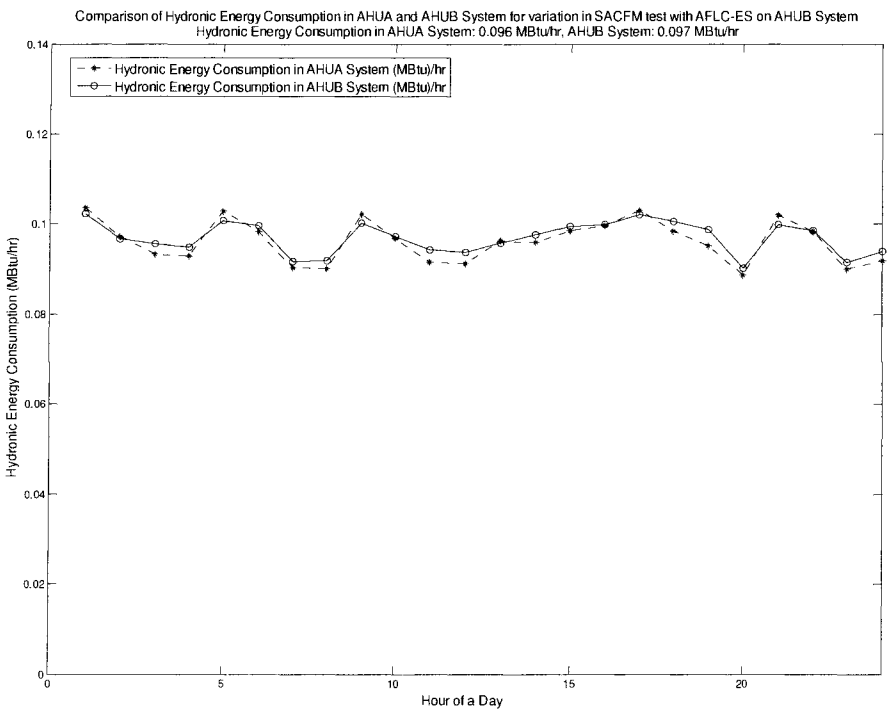

Figure 8.4-2: Comparison of average hydronic energy/hr consumption between AHUA (PIDL) and AHUB (AFLC-ES) for variation in SACFM test

8.4.3 Actuator Travel Distance

Table 8.4 3 and Figure 8.4 3 show the total actuator travel distance/hr. AHUB controlled by the AFLC-ES had approximately 21.5% more valve movement than that for AHUA controlled by the PIDL controller. The average of total actuator travel distance for the 3rd

day for AHUB (AFLC-ES) was 34.2% Valve Movement/hr and for AHUA (PIDL) was27.8% Valve Movement/hr.

Table 8.4-3: Comparison of total actuator travel distance/hr between chilled water value on AHUA (PIDL) and AHUB (AFLC-ES) for variation in SACFM test

Comparison test results for variation in SACFM AFLC-ES on AHUB Test Results for 4/2/05			
Average % Chilled Water Valve Movement/hr			
	Controller		
Hour of a Day	PIDL (AHUA) A	AFLC-ES (AHUB) B	% Difference (B –A)*100/A
1	64.7260	62.4080	-3.58
2	14.1820	15.6130	10.09
3	26.8920	34.0990	26.80
4	43.2020	49.0300	13.49
5	31.3520	35.2050	12.29
6	19.0030	18.1550	-4.46
7	19.4970	12.8530	-34.08
8	37.9520	43.0560	13.45
9	21.4610	26.1230	21.72
10	12.9170	12.7040	-1.65
11	29.2750	29.2690	-0.02
12	41.6530	49.4580	18.74
13	29.6900	69.5760	134.34
14	22.8290	21.5860	-5.44
15	35.7220	51.7340	44.82
16	33.3120	45.8770	37.72
17	29.9360	51.3990	71.70
18	23.6130	40.8050	72.81
19	27.7940	24.1590	-13.08
20	15.3290	16.8640	10.01
21	27.7580	47.6050	71.50

Table 8.4.3: continued

22	16.7840	12.7930	-23.78
23	13.8220	16.5070	19.43

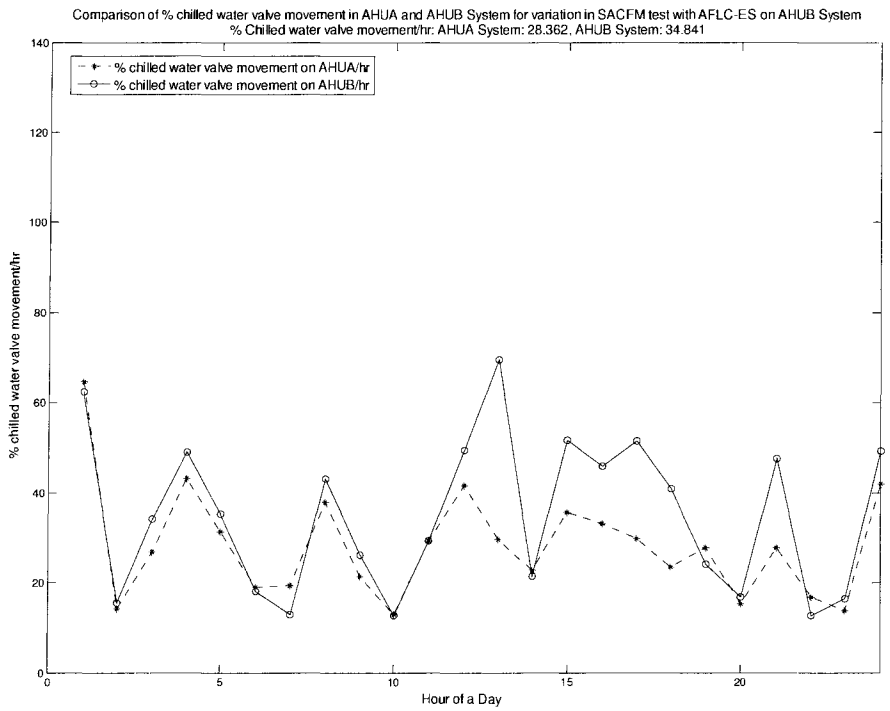

Figure 8.4-3: Comparison of total actuator travel distance/hr between chilled water value on AHUA (PIDL) and AHUB (AFLC-ES) for variation in SACFM test

Similarly the calculations for all six days were done when the AFLC-ES was controlling AHUB and PIDL on AHUA. Table 8.4-4 shows different performance indices for both the controller. Table 8.4-5 and Figure 8.4-4 shows the percentage difference between the performance indices for AFLC-ES and PIDL controller.

Table 8.4-4: Comparison of performance indices between AHUA (PIDL) and AHUB (AFLC-ES) for variation in SACFM test

Comparison test results for variation in SACFM						
AFLC-ES on AHUB						
	Daily Average RMS error (°F)		% Chilled water valve movement		Hydronic Energy (MBtu/hr)	
ExpData	AHUA (PIDL)	AHUB (AFLC-ES)	AHUA	AHUB	AHUA	AHUB
50331	0.31	0.42	26.56	43	0.096	0.0966
50401	0.33	0.28	29.64	35.97	0.0957	0.0968
50402	0.32	0.29	28.36	34.84	0.096	0.097
50403	0.33	0.28	28.69	38.68	0.096	0.0976
50404	0.29	0.27	25.82	34.93	0.0968	0.0982
50504	0.37	0.34	30.77	37.69	0.0967	0.0977

Table 8.4-5: Percentage difference in performance indices between AHUA (PIDL) and AHUB (AFLC-ES) for variation in SACFM test

Comparison test results for variation in SACFM AFLC-ES on AHUB		
% Percentage (AFLC-ES -PIDL)/PIDL		
RMS error	% Chilled water valve Movement	Hydronic Energy
32.54	61.92	0.59
-15.66	21.38	1.10
-8.71	23.20	0.75
-14.86	34.82	1.65
-8.68	35.30	1.48
-9.14	22.52	1.01

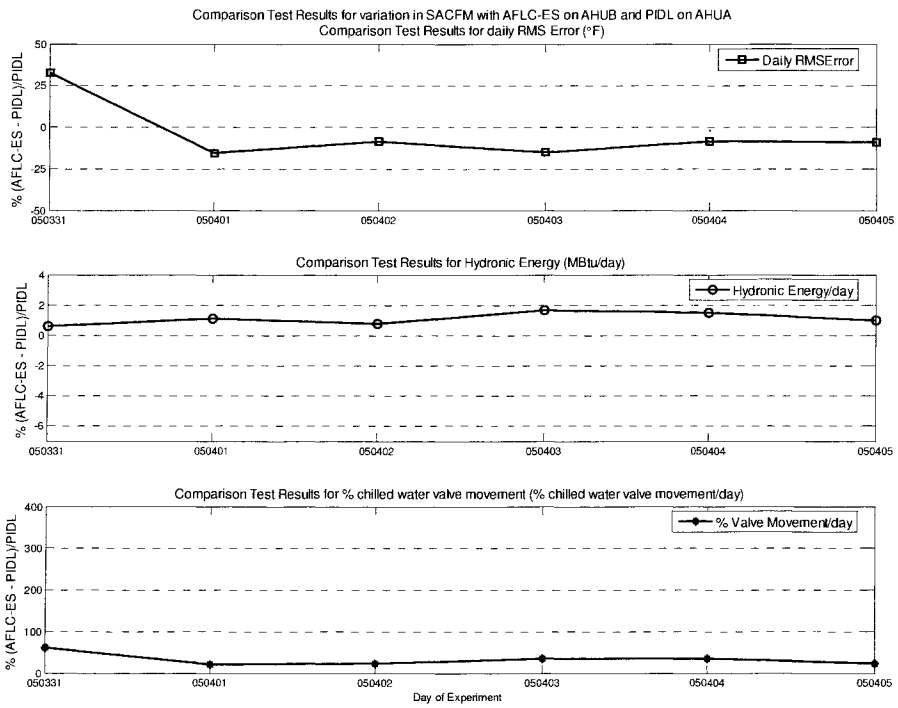

Figure 8.4-4: Percentage difference in performance indices between AHUA (PIDL) and AHUB (AFLC-ES) for variation in SACFM test

Table 8.4-6: Percentage difference in hydronic energy/day consumption for variation in SACFM test with AFLC-ES on AHUA and AFLC-ES on AHUB compared with PIDL

Percentage hydronic energy saved by AFLC-ES		
Test Day	Using AFLC-ES on AHUA	Using AFLC-ES on AHUB
1	-6.06	0.59
2	-5.57	1.10
3	-5.49	0.75
4	-5.98	1.65
5	-6.18	1.48
6	-6.24	1.01
Average	-5.92	1.10

From Table 8.4-6, it was found that when the AFLC-ES was controlling System A, it consumed almost 5.9% less hydronic energy and when controlling System B, it consumed almost 1.1% more hydronic energy as compared with other system when controlled by PIDL controller.

8.4.4 Rise Time, Overshoot and Settling Time

Figure 8.4-5 shows the response of SATA and SATB for SACFM SPT changed from 1500 to 2100 CFM in steps of 200 CFM. It was observed that for the changes in SACFM SPT, for all the cases, SATB, controlled by AFLC-ES, had overshoot less than 1°F and was always less than that of SATA, controlled by the PIDL. Also SAT settling time was always less for the system B, controlled by AFLC-ES, compared to system A, controlled by PIDL.

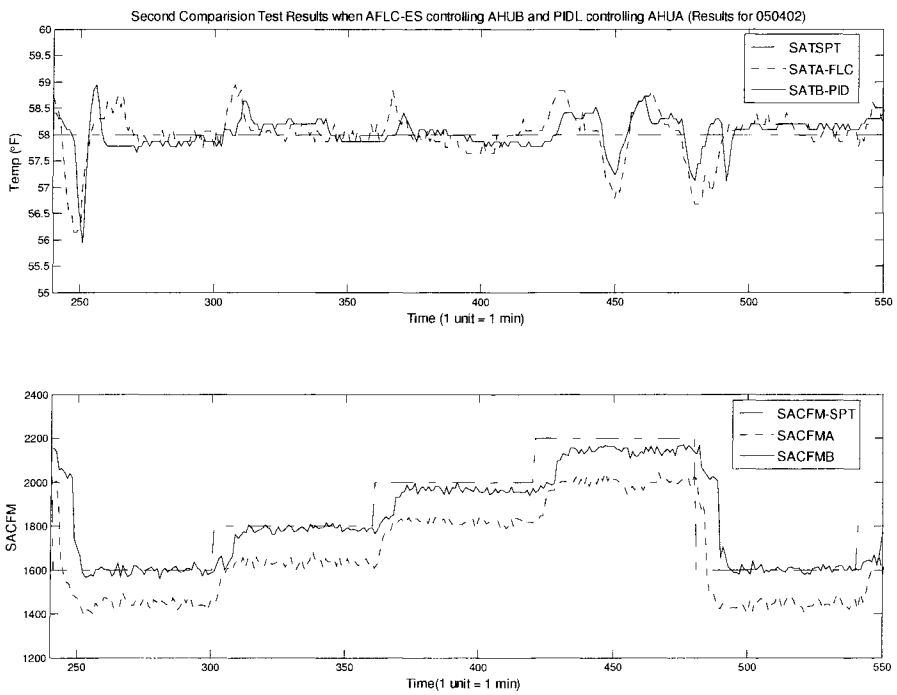

Figure 8.4-5: Comparison of rise time, overshoot and settling time in SATB (AFLC-ES) and SATA (PIDL) for variation in SACFM test

8.5 Summary

From the first comparison test results, in which the supply air setpoint was cyclically changed between 52°F and 58°F, it was observed that for most of the cycles, the AHUA controlled by the AFLC-GA had higher RMS error when compared to AHUB controlled by the PIDL controller, but it was decreasing with the number of cycles as it adapted to operating conditions. If the experiment would have been continued, RMS error/cycle for AHUA would probably have decreased further. Also, it was observed that the AFLC-GA controlling system A consumed 1 to 7% less hydronic energy/cycle than for system B controlled by the PIDL. For actuator travel distance, the ATD/cycle for AHUA, controlled by an AFLC-GA, was approximately 1 to 140% more than for AHUB, controlled by PIDL controller.

From the control performance view point, the first comparison test which involved cyclic change in SATSPT between 52°F to 58°F, both of the controllers were able to control the process within +/- 1.5°F. It was observed that even though the controlled variable for PIDL had less rise time, it also had more overshoot as compared to an AFLC-GA. Also, the SAT settling time for the PIDL was more than for the AFLC-GA.

From the second comparison test results, in which SATSPT was held constant at 58°F and SACFM was cyclically varied from 1600 to 2200 CFM in steps of 200 after every hour, the root mean square error between SAT and SATSPT, was 8 to 30% less for both A and B systems when the AFLC-ES was controlling them compared to when they were controlled by the PIDL. When the AFLC-ES was controlling AHUA and the PIDL on AHUB, considering the bias between the systems, the system using the AFLC-ES consumed 5 to 7% less hydronic energy compared to the system using the PIDL controller. When the AFLC-ES was controlling AHUB and the PIDL was on AHUA, the system using the AFLC-ES consumed 0 to 2% more hydronic energy than the system using the PIDL controller. It was found that AFLC-ES caused more actuator travel distance (ATD) than the PIDL control. When the AFLC-ES was controlling system A, ATD was 220 to 400% more than for PIDL control.

When the AFLC-ES was controlling system B, ATD was 20 to 35% more. The feasibility and potential to save the energy for the adaptive FLC were demonstrated by these tests.

From the control performance view point, for variation in SACFM comparison test which involved cyclic changes in SACFM between 1600 to 2200 cfm in steps of 200 cfm, both the controllers were able to control the process within +/- 1°F. It was observed that for changes in the SACFM set point, for all the cases studied, overshoot in the controlled variable for AFLC-ES was less than 1°F and was always less than that of the system controlled by PIDL. Also the SAT settling time was less for the system controlled by AFLC-ES, compared to system controlled by PIDL.

Chapter 9 Conclusions

9.1 Details

The objective of this study has been to develop, implement, and test an adaptive fuzzy logic controller on one of the air handling units at the IEC Energy Resource Station (ERS). This was accomplished by the following tasks:

1. Develop and implement a manually tuned fuzzy logic controller (FLC) at the ERS
2. Compare the performance of the FLC to current control strategies
3. Develop and implement an adaptive FLC (AFLC) at the ERS
4. Compare the performance of the AFLC to current control strategies

A FLC was developed and implemented on a chilled water coil to control the chilled water valve position to maintain the supply air temperature (SAT) at its set point (SATSPT) value. Using a default fuzzy rule matrix, the FLC was able to control the SAT, but more tuning was necessary to obtain quicker response to SATSPT and environmental changes and to have minimal overshoot of the controlled variables, thus achieving an optimal performance.

Lump Capacitance Models (LCM), Neural Network (NN) models and General Regression Neural Network (GRNN) models of the system were developed to help offline tuning of the FLC. The LC and GRNN models were used for developing an adaptive FLC.

The LCMs were developed based on the basic principles of heat and mass transfer and the best LCM was able to predict the SAT within 2°F. The GRNN-II model developed with 18 inputs, including 10 time steps of previous information for the SAT, is able to predict the SAT within 1°F. Two other models, the NN and GRNN-I, were able to predict the SAT within 4°F.

Genetic Algorithms (GA) and Evolutionary Strategies (ES) were used to develop an adaptive FLC. The GA technique was used to modify the fuzzy rule matrix and ES technique was

used to modify fuzzy membership relationships in real-time experiments at the ERS. Improvements in the FLC performance were achieved, demonstrating that adaptive FLCs could be developed for HVAC applications.

Two types of comparison tests were performed, the first with step changes in SATSPT between 52°F and 58°F and the second with cyclic variations in supply air flow rate (SACFM) for fixed SATSPT at 58°F. Both of the comparison tests were conducted for at least seven consecutive days without manual alteration of any FLC parameters. The FLC was used to control one of the A or B AHUs while the PIDL was used to control the other. Both of the systems had identical thermal loads.

For the variation in SATSPT comparison tests using GA, the fuzzy rule matrix was adapted to achieve an optimal performance of an AFLC-GA controller. Depending on whether the AFLC-GA was controlling AHUA and the PIDL was on AHUB, or vice versa, the AHU using the AFLC-GA consumed 1 to 7% less hydronic energy than the AHU using a PIDL controller. Large variations in the percentage of hydronic energy consumption for the AHU using AFLC-GA controlling were observed since it was still adapting. When the AFLC-GA was controlling AHUA, initially larger RMS errors between SAT and SATSPT were observed compared to RMS errors on AHUB controlled by PIDL controller. But the RMS error for AHUA, controlled by the AFLC-GA, was decreasing. It was found that AFLC-GA caused more actuator travel distance (ATD) than the PIDL control. When AFLC-GA was controlling AHUA, ATD was 10 to 140% more. It was also observed that the controlled variable for the AFLC-GA system required 0 to 185% more rise time, had 9 to 68% less overshoot and required 11 to 45% less settling time as compared to the PIDL controlled system.

For the variation in SACFM comparison tests using ES, the fuzzy membership functions were modified to achieve optimal performance of an AFLC-ES. When the AFLC-ES was controlling AHUA, the RMS error between supply air temperature (SAT) and supply air temperature set point (SATSPT), was 19 to 30% less compared to AHUB controlled by the

PIDL. When AFLC-ES was controlling AHUB, the RMS error was 9 to 15% less compared to AHUA controlled by the PIDL. When the AFLC-ES was controlling AHUA and the PIDL on AHUB, the system using the AFLC-ES consumed 5 to 7% less hydronic energy than the system using the PIDL controller. On the other hand, when the FLC was on AHUB and the PIDL was on AHUA, the system using an AFLC-ES consumed 1 to 2% more hydronic energy than the system using the PIDL controller. When an AFLC-ES was controlling either AHUA or AHUB, the RMS error between SAT and SATSPT was less than the PIDL controller. It was also found that the AFLC-ES caused more actuator travel distance (ATD) than for the PIDL control. When the AFLC-ES was controlling AHUA, ATD was 220 to 400% more and when an AFLC-ES was controlling AHUB, ATD was 20 to 35% more. It was also observed that the controlled variable for the AFLC-ES when controlling AHUA had 40 to 60% less overshoot and required 45 to 75% less settling time as compared to AHUB controlled by PIDL controller. It was also observed that the controlled variable for the AFLC-ES when controlling AHUB had 0 to 30% less overshoot and required 13 to 25% more settling time as compared to AHUA controlled by PIDL controller.

9.2 Contributions

The contributions of this study include:

1. Development and implementation of an FLC for controlling the chilled water valve position on real air handling units;
2. Development and validation of simple Lump Capacitance, Neural Network, and General Regression Neural Networks cooling coil models that do not require extensive information about the HVAC systems;
3. Development and implementation of an AFLC in a commercial building by using Genetic Algorithms and Evolutionary Strategies to modify fuzzy rule matrix; and fuzzy membership functions; and
4. Experimentally comparing the performance of the AFLC with a standard PIDL controller.

9.3 Recommendations

The findings of this study provide a better understanding about adaptive fuzzy logic controllers, including the experimental validation and analysis of test data to determine how the AFLC performs when it is applied to a real HVAC system. Different cooling coil models were also investigated. Because the AFLC shows the potential of saving system energy, more research is needed in the future to enhance the capabilities of the AFLC. Some recommendations for the future study include:

1. More normalization tests are needed to define the bias between the A- and B-Test Systems at the ERS from a statistical viewpoint.
2. More experiments need to be conducted when both the systems are using AFLC;
3. Additional studies are needed to examine the modeling of the AHU cooling coil;
4. For the cooling coil model, moisture effects should be included to allow the model to be applicable to a wider range of conditions;
5. Application of data clustering for training the neural network cooling coil models should be studied;
6. More studies for updating techniques for neural networks, used in the cooling coil model, are needed;
7. To deal with missing and erroneous data, an auto associative neural network should be studied;
8. Conduct more validation tests without 100% recirculation of return air to compare the AFLC with the PIDL; and
9. More validation tests are needed to compare the AFLC with the PIDL.

Appendix A Energy Resource Station (ERS) Air Handling Unit (AHU) and Cooling Coil Specifications

Table A-1: AHU Specification

AHU Equipment Specification				
Component	Capacity			
	AHUA and AHUB			
	Flow Rate	Speed	Static Pressure	Power
	Cfm	rpm	psi	bhp
Supply Fan	3200	1834	1.75	2.98
Return Fan	3200	1257	1.5	1.21

Table A-2: Cooling Coil Specifications

Cooling Coil Parameters AHUA and AHUB	
Parameters	Value
Number of rows	6
Number of tubes per row	18
Tube Material	1/2 in. std copper
Tube Enhancement	internal
	turbulators
Fin Type	wavy plate
Fin Material	aluminum
Fin Spacing, fins/ft	115
Fin Thickness, in	0.0075

Table A-3: ERS data points and accuracy for the sensors used

Data point	Point Name	Units	Accuracy
chilled water flow rate	CHWPGPM	gpm	± 0.09 gpm (0-18 gpm)
			± 0.5% rdg (18-180 gpm)
			plus ± 0.03 gpm
chilled water pump power	CHWPWAT	W	± 0.2% reading
cooling coil discharge air temp.	CHWCDAT	°F	± 0.25°F
cooling coil entering water temp.	CHWCEWT	°F	± 0.25°F
cooling coil leaving water temp.	CHWCLWT	°F	± 0.25°F
cooling coil mixed water temp.	CHWCMWT	°F	± 0.25°F
cooling valve control	CHWCVLV	% open	n/a
duct static pressure	SASP	in. W.G.	± 0.025 in. W.G.
duct static pressure set point	SASPSPT	in. W.G.	n/a
exhaust air damper position	EADP	% open	n/a
heating coil discharge air temp.	HWCDAT	°F	± 0.25°F
mixed air temperature	MAT	°F	± 0.25°F
outdoor air damper position	OADP	% open	n/a
outdoor air flow rate	OACFM	cfm	± 2% of rdg (> 500 fpm)
			± 10 fpm (< 500 fpm)
recirculated air damper position	RADP	% closed	n/a

Table A-3: continued

return air flow rate	RACFM	cfm	± 2% of rdg (> 500 fpm)
			± 10 fpm (< 500 fpm)
return air humidity	RAHUMD	%RH	± 2% RH (0-90%RH)
			± 3% RH (90-100%RH)
return air temperature	RAT	°F	± 0.18°F
return fan differential pressure	RFDP	in. W.G.	± 0.025 in. W.G.
return fan differential pressure	RFDP	in. W.G.	± 0.025 in. W.G.
return fan power	RFWAT	W	± 0.2% reading
return fan speed	RFSPD	% Speed	n/a
return fan vfd start/stop	RFSST	start/stop	n/a
return fan vfd status	RFSTS	on/off	n/a
return vfd alarm	RFALM	normal/alarm	n/a
sum of zone air flow rate	RMTCFM	cfm	n/a
supply air flow rate	SACFM	cfm	± 2% of rdg (> 500 fpm)
			± 10 fpm (< 500 fpm)
supply air humidity	SAHUMD	%RH	± 2% RH (0-90%RH)
			± 3% RH (90-100%RH)
supply air temp. set point	SATSPT	°F	n/a
supply air temperature	SAT	°F	± 0.18°F
supply fan differential pressure	SFDP	in. W.G.	± 0.025 in. W.G.
supply fan power usage	SFWAT	W	± 0.2% reading
supply fan speed	SFSPD	% speed	n/a

Table A-3: continued

supply fan vfd start/stop	SFSST	start/stop	n/a
supply fan vfd status	SFSTS	on/off	n/a
supply vfd alarm	SFALM	normal/alarm	n/a

Appendix B DDE (Matlab, 2001)

B.1 DDE Concepts and Terminology

Applications communicate with each other by establishing a DDE conversation. The application that initiates the conversation is called the client. The application that responds to the client application is called the server.

When a client application initiates a DDE conversation, it must identify two DDE parameters that are defined by the server:

- The name of the application it intends to have the conversation with, called the service name.
- The subject of the conversation, called the topic.

When a server application receives a request for a conversation involving a supported topic, it acknowledges the request, establishing a DDE conversation. The combination of a service and a topic identifies a conversation uniquely. The service or topic cannot be changed for the duration of the conversation, although the service can maintain more than one conversation. During a DDE conversation, the client and server applications exchange data concerning items. An item is a reference to data that is meaningful to both applications in a conversation. Either application can change the item during a conversation.

B.2 DDE Advisory Links

You can use DDE to notify a client application when data at a server has changed. For example, if you use MATLAB to analyze data entered in an Excel spreadsheet, you can establish a link that causes Excel to notify MATLAB when this data changes. You can also establish a link that automatically updates a matrix with the new or modified spreadsheet data.

MATLAB supports two kinds of advisory links, distinguished by the way in which the server application advises MATLAB when the data that is the subject of the item changes at the server:

- A hot link causes the server to supply the data to MATLAB when the data defined by the item changes.
- A warm link causes the server to notify MATLAB when the data changes but supplies the data only when MATLAB requests it.

You set up and release advisory links with the ddeadv and ddeunadv functions. MATLAB supports links only when MATLAB is a client.

Different DDE functions used in program are:

- ddeadv Set up advisory link
- ddeinit Initiate DDE conversation
- ddepoke Send data to application
- ddereq Request data from application
- ddeunadv Release advisory link

Appendix C Fuzzy Logic (Driankov, 1993)

Fuzzy logic (FL) is about the relative importance of precision. It is considered to be a mathematical logic system which is aimed at providing a model for modes of human reasoning that are approximate rather than exact. FL is a tool for embedding human knowledge into workable algorithms. It is a convenient way to map an input universe of discourse to an output universe of discourse.

FL is a rule base that consists of a collection of fuzzy IF-THEN rules. The fuzzy inference engine uses these rules to determine a mapping from the fuzzy sets in the input universe of discourse to fuzzy sets in the output universe of discourse based on the fuzzy principles. An ordinary set membership function takes only two values {0, 1} and is considered to be a crisp set. A fuzzy set membership function can take any real value from 0 to 1 and can closely represent human thinking.

This part of report is divided into:

- Fuzzy Sets and Logical Operators
- Membership Functions
- Fuzzy Rule Base
- Fuzzy Inference System
- Fuzzification
- Defuzzification
- Centroid Defuzzification Method

C.1 Fuzzy Sets and Logical Operators

Fuzzy set operations are analogous to crisp set operations. The important thing in defining fuzzy set logical operators is that if we keep fuzzy values to the extremes i.e. 1 (True) or 0 (False), the standard logical operations should hold true. In order to define fuzzy set logical operators, let us first consider crisp set operators. The most elementary crisp set operations

are union, intersection, and complement, which essentially correspond to OR, AND, and NOT operators, respectively.

Let A and B be two subsets of U. The union of A and B, denoted $A \cup B$, contains all elements in either A or B; that is $\mu_{A \cup B}(x) = 1$, if x ∈ A or x ∈ B. The intersection of A and B, denoted $A \cap B$, contains all the elements that are simultaneously in A and B; that is, $\mu_{A \cap B}(x) = 1$, if x ∈ A and x ∈ B. The complement of A is denoted by $\bar{A}$, and it contains all elements that are not in A; that is $\mu_{\bar{A}}(x) = 1$, if x ∉ A and $\mu_{\bar{A}}(x) = 0$, if x ∈ A. The truth tables for these operators are shown in Table C-1.

Table C-1: Truth tables for AND, OR, and NOT operators.

AND			OR			NOT	
A	B	$A \cap B$	A	B	$A \cup B$	A	$\bar{A}$
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

In FL, the truth of any statement is a matter of degree. In order to define FL operators, we have to find the corresponding operators that preserve the results of statement using AND, OR, and NOT operators. The answer is min, max, and complements operations respectively. These operators are defined, respectively, as

$$\mu_{A \cup B}(x) = \max[\mu_{(A)}, \mu_{(B)}]$$
$$\mu_{A \cap B}(x) = \min[\mu_{(A)}, \mu_{(B)}]$$
$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

The formulas for AND, OR, and NOT operators in the above Equations are useful for proving the other mathematical properties about sets; however, min and max are not the only ways to describe the intersection and union of two sets.

Let U be a collection of objects and be called the universe of discourse. A Fuzzy Set F in U is characterized by a membership function μF: U-- [0, 1], with μF(u) representing the grade of membership of u Є U in the fuzzy set F. The curve which represents the grade of membership is called a Membership Function.

C.2 Membership Functions

A Membership Function is a curve that defines how each point in the universe of discourse (input space) is mapped to a membership value (or degree of membership) between 0 and 1.The only condition which a membership function must satisfy is that it must vary from 0 to 1. The function itself can be an arbitrary curve whose shape is user defined as a function that suits the application from the point of view of simplicity, convenience, speeds and efficiency. Various types of membership functions are used, including triangular, trapezoidal, generalized bell shaped, Gaussian curves, polynomial curves, and sigmoid functions. Triangular curves depend on three parameters a, b, and c and are given by

$$\mu(x) = \begin{cases} 0 & \text{if } x < a \\ \dfrac{x-a}{c-a} & \text{if } x \in [a,c] \\ \dfrac{b-x}{b-c} & \text{if } x \in [c,b] \\ 0 & \text{if } x > b \end{cases},$$

... C-1

A plot of a Triangular Membership Function is shown in Figure C-1

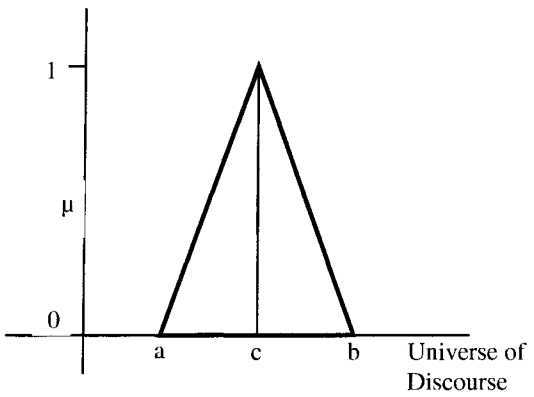

Figure C-1: Triangular Membership Functions

To use this fuzzy logic system for practical problems, where inputs and outputs are real valued variables, a fuzzifier is added to the inputs and a defuzzifier is added to the output. The basic configuration of FL System is shown in Figure C-2.

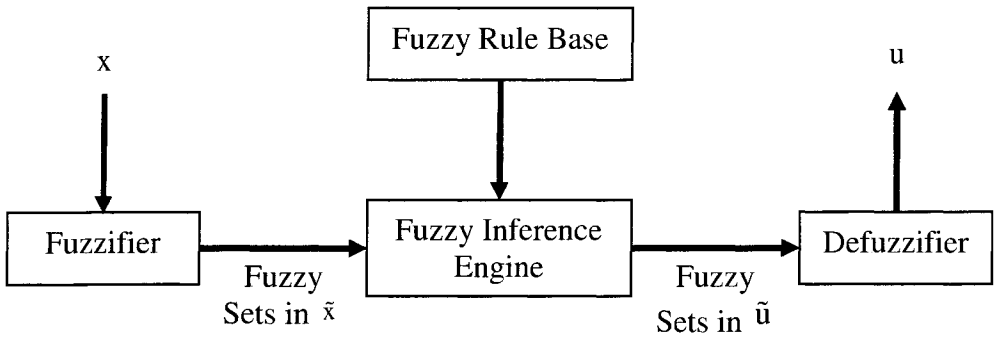

Figure C-2: Basic configuration of Fuzzy Logic System

The fuzzifier maps real variables in x to fuzzy sets in $\tilde{x}$ and the defuzzifier maps fuzzy sets in $\tilde{u}$ to real variables u.

C.3 Fuzzy Rule Base

A Fuzzy Rule Base consists of a collection of fuzzy IF-THEN rules in the following form:

R l: IF x1 is Fl1 and ....and xn is Fnl then y is Gl1

where,

x1..... xn are the inputs and y are the outputs from the FL system, respectively.

Fl1 and Gl1 are fuzzy sets in U and V, respectively.

Let M be the number of fuzzy IF-THEN rules in the form of the above equation in the fuzzy rule base, l = 1, 2 ....M. The Fuzzy Rule Base is the core of the FL system used to interpret these rules and make them usable for specific problems.

C.4 Fuzzy Inference System

Fuzzy inference is the process of formulating the mapping from a given input to an output using FL. The mapping then provides a basis from which decisions can be made. The process

of fuzzy inference involves use of membership functions, fuzzification, fuzzy rule base, aggregation of output sets, and defuzzification.

C.5 Fuzzification

The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. The input is always a real numerical value limited to the universe of discourse of the input variable and the output is a fuzzy degree of membership in the qualifying fuzzy sets.

C.6 Defuzzification

Several methods for defuzzification are used in practice, including the centroid, maximum, mean of maxima, height, and modified height defuzzifiers. The most popular defuzzification method is the centroid method, which calculates and returns the center of gravity of the aggregated fuzzy set.

A fuzzy inference system maps an input vector to a real output value. In order to obtain a real output, we need a defuzzification process. The input to the defuzzification process is a fuzzy set (the aggregated output fuzzy set), and the output of the defuzzification process is a single number. For the current project, the centroid defuzzification method is used for defuzzification.

C.7 Centroid Defuzzification Method

In this method, the defuzzifier determines the center of gravity (centroid) of membership values and uses that value as the output of the FL System. For an aggregated fuzzy set, the centroid is given by

$$y' = \frac{\int_{y} y_i \mu_B(y) dy}{\int_{y} \mu_B(y) dy} \qquad \text{... C-2}$$

where, y -- [0,1] Often, discrete variables are used so that y' can be approximated using summations instead of integration.

$$y' = \frac{\sum_{i=1}^{n} y_i \mu_B(y) dy}{\sum_{i=1}^{n} \mu_B(y) dy} \quad \text{... C-3}$$

The centroid defuzzification method finds the "balance" point of the solution fuzzy region by calculating the weighted mean of the output fuzzy region. It is the most widely used technique because, when it is used, the defuzzified values tend to move smoothly around the output fuzzy region. The technique is unique, however, and not easy to implement computationally. The method of centroid defuzzification is illustrated in Figure C-3.

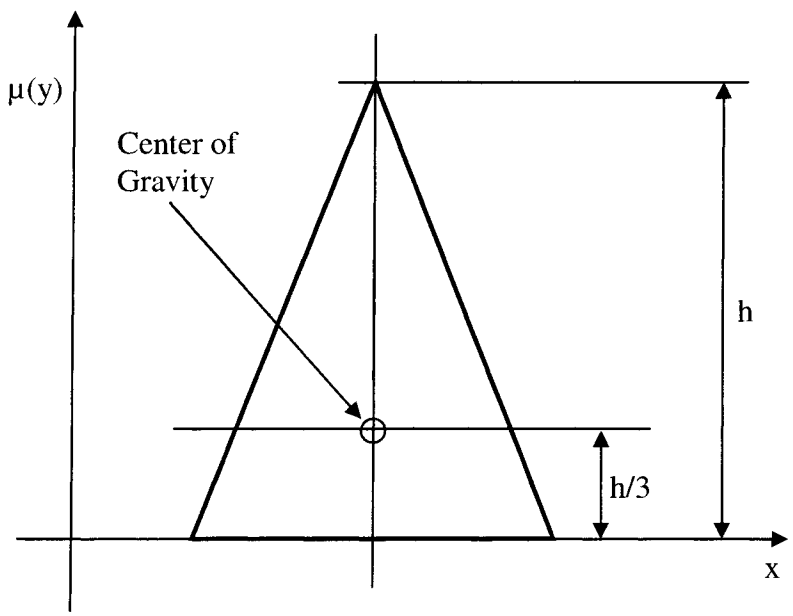

Figure C-3: Centroid Defuzzification Method

C.8 Example for Fuzzification Process

Fuzzification is the process by which a real value is converted into membership values of the fuzzy set. This maps every real value to memberships from 0 to 1 in the elements of the fuzzy set.

In order to explain fuzzification process, considered membership functions for error, derivative of error and control signal as shows in Figure C-4, Figure C-5, and Figure C-6 respectively and data shown in

Table C-2.

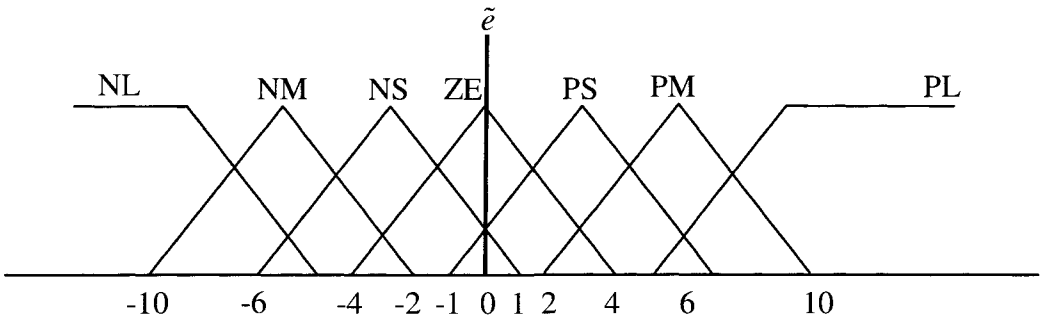

Figure C-4: Membership Functions for Error

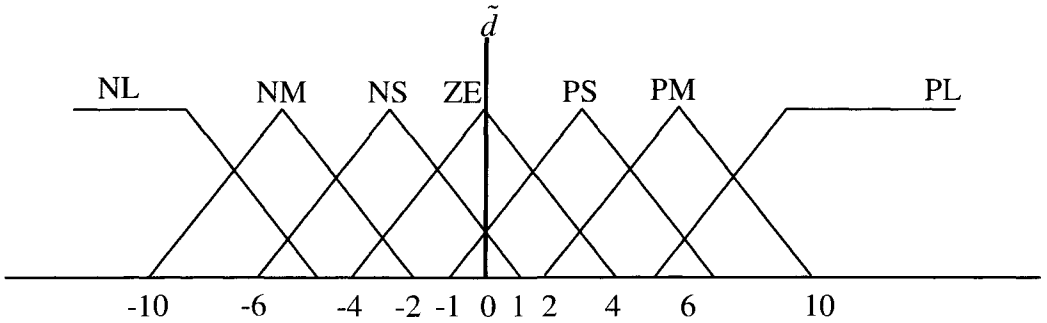

Figure C-5: Membership Functions for derivative of error

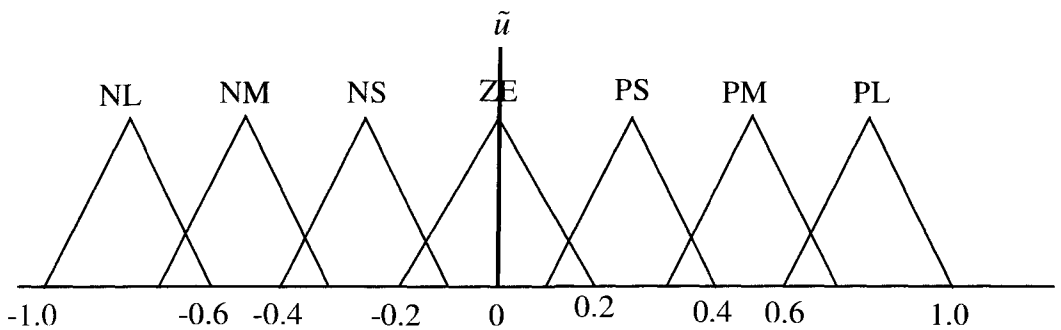

Figure C-6: Membership Functions for Control Signal

Table C-2: Sample Data from the ERS AHUA

Sr. No.	SAT	SATSPT	Error	Derivative of Error	Control Signal	Chilled water valve position
	°F	°F	°F	°F/Sampling Period (sec)	% Change in valve position	% Open
1	62.08	60	2.08	0.11	0.21	61.52
2	61.86	60	1.86	-0.22	-0.43	61.1
3	62.08	60	2.08	0.22	0.44	61.54
4	61.86	60	1.86	-0.22	-0.44	61.1
5	62.08	60	2.08	0.22	0.44	61.54

SAT is set at 60°F and current SA Temp is 61.86 °F. So the error is 1.86 degree F. Then using the membership functions for error, the approximate values for memberships in the error fuzzy set are:

NL = Negative Large = 0.0

NM = Negative Medium = 0.0

NS = Negative Small = 0.0

ZE = Zero = 0.65

PS = Positive Small = 0.75

PM = Positive Medium = 0.0

PL = Positive Large = 0.0

Using information from the previous time steps provided in Table C-2, the second value of derivative of error is calculated as -0.22°F/Sampling Period (61.86 – 62.08°F/1 sec). Now using the membership functions for derivative of error, the approximate values for memberships in the derivative fuzzy set are:

NL = Negative Large = 0.0

NM = Negative Medium = 0.0

NS = Negative Small = 0.45

ZE	= Zero	= 0.85
PS	= Positive Small	= 0.25
PM	= Positive Medium	= 0.0
PL	= Positive Large	= 0.0

Once the fuzzy error and derivative sets are known, FL is used to obtain the fuzzy output set. To continue with the above example, the following rules are active:

If (error is ZE) and (derivative is NS) then (control signal = NS)

If (error is ZE) and (derivative is ZE) then (control signal = ZE)

If (error is ZE) and (derivative is PS) then (control signal = PS)

If (error is PS) and (derivative is ZE) then (control signal = PS)

If (error is PS) and (derivative is NS) then (control signal = ZE)

If (error is PS) and (derivative is PS) then (control signal = PM)

All the other rules have zero antecedents (the conditions in the "if" part were all zero).

C.9 Example for Defuzzification Process

Using the active rules, the fuzzy control signal output will depend on all the above cases.

U = (0.0, 0.0, 0.45, 0.85, 0.35, 0.10, 0.0)

The values for the memberships in the fuzzy set are determined by the strength of application of each rule as shown in Figure C-7.

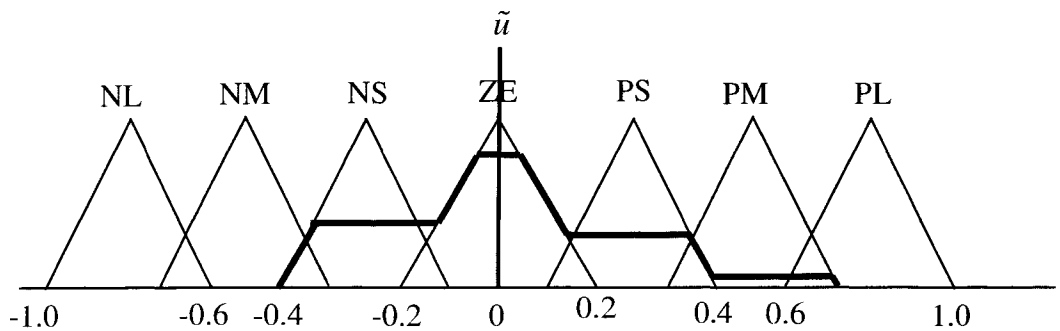

Figure C-7: Defuzzification of Output (Centroid Method)

The defuzzification of the control signal output is performed by calculating the centroid of the area under the curve (in bold) shown in Figure C-7. This value is 0.4. So the valve position will be changed to 61.5 % open from its current position of 61.1% open. The value of control signal output is dependent upon the magnitude of error and derivative of error. Zero control signal output assumes no change in the valve position.

Appendix D Neural Networks

(Matlab, 2001)

D.1 Simple Neuron

A neuron with a single input and no bias is shown on the left side of Figure D-1 and with bias on the right side.

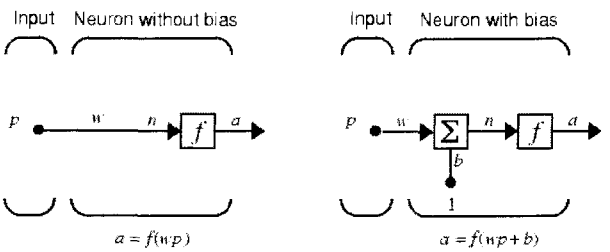

Figure D-1: Simple Neuron Network

where,

p = input

w = weight

f = transfer function

a = output

b = bias,

a = f(wp +b)

with b = 0,

a = f(wp)

This above network is trained by adjusting w and b to get desired output for the inputs provided.

D.2 Transfer Functions

Various transfer functions are available to provide output depending upon the magnitude of the input. Some common examples are discussed below:

1. Hard-Limit Transfer Function

A Hard-Limit Transfer Function is shown in Figure D-2. Output from this function is 0 if the input is less than 0 and the output is 1 if the input is greater or equal to 0.

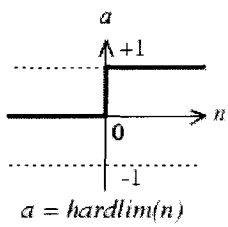

Figure D-2: Hard-Limit Transfer Function

2. Linear Transfer Function

A Linear Transfer Function is shown in Figure D-3.Output can take any value, depending upon the input/s value. Input may vary from positive infinity to negative infinity.

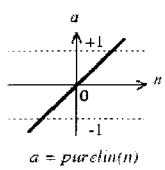

Figure D-3: Liner Transfer Function

3. Tan-Sigmoid Transfer Function

A Tan-Sigmoid Transfer Function is shown in Figure D-4. Output from this function varies from -1 to 1. Input may vary from positive infinity to negative infinity.

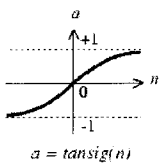

Figure D-4: Tan-Sigmoid Transfer Function

D.3 Neuron with Multiple Inputs

Figure D-5, shows a single layer NN with multiple (R) inputs and one neuron. Let p1, p2..... pR be the R inputs, let w1, 1, w1, 2, ... w1, R, be the weights. w1, 1 is the weight connection between first neuron and first input. Similarly, w1, 2, = is the weight connection between first neuron and second input. b is bias.

The output n = w1, 1 * p1 + w1, 2 * p2 ..... w1, R * pR + b

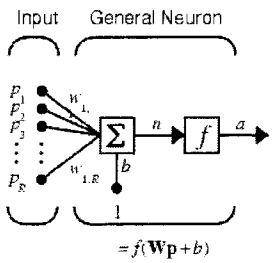

Figure D-5: Neuron with Multiple Inputs

D.4 Neural Network Architecture

A one layer network with R inputs and S neurons in a single hidden layer is shown in Figure D-6.

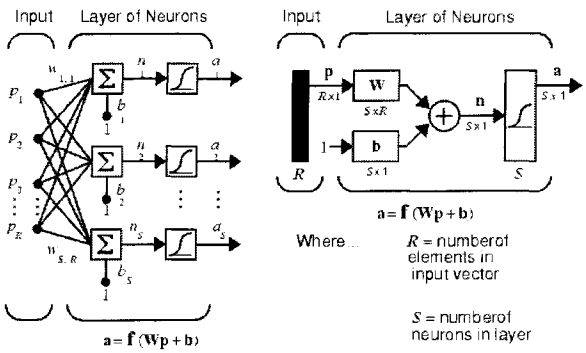

Figure D-6: Generalized Neural Network with a single Hidden Layer Architecture

where,

p = input vector,

R = number of elements in input vectors,

S = number of neurons in layer,

wsr = weight from sst neuron to rst input,

bs = bias associated with sst neuron,

n_s = net input to the transfer function = $\sum(w_{sr} * p_R) + b_s$,

as =scalar output associated with sst neuron,

Each input is weighted with an appropriate w. The sum of the weighted inputs and the bias forms the input to the transfer function f. Tan-Sigmoid (tansig) and Linear (pureline) Transfer Function are used in the model. Neurons may use any differentiable transfer function f to generate their output. Use of sigmoid neurons in the last layer restricts the outputs of the network over a small range, hence linear neurons are used which provide any value of the network output.

D.5 Feedforward Network

Feedforward networks generally have one or more hidden layers, followed by an output layer. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between inputs and output vectors. The linear output layer produces the network output in the range of-1 to +1, if a linear transfer function is used and in the range of 0 to 1 if a logsig transfer function is used.

Figure D-7 shows generalized NN architecture, with logsig as a transfer function. Figure D-7 shows NN architecture used for the project. The tansig transfer function is used in the hidden layer and pureline in the output layer for one neuron.

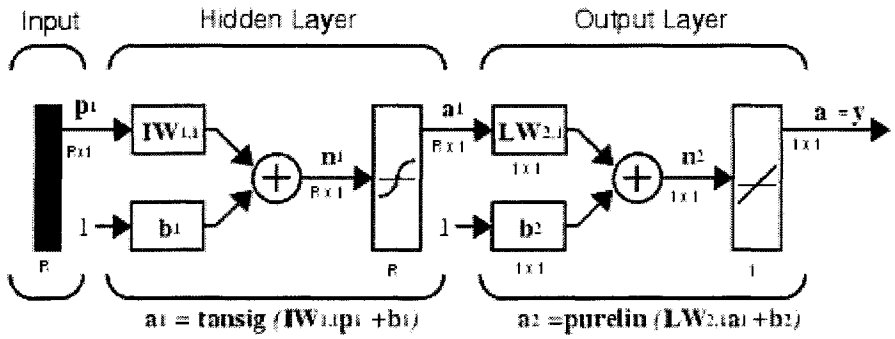

Figure D-7: General Function Approximator

where,

IW = Input Weight Matrices,

LW = Layer Weight matrices,

a = output,

D.6 Backpropagation Algorithm

The different training algorithms for feedforward networks use the gradient of the performance to determine how to adjust the weights to minimize performance. A technique which involves performing computations backward through the network is called Backpropagation and is derived using the chain rule of calculus.

Backpropagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly - the negative gradient. An iteration of this algorithm can be written as:

$$x_{k+1} = x_k - \alpha_k g_k \qquad \text{... D-1}$$

where,

xk is a vector of current weights and biases,

gk is the current gradient, and

αk is the learning rate.

Incremental and Batch modes are two different ways in which this gradient descent algorithm can be implemented. In the incremental mode, the gradient is computed and the weights are

updated after each input is applied to the network. In the batch mode all the inputs are applied to the network before the weights are updated. In this project, batch mode is used during training and incremental mode is used for updating the network.

D.7 Fast Training Algorithms and Learning Rate

Fast training algorithm uses standard numerical optimization techniques and heuristic techniques to analyze the performance of the standard steepest descent algorithm. A fast training algorithm that uses standard numerical optimization techniques is used in this project.

With standard steepest descent, the learning rate is held constant throughout training. The performance of the algorithm is very sensitive to the proper setting of the learning rate. If the learning rate is set too high, the algorithm may oscillate and become unstable. If the learning rate is too small, the algorithm will take too long to converge. It is not practical to determine the optimal setting for the learning rate before training, and, in fact, the optimal learning rate changes during the training process, as the algorithm moves across the performance surface.

The performance of the steepest descent algorithm can be improved if we allow the learning rate to change during the training process. An adaptive learning rate will attempt to keep the learning step size as large as possible while keeping learning stable. The learning rate is made responsive to the complexity of the local error surface.

First, the initial network output and error are calculated. At each epoch, new weights and biases are calculated using the current learning rate. New outputs and errors are then calculated. If the new error exceeds the old error by more than a predefined ratio, the new weights and biases are discarded. In addition, the learning rate is decreased. Otherwise, the new weights are kept. If the new error is less than the old error, the learning rate is increased. This procedure increases the learning rate, but only to the extent that the network can learn without large error increases. Thus, a near-optimal learning rate is obtained for the local terrain. When a larger learning rate could result in stable learning, the learning rate is increased. When the learning rate is too high to guarantee a decrease in error, it gets decreased until stable learning resumes.

Appendix E Sensitivity Analysis

To study the effect of each variable on predicted SAT, a sensitivity analysis was performed. To conduct this study, one input variable to a NN model was varied over its operating range in fixed steps, keeping the other variables constant and the output was recorded. Since the NN was trained on the data taken during actual operation, the resulting analysis is for the combined coil/control system.

Sensitivity Analysis in the present study has 15 parameters, namely;

1. CHWC-EWT – Chilled Water Coil Entering Water Temperature
2. CHWC-LWT – Chilled Water Coil Leaving Water Temperature
3. CHWC-MWT – Chilled Water Coil Mixed Water Temperature
4. CHWC-VLV – Chilled Water Coil Value Position (% Open)
5. CHWP-GPM – Chilled Water Pump Flow Rate (GPM)
6. EA-DMPR – Exhaust Air Damper (% Open)
7. MA-TEMP – Mixed Air Temperature
8. OA-CFM – Outside Air Flow Rate (CFM)
9. OA-DMPR – Outside Air Damper (% Open)
10. OA-TEMP – Outside Air Temperature
11. RA-CFM – Return Air Flow Rate (CFM)
12. RA-DMPR – Return Air Damper (% Closed)
13. RA-TEMP – Return Air Temperature
14. SA-CFM – Supply Air Temperature
15. SATSPT – Supply Air Set Point Temperature

which are considered as input variables to the NN model. The analysis was performed to check how the SAT changes as each of the input parameters was varied in the operating range, with other parameters held constant at the base case.

The advantage of using sensitivity analysis is to show how significant any given input variable is in determining SATPRED. This is done by displaying the range of possible SAT for a range of input values for each parameter. This assists in making a decision whether a particular parameter should be included in the NN modeling or not. It also helps to identify

critical input parameters that need to be monitored for data collection and analysis in order to check for erroneous data and the performance of the NN model. The operating range used for each parameter was determined from previously collected data. Histograms for the previous data sets are shown in Figure E-1 and Figure E-2

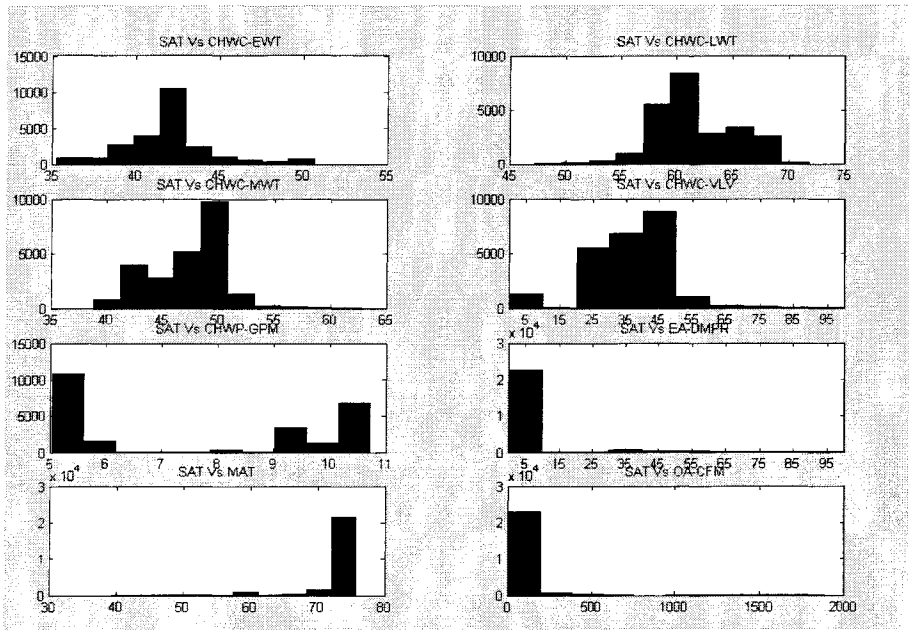

Figure E-1: Training Data Distribution - I

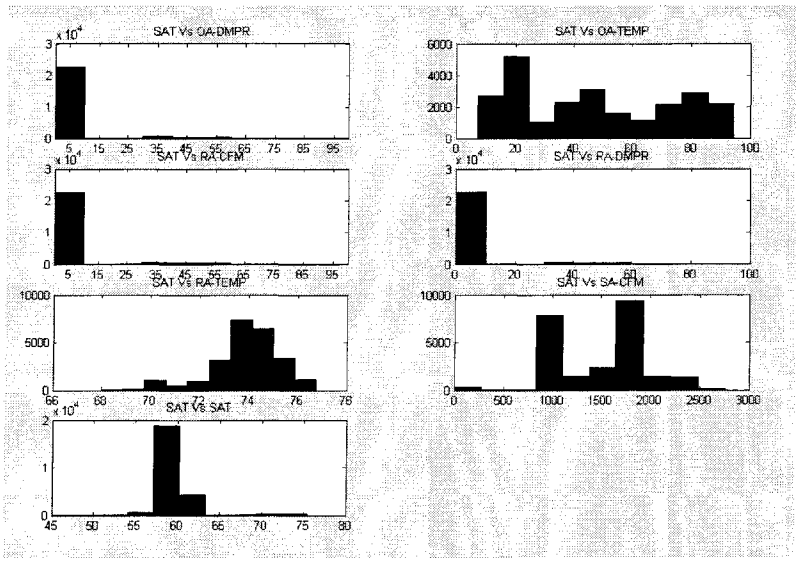

Figure E-2: Training Data Distribution - II

This NN model was trained with all the above mentioned study parameters. This trained NN has 15 input parameters, 15 nodes in the hidden layer and one output (predicted SAT). The trained NN has RMS error of 0.8.

The effect of each parameter for its operating range on SAT was studied, holding other parameters constant at their base case values. Base case values were found using the above histogram results as an average of operating range and occurrence of maximum value Table E-1 provides the base case and variation in operating range for each studied input.

Table E-1: Base Case and Variation in each input for Sensitivity Analysis study

Parameter	Base Case	Variation
EWT (°F)	45	35-2-55
LWT (°F)	60	45-2-75
MWT (°F)	50	40-2-60
CHWC VLV Position (% Open)	50	0-5-100
CHWP-GPM (GPM)	8	4/1/2012
EA-DMPR (% Open)	50	0-5-100
MAT (°F)	70	60-1-80
OA-CFM (CFM)	200	0-20-400
OA-DMPR (% Open)	50	0-5-100
OAT (°F)	50	0-5-100
RA-CFM (CFM)	1000	0-100-2000
RA-DMPR (% Close)	50	0-5-100
RAT (°F)	75	65-2-85
SA-CFM (CFM)	1800	600-100-3000

Note: Three numbers in variation column are minimum value for that variable, step size in which that variable was incremented and maximum value for that variable respectively.

Simulations were performed for all the cases and results were recorded. Figure E-3 and Table E-2 show the variation in SAT for changes in the inputs for the most significant parameters.

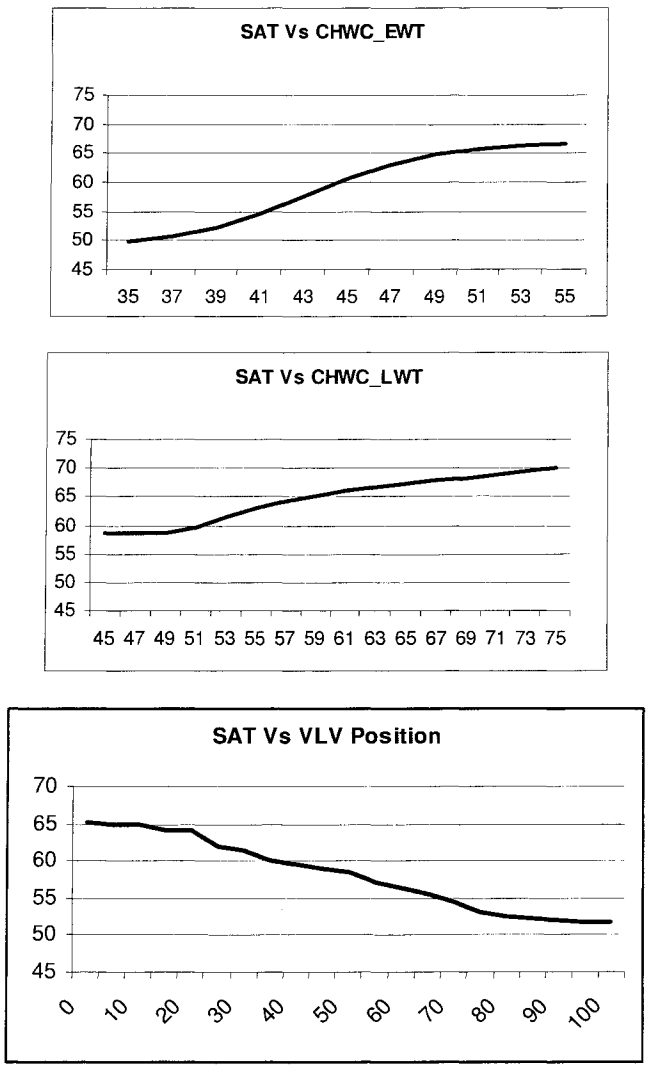

Figure E-3: Results for Sensitivity Analysis Study

Table E-2: Result for Sensitivity Analysis Study

Parameter	SAT Variation
EWT (°F)	16.65
LWT (°F)	11.63

Table E-2: continued...

MWT (°F)	3.91
CHWC VLV Position (% Open)	6.66
CHWP-GPM (GPM)	1.00
EA-DMPR (% Open)	8.00
MAT (°F)	7.51
OA-CFM (CFM)	3.88
OA-DMPR (% Open)	8.06
OAT (°F)	13.62
RA-CFM (CFM)	9.39
RA-DMPR (% Close)	1.25
RAT (°F)	18.90
SA-CFM (CFM)	16.34

Table E-2 suggested CHWC-EWT, CHWC-LWT, CHWC-VLV Position, MAT, CHWP-GPM, and SA-CFM should be used as inputs to the NN in predicting the SAT. These variables cause the largest SAT variations. Even though it is observed from Table E-2 that OAT, RAT, OA-DMPR and EA-DMPR have significant effect in SAT, effects of these variables will be resulted in change in MAT. So to keep the NN model simple and with less number of inputs only MAT was considered.

Appendix F Genetic Algorithms

F.1 Introduction

A Genetic Algorithm is an optimization technique based on the evolutionary processes found in nature. It is very flexible and robust and thus suitable for many optimization problems.
In a FLC, it is difficult to construct optimal Fuzzy Rule Matrix (FRM) and membership functions. Based upon the parameters in the FRM, control actions are taken for a particular system for which it is developed. Genetic Algorithms (GAs) can be used to find optimal values for the FLC parameters. Figure F-1shows a Flow Chart for GAs.

Explanation of each step of the GA

1. **Representation of Problem in GA**

   Before a GA can be initialized, a representation for the problem must be done. The real world problem must be encoded in the form of binary or integer genes. In this study, to keep the problem analogous with the FRM and easy to understand, a non-binary gene of membership values is used for the representation.

   Two ways were considered for aligning the FRM values:

   - Row Form
   - Spiral Form (Spiral moving outside from the center)

2. **Initialization**

   The initial population is randomly generated within a range of values.

3. **Generalization**

   Maximum generation is the number of times a new population will be created from the old population. Generation is a loop counter. The number of generations in a GA should be executed is population size and problem dependent.

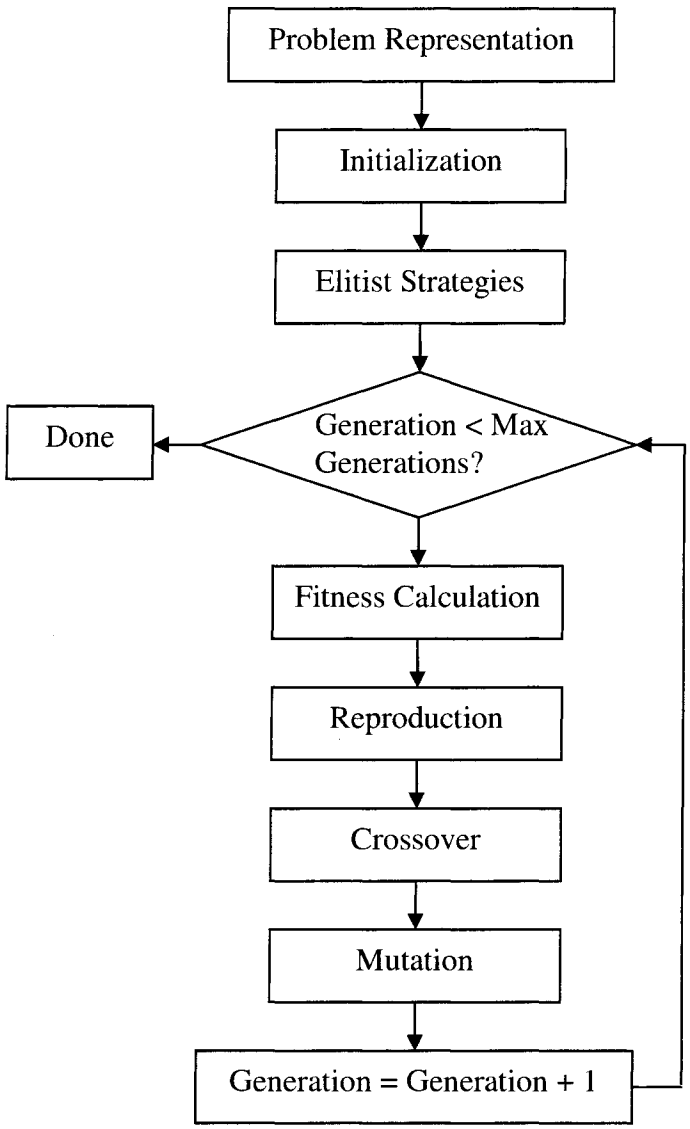

Figure F-1: Flow chart for GA

4. Calculating Fitness Function

In a HVAC control system, a controller which drives the system to zero error and zero derivative of error is considered to be a better controller. Using this mean, RMS error of a point defined by error (x-axis) and derivative of error (y-axis) is used as a Fitness Function or Figure of Merit (FOM).

$$\text{Fitness Function} = \text{RMS}_{\text{ERROR}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\text{SAT-SATSPT})^2} \qquad \text{... F-1}$$

GAs will be used to evolve different sets of FRM and Fitness Functions to improve FLC performance.

In this case, other potential fitness functions could be rise time, overshoot, controller travel, etc.

5. **Elitist Selection Strategies**

Elitist Section Strategies ensure that the best individuals are never lost in the transition from one generation to the next. Elitism is a technique used to create an early convergence by ensuring the survival of the best member in each population. Elitism compares the members of the most recent generation to the members of the previous generation. It then combines the two generations and determines the best members from both generation in the decreasing order of fitness function value or FOM. The elitist selection policy carries forward the fittest individual from the previous generation into the next generation.

6. **Tournament Selection for Reproduction**

Tournament selection randomly chooses members from the population for reproduction. A Fitness Ranking technique is used to sort the members in a tournament. Two different tournament sizes, 4 and 7, are studied.

The two fittest members, called parents, based upon the fitness function value are selected from the population and are combined to reproduce two new members, called children in the Crossover Step.

7. **Crossover**

Crossover is the primary genetic operator which promotes the exploration of new regions in the search space. Crossover is a structured, yet randomized mechanism of exchanging information between genes. Standard Crossover begins by selecting two members previously placed in the mating pool during reproduction. A crossover point is then selected at random and information from one parent, up to another crossover point, is exchanged with the other member. This creates two new members for the next generation. Single and Standard Two Point Crossover are studied in this project.

Single Point Crossover:

This operator randomly chooses a crossover point and exchanges the subsequences before and after that point between two genes to create two offsprings

For example:

1st Gene -- [1 0 1 0 0 1 0 1 0 0 1 1]

2nd Gene -- [0 0 1 1 1 0 1 0 1 0 1 0]

And crossover position is 4th

After crossover, following offsprings are created

1st Offspring -- [1 0 1 0 1 0 1 0 1 0 1 0]

2nd Offspring -- [0 0 1 1 0 1 0 1 0 0 1 1]

Two Point Crossover:

This operator randomly chooses two crossover points and exchanges the subsequences after 1st crossover point until 2nd crossover point between two genes to create two offsprings

For example:

1st Gene -- [1 0 1 0 0 1 0 1 0 0 1 1]

2nd Gene -- [0 0 1 1 1 0 1 0 1 0 1 0]

And crossover positions are 4th and 8th

After crossover, following offsprings are created

1st Offspring -- [1 0 1 0 1 0 1 0 0 0 1 1]

2nd Offspring -- [0 0 1 1 0 1 0 1 1 0 1 0]

8. Mutation

Mutation is a secondary operator. This operator ensures that all the points in the solution space have a chance of being searched. Generally, mutation is performed by toggling the binary bits. Since integers are used in the representation system, a random number for all the possible representative integers is generated and exchanged with the randomly selected number.

For example:

Gene -- [1 0 1 0 0 1 0 1 0 0 1 1]

Say three locations namely 2nd, 5th and 9th bit are randomly selected for mutation. In the present study mutation bit is replace by binary number generated randomly. In this example, since three numbers are chosen for mutation three binary numbers are generated randomly. Let 0, 1, 1 be these three binary numbers generated randomly, then offspring is:

Gene -- [1 0 1 0 0 1 0 1 0 0 1 1]

Offspring -- [1 0 1 0 1 1 0 1 1 0 1 1]

Appendix G Evolutionary Strategies

(http://csep1.phy.ornl.gov/CSEP/MO/NODE44.html)

Evolution Strategies (ES) are in many ways very similar to Genetic Algorithms (GAs). As their name implies, ES too simulate natural evolution. The differences between GAs and ESs arise primarily because the original applications for which the algorithms were developed are different. While GAs were designed to solve discrete or integer optimization problems, ESs were applied first to continuous parameter optimization problems associated with laboratory experiments.

ES were introduced in the 1960s by Rechenberg [65] working in Berlin and further developed by Schwefel [66]. The first numerical simulations were performed by Hartmann [64], and the first attempts at using ESs to solve discrete optimization were made by Schwefel [67].

Like GAs, ES differ from traditional optimization algorithms in some important respects:

1. They search from one population of solutions to another, rather than from individual to individual.
2. They use only objective function information, not derivatives.
3. They use probabilistic, not deterministic, transition rules.

The basic structure of an ES is very similar to that of a basic GA. One minor change from the standard optimization routine flow diagram is the use of the word `population' rather than `solution'. A more major difference is that the usual operation of generating a new solution has been replaced by three separate activities --- population selection, recombination and mutation. It is in the implementation of these operations that the differences between ESs and GAs lie.

ES, GAs and associated algorithms are now known collectively as evolutionary algorithms and their use as evolutionary computation. This exciting new field is thoroughly reviewed in

much more detail than is possible here in two excellent new publications by Baeck [63] and Schwefel [69].

Mutation:

In GA implementations mutation is usually a background operator, with crossover (recombination) being the primary search mechanism. In ES implementations mutation takes a much more central role. This mutation mechanism enables the ES algorithm to evolve its own strategy parameters appropriate to the problem being tackled as the search progresses, a process termed self-adaptation by Schwefel [67]. For ES, same mutation technique given in Appendix G for Genetic Algorithm is used.

Appendix H ERS System Test Setup

ENERGY RESOURCE STATION
SYSTEM TEST SETUP

GENERAL INFORMATION VERSION:

Principle Investigator: Dr. Ron Nelson
Test Group:
Test Name: FLC 3.4 Comparison
Expected duration of test:
Date: 03/31 - 04/06/05

Phone:
E-mail:
Program:
Data: ☒ Public ☐ Private
☒ Send to FTP?

Description of test: Fuzzy Logic Control of AHU SA TEMP via chilled water valve.

Objective of test: Test of Adaptive FLC model, comparison with other model types. Comparison of FLC vs. PIDL schemes.

SPECIAL REQUIREMENTS

Need cooling load in testrooms, see baseboard heating and lighting schedule for times.

AHU SETUP (TEST SYSTEMS) COMPLETED BY:

AHU-A
System Control Mode (SYS-CTRL):
☒ Always occupied.
☐ Night setback:
Start-Up time start stop
Occupied time start stop
Set-Back time start stop
UnOcc. time: start stop

Outside Air Mode
☐ Minimum Flow rate: cfm
☒ Minimum damper position: Note 2 % open

☐ Economizer Enabled: N/A °F
% RH

Return Fan Mode
☐ Speed tracking: % Spd of SF Spd
☒ Flow tracking: Note 3 % of SA FLOW
☐ Flow rate differential: cfm offset

System Settings
Static pressure set point: 1.2 in. w. g.
Supply air temperature: Note 1 °F
PID: ☒ Summer(Pband=-45.7, Itime=120)
☐ Winter(Pband=-75, Itime=150)

Notes: 1. SAT-SPT varies by researcher. 2. SWD OA DMPRs to fixed positions by researcher. 3. SWD A, B \ RF-SPD to 80 % for rm press control.

AHU-B (SAME AS AHU-A ☒)
System Control Mode (SYS-CTRL):
☐ Always occupied.
☐ Night setback:
Start-Up time start stop
Occupied time start stop
Set-Back time start stop
UnOcc. time: start stop

Outside Air Mode
☐ Minimum Flow rate: cfm
☐ Minimum damper position: % open

☐ Economizer Enabled: °F
% RH

Return Fan Mode
☐ Speed tracking: % Spd of SF Spd
☐ Flow tracking: % of SA FLOW
☐ Flow rate differential: cfm offset

System Settings
Static pressure set point: in. w. g.
Supply air temperature: °F
PID: ☐ Summer(Pband=-45.7, Itime=120)
☐ Winter(Pband=-75, Itime=150)

Notes:

TEST ROOM SETUP COMPLETED BY: _____

TEST ROOMS A

Room Temperature Set points
Occupied time: start 0:00 stop 23:59
Heating setpoint 70 °F
Cooling setpoint 72 °F
Unoccupied time:start _____ stop _____
Heating setpoint _____ °F
Cooling setpoint _____ °F

Air Flow Rate - Interior
Occupied time: start _____ stop _____
Minimum flow rate Note 3 cfm
Maximum flow rate _____ cfm
Unoccupied time:start _____ stop _____
Minimum flow rate _____ cfm
Maximum flow rate _____ cfm

Air Flow Rate - Exterior
Occupied time: start _____ stop _____
Minimum flow rate Note 3 cfm
Maximum flow rate _____ cfm
Unoccupied time:start _____ stop _____
Minimum flow rate _____ cfm
Maximum flow rate _____ cfm

Terminal Heat Selection
☐ Hydronic coils
☒ Electric coils

Internal Loads
☒ BB heat stage-1: on Notes 1, 2 off _____
☒ BB heat stage-2: on _____ off _____
☒ Lighting stage-1 on 06:00 off 18:00
☒ Lighting stage-2 on 06:00 off 18:00

Return Duct Configuration:
☐ Ducted return
☒ Plenum return

Door Orientation:
☐ Open.
☒ Closed.
☒ Locked.

Window Configuration:
☒ No Blinds
☐ Blinds set to _____ % Open
☐ Other: _____

TEST ROOMS B same as Test Rooms A ☒

Room Temperature Set points
Occupied time: start _____ stop _____
Heating setpoint _____ °F
Cooling setpoint _____ °F
Unoccupied time:start _____ stop _____
Heating setpoint _____ °F
Cooling setpoint _____ °F

Air Flow Rate - Interior
Occupied time: start _____ stop _____
Minimum flow rate _____ cfm
Maximum flow rate _____ cfm
Unoccupied time:start _____ stop _____
Minimum flow rate _____ cfm
Maximum flow rate _____ cfm

Air Flow Rate - Exterior
Occupied time: start _____ stop _____
Minimum flow rate _____ cfm
Maximum flow rate _____ cfm
Unoccupied time:start _____ stop _____
Minimum flow rate _____ cfm
Maximum flow rate _____ cfm

Terminal Heat Selection
☐ Hydronic coils
☐ Electric coils

Internal Loads
☐ BB heat stage-1: on _____ off _____
☐ BB heat stage-2: on _____ off _____
☐ Lighting stage-1 on _____ off _____
☐ Lighting stage-2 on _____ off _____

Return Duct Configuration:
☐ Ducted return
☐ Plenum return

Door Orientation:
☐ Open.
☐ Closed.
☐ Locked.

Window Configuration:
☐ No Blinds
☐ Blinds set to _____ % Open
☐ Other: _____

System Selection:
☒ VAV ☐ FCU

Notes: 1. BBHTS1 sched: On 0:00 - 06:00, Off 06:00 - 12:00, On 12:00 - 23:59, Off 18:00 - 23:59.
2. BBHTS2 sched: On 0:00 - 06:00, Off 06:00 - 12:00, On 12:00 - 18:00, Off 18:00 - 23:59.
3. Rm airflow rates scheduled as per researcher. See schedules for values and times.
4. All testrms switched from ducted to plenum returns. 17:00 03/31.

GENERAL SYSTEMS SETUP

COMPLETED BY: _____

AHU-1

Occupied/Unoccupied mode:

☒ Always occupied.

☐ Night setback:

Occupied time start _____ stop _____

UnOcc. time: start _____ stop _____

Outside Air Mode

☐ Minimum Flow rate: _____ cfm

☐ Minimum damper position: 0 % open

☒ Economizer Enabled: _____ °F

_____ % RH

Return Fan Mode

☐ Speed tracking: _____ % Spd of SF Spd

☐ Flow tracking: _____ % of SA FLOW

☐ Flow rate differential: _____ cfm offset

☐ Building static: _____ in. w.g.

Supply Fan Settings:

Static pressure set point: _____ in. w. g.

Supply air temperature: _____ °F

Chilled Water Pump Control:

☐ Constant valve, pump VFD control

☒ Constant pump speed, valve control

(Pump on at valve 15% open, off at 5%)

Notes:

1. Normal Settings, continuously Occupied because of day and evening classes.

2.

GENERAL SERVICES AREA

Zone Temperature Set points:

Occupied time: start _____ stop _____

Zone temperature setpoint _____ °F

Unoccupied time: start _____ stop _____

Zone temperature setpoint _____ °F

Notes: _____

COOLING SYSTEM SETUP COMPLETED BY: _____

COOLING SYSTEM

Chilled Water Supply Set Up:

☒ AHU-A: Local ☒ Campus ☐
☒ AHU-B: Local ☒ Campus ☐
☒ AHU-1: Local ☐ Campus ☒
☐ Loop-C: Local ☐ Campus ☐

Chilled Water Pump Set Up:

Primary Pump CWP-CH:
- ☒ Constant speed: 100 % Speed.
- ☐ Processed controlled.

LOCAL CHILLER SYSTEM

- ☒ Enabled.
- ☐ Laid up.

Valves Settings:

- ☐ Constant:
  - Ice Make Bypass: 100 % Open to Air handlers.
  - Thermal Storage Tank: 100 % Open to Tank.
- ☐ Ice make mode.
  - Begin Ice Make mode: start _____ stop _____
  - End Ice Make mode: start _____ stop _____

Notes: AHU's-A & B both use ACCH-CH for chilled water. AHU-1 uses campus chilled water. Manual vlvs in Equip Yard set to Chiller Priority Mode. No TES ice making required.

HEATING SYSTEM SETUP COMPLETED BY: _____

Boiler setpoint:

- ☐ Constant temperature: Boiler not used °F
- ☐ Processed controlled.

Notes:
1. Constant CLG-WTR \ ACCH-LWT desired for FLC CHWC-VLV control.
2. AHU-A using std. PIDL control of CHWC-VLV. AHU-B using FLC of CHWC-VLV.
3. AHU-A, B \ SAT-SPTs switched from winter to summer PIDL parameters before start of test.

1. Print all schedules to a text file named 'sch.txt' and store it in the Doc folder.
2. Print all SWO to a text file named 'swo.txt' and store it in the Doc folder.

References

Acosta, G. G. and M. A. Mayobky, 1992. "Fuzzy Logic and Pattern Recognition in a Self-Tuning Controller" Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, NC. p 759-765.

Alcala, R., J. M. Benitez, J. Casillas, O. Cordon, and R. Perez, 2003. "Fuzzy control of HVAC systems optimized by genetic algorithms" Applied Intelligence, Vol. 18, Part 2, p 155-177.

Arima, Masanori, E. H. Hara, and J. D. Katzberg, 1995. "Fuzzy logic and rough sets controller for HVAC systems" Proceedings of the 1995 IEEE WESCANEX Communications, Power, and Computing Conference. Part 1, May 15-16 1995, p 133-138.

ASHRAE, 1997, ASHRAE Handbook – Fundamentals, ASHRAE, Atlanta, Georgia.

ASHRAE, 2000. "DOE Plan Projects Savings for New Building Energy Costs", ASHRAE Journal, p 6, November, 2000.

Borut, Z., Klopcic Marko, and Riharrd Karba, 1999. "Tuning of fuzzy logic controller with genetic algorithm" Informatica, Vol. 23, No. 4, p 559-564.

Clark, D. R., C. W. Hurley, and C. R. Hill, 1985. "Dynamic models for HVAC system components" ASHRAE Transactions Vol. 91, Part 1B, p 737-751.

Curtiss, P. S., J. F. Kreider, and M. J. Brandemuehl, 1993. "Adaptive Control of HVAC Processes using Predictive Neural Networks" ASHRAE Transactions, Vol. 99, Part 1, p 496-504.

Curtiss, P. S., 1996. "Experimental results from a network-assisted PID controller" ASHRAE Transactions, Vol. 102, Part 1, p 1157-1168.

Driankov, D., H. Hellendoorn, and M. Reinfrank, 1993. An Introduction to Fuzzy Control. Springer-Verlag, Berlin.

EERE, 2002. Report: Energy Consumption Characteristics of Commercial Building HVAC Systems Volume III: Energy Savings Potential, July.

http://www.eere.energy.gov/buildings/info/documents/pdfs/hvacvolume3finalreport.pdf, Date Accessed: 27th Feb. 2005.

EERE, 2005. http://www.eere.energy.gov/consumerinfo/saveenergy/save_space.html, Date Accessed: 27th Feb. 2005.

Evolutionary Strategies, 2005. http://csep1.phy.ornl.gov/CSEP/MO/NODE44.html. Date Accessed: 27th Feb. 2005.

Goldberg, D. E., 1989. GAs in Search, Optimization and Machine Learning, Addison Wesley, Reading, New York.

Gurocak, H.B., 1999. "Genetic-algorithm-based method for tuning fuzzy logic controllers" Fuzzy Sets and Systems, Vol. 108, No. 1, p 39-47.

Hyland, R. W. and A. Wexler, 1983. "Formulations for the thermodynamic properties of dry air from 15K TO 473.15K, and of saturated moist air from 173.15K TO 372. 15K, at pressure to MPa." ASHRAE Transactions, Vol. 89, Part 2A, p 520-535.

Huang, S. H. and R. M. Nelson, 1991. "A PID-law-combining fuzzy controller for HVAC applications", ASHRAE Transactions, Vol. 97, Pt. 2.

Huang, S. H. and R. M. Nelson, 1993. "Artificial Neural Networks Used as a Rule Selector for Fuzzy Logic Controllers", Proceedings, ASME Computers in Engineering Conference, San Diego, California.

Huang, Shou-Heng and R. M. Nelson, 1994a. "Rule Development and Adjustment Strategy of a Fuzzy Logic Controller for an HVAC System: Part One – Analysis" ASHRAE Transactions, Vol.100, Pt.1, p 841-850.

Huang, Shou-Heng and R. M. Nelson, 1994b. "Rule Development and Adjustment Strategy of a Fuzzy Logic Controller for an HVAC System: Part Two – Experiments" ASHRAE Transactions, Vol.100, Pt.1, p 851-856.

Huang, S. H. and R. M. Nelson, 1999. "Development of a Self-Tuning Fuzzy Logic Controller", ASHRAE Transactions, Vol. 105, Part 1, 1999, p 206-213.

Huang, W. and H. N. Lam, 1997. "Using genetic algorithms to optimize controller parameters for HVAC systems" Energy and Buildings, Vol. 26, No. 3, p 277-282.

Huang, Y., 1999. "Uncertainty Assessment Methodology and Applications for HVAC Systems", Technical Report, Iowa Energy Center, Ankeny, Iowa.

Incropera, F. P. and D. P. Dewitt, 1996. Fundamentals of Heat and Mass Transfer. 4th Edition, John Wiley and Sons, New York.

JCI, 1995. Digital Control Module 140 Technical Bulletin, Johnson Controls, Inc., Milwaukee, Wisconsin.

Jeannette, E., K. Assawamartbunlue, P. S. Curtiss, and J. F. Kreider, 1998. "Experimental results of a predictive neural network HVAC controller", ASHRAE Transactions, Vol. 104, No. 2, 1998, p 192-197.

Karr, C. L., 1991. "Design of an Adaptive Fuzzy Logic Controller Using a Genetic Algorithm" Proceedings of the 4th International Conference on Genetic Algorithms, San Deigo, p 450-457.

Kazuo Tanaka, and Hua Wang, 2001. Fuzzy Control Systems Design and Analysis. John Wiley and Sons, Inc, New York.

Kindel, J., F. Klawonn, and R. Kruse, 1994. "Modifications of genetic algorithms for designing and optimizing fuzzy controller" Proceedings 1st IEEE Conference on Evolutionary Computation, p 8-33.

Kiriakidis, Kiriakos, and A. Tzes, 1995. "Design of a fuzzy logic adaptive sliding controller and its application to an air flow/cooling system" Proceedings of the 1995 IEEE Conference on Control Applications. Sep 28-29 1995, p 341-346.

Klaassen, C., 2004. Personal communication, Energy Resource Station, Iowa Energy Center, Ankeny, Iowa, November.

Kolokotsa, D., D. Tsiavos, G. Stavrakakis, K. Kalaitzakis, and E. Antonidakis, 2001. "Advanced fuzzy logic controllers design and evaluation for buildings' occupants thermal-visual comfort and indoor air quality satisfaction" Energy and Buildings Vol. 33, No. 6, p 531-543.

Lebrun, J. 1990. "Simplified modeling of cooling coils" IEA Annex 17 report. Belgium, University of Liege, August.

Lee, K.Y., Y. T. Cha, and J. H. Park, 1992. "Short-Term Load Forecasting Using an Artificial Neural Network" IEEE Transactions on Power Systems, Vol. 7, No. 1, p 124-129.

Ling, K.V., A. L. Dexter, G. Geng, and P. Haves, 1991. "Self-tuning control with fuzzy rule-based supervision for HVAC applications" IFAC Symposia Series, No. 7, p 205-209.

Mamdani, E.H., 1979. "Advances in the Linguistic Synthesis of Fuzzy Controllers" Int. J. Man-Machine Studies, Vol. 8, p 669-678.

Matlab, 2001. Neural Network Toolkit, Matlab 6.5 2002, Natick, MA, 2002.

Metasys, 1995. Product Literature, Rapid Information Access (CD), Johnson Controls Inc., Milwaukee, Wisconsin.

Osman, A. M. R. B. Breekweg, and P. Gruber, 2000a. "Development of a generalized neural network model to detect faults in building energy performance - Part I" ASHRAE Transactions, Vol. 106, p 61-73.

Osman, A. M. R. B. Breekweg, and P. Gruber, 2000b. "Development of a generalized neural network model to detect faults in building energy performance - Part II" ASHRAE Transactions, Vol. 106, p 74-93.

Ozsoy, C., 1993. "Self-tuning Control of a Heating, Ventilating and Air-Conditioning System", Proceedings of the Institution of Mechanical Engineering, Part I, Journal of Systems & Control Engineering, Vol. 207, No. 4, p 243 – 251.

Park, Seihwan, and H. Lee-kwang, 2000. "Designing fuzzy logic controllers by genetic algorithms considering their characteristics" Proceedings of the 2000 Congress on Evolutionary Computation CEC, p 683-690.

Park, Young Jun, H. S. Cho, and D. H. Cha, 1995. "Genetic algorithm-based optimization of fuzzy logic controller using characteristic parameters" Proceedings of the 1995 IEEE International Conference on Evolutionary Computation. Part 2, p 831-836.

Pham, D.T., and G. Gin, 1994a. "Evolutionary design of an adaptive fuzzy logic controller for processes with time delays" Proceedings of the 1994 IEEE International Conference on Systems, Man and Cybernetics. Part 1, p 431-436.

Pham, D.T., and D. Karaboga, 1994b. "Design of an adaptive fuzzy logic controller" Proceedings of the 1994 IEEE International Conference on Systems, Man and Cybernetics. Part 1, p 437-442.

Procky, T.J. and E. H. Mamdani, 1979. "A Linguistic Self organizing Process Controller" Automatica, Vol. 15, p 15-30.

Qiao, W. Z., W. P. Zhuang, and T. H. Heng, 1992. "A Rule Self-regulating Fuzzy Controllers" Fuzzy Sets and Systems, Vol. 47. p 13-21.

Seem, J. E., 1999. "A New Pattern Recognition Adaptive Controller with Application to HVAC Systems", Automatica, Vol. 34, No. 8, p 969 – 982.

Shao, S., 1988. "Fuzzy Self-Organizing Controller and Its Application for Dynamic Process" Fuzzy Sets Systems, Vol. 26, p 151-164.

Shavit, G. and G. G. Brandt, 1982. "Dynamic Performance of a discharge air temperature system with a PI controller" ASHRAE Transactions Vol. 86, Part 2, p 826-838.

So, A. T. P., W. L. Chan, T. T. Chow, and W. L. Tse, 1998. "New HVAC Control by System Identification" Building and Environment, Vol. 30, No. 3, p 349 – 357.

Specht, Donald F., 1991. "IEEE Transactions on Neural Networks" Vol. 2, No. 6, Nov, 1991, p 568-576.

Stoecher, W. F., 1975. Procedures for simulating the performance of components and systems for energy calculations ASHRAE, 3rd Edition. Atlanta, Georgia.

Tansheit, R. and E. M. Scharf, 1988. "Experiments with the Use of a Rule-Based Self-Organizing Controller for Robotics Applications" Fuzzy Sets Systems, Vol. 26, p 195-214.

Velez-Reyes, M and B. Arguello-Serrano, 1999. "Nonlinear control of a heating, ventilating, and air conditioning system with thermal load estimation" IEEE Transactions on Control Systems Technology, Vol. 7, No. 1, p 56-63.

Wallenborg, A.O., 1991. "New self-tuning controller for HVAC systems" ASHRAE Transactions, pt 1, p 19-25.

Wang, Qing-Guo., 1999. "Multivariable controller auto-tuning with its application in HVAC systems" Proceedings of the 1999 American Control Conference (99ACC), p 4353-4357.

Wang, S. and Jin, X., "Model-based Optimal Control of VAV Air-conditioning System Using Genetic Algorithm," Building and Environment, Vol. 35, No. 6, p 471 - 487, 2000.

Wen, J. and T. F. Smith, 2003. Development and Validation of Adaptive Optimal Operation Methodology for Building HVAC Systems: Volume I - Description, Department of Mechanical and Industrial Engineering, The University of Iowa, Iowa City, Iowa.

Wu, Jian, and W. Cai, 2000. "Development of an Adaptive Neuro-Fuzzy method for supply air pressure control in HVAC system" 2000 IEEE International Conference on Systems, Man and Cybernetics. p 3806-3809.

Xu, Chen-Wei, 1987. "Fuzzy System Identification" IEEE Proceedings, Vol. 136, Pt. D, No. 4, p 146-150.

Ying-guo, Piao, Z. Hua-guang, and Z. Bien, 1998. "Simple fuzzy adaptive control method and application in HVAC" IEEE International Conference on Fuzzy Systems, Vol. 1, IEEE World Congress on Computational Intelligence, p 528-532.

Zaheer-uddin, M., 1993a. "Optimal, Sub-optimal and Adaptive Control Methods for the Design of Temperature Controllers of Intelligent Buildings" Building and Environment, Vol. 28, No. 3, p 311 – 322.

Zaheer-uddin, M., 1993b. "Load-preview decentralized robust controllers for HVAC systems" Proceedings of the 1993 American Control Conference, Part 3, p 2740-2743.