Residential Air-Conditioner and Heat Pump System, Cooling Mode, Rule-Based Chart Fault Detection and Diagnosis Software User's Guide

W. Vance Payne

This publication is available free of charge from: https://doi.org/10.6028/NIST.TN.2130



Residential Air-Conditioner and Heat Pump System, Cooling Mode, Rule-Based Chart Fault Detection and Diagnosis Software User's Guide

W. Vance Payne HVAC&R Equipment Performance Group Energy and Environment Division Engineering Laboratory

This publication is available free of charge from: https://doi.org/10.6028/NIST.TN.2130

December 2020



U.S. Department of Commerce Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology Walter Copan, NIST Director and Undersecretary of Commerce for Standards and Technology This software package was developed by the National Institute of Standards and Technology (NIST), is not subject to copyright protection, and is in the public domain. It can be used freely provided that any derivative works bear some notice that they are derived from it.

NIST used its best efforts to provide a high-quality software package and to select modeling methods and correlations based on sound scientific judgement. However, NIST assumes neither responsibility nor liability for any damage arising out of or relating to the use of these FDD codes. The software is provided "AS IS"; NIST makes NO GUARANTIES and NO WARRANTIES OF ANY TYPE, expressed or implied, including NO WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using and distributing the software and you assume all risks associated with its use, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and the unavailability or interruption of operation. This software is not intended to be used in any situation where a failure could cause risk of injury or damage to property.

Certain commercial entities, equipment, or materials may be identified in this document to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

National Institute of Standards and Technology Technical Note 2130 Natl. Inst. Stand. Technol. Tech. Note 2130, 53 pages (December 2020) CODEN: NTNOEF

> This publication is available free of charge from: https://doi.org/10.6028/NIST.TN.2130

Abstract

The software, used to process raw data coming from a residential air-conditioner or heat pump that is operating in the cooling mode, is described. The code was written in LabView 2015 with five main code modules grouped together in a project file. This technical note describes the use of each code module and is meant to accompany a copy of the project code with example files illustrating how each module is used. All the project files and example files are being made available to the public in a ZIP type archive. This software package was developed by the National Institute of Standards and Technology (NIST), is not subject to copyright protection, and is in the public domain.

Key words

Fault detection and diagnosis; Energy efficiency; Heat pump; LabView software archive; Residential air-conditioner; Reliability; Rule-based chart.

Table of Contents

1.	Int	roduction	1
2.	INS	STALLATION	2
	2.1.	System Requirements	2
	2.2.	Installation Procedure	2
3.	FD	D COOLING SS Detector Self TRAINING Ver03.vi	3
	3.1.	TAB: Datafile & Setup	3
	3.2.	TAB: Setup Data Filter	3
	3.3.	TAB: VIEW Data Filters	4
	3.4.	TABs: Plots of Selected Data 1, 2, and 3	4
	3.5.	TAB: Setup and Apply SS Detector	4
	3.6.	TABs: SS Detector 2, 3, and 4	4
4.	FD	D Self TRAINING Parse Datafiles into BINS Ver06.vi	11
	4.1.	TAB: Input Data #1	1
	4.2.	TAB: Parse Into BINS #2	1
	4.3.	TAB: Parse Into BINS #3	1
	4.4.	TAB: Parse Into BINS #4	12
	4.5.	TAB: SAVE Parsed BINS #5	12
5.	FD	D Self Training POLYNOMIAL FIT Ver07.vi1	18
	5.1.	TAB: Select COL Data File #1	18
	5.2.	TAB: RAW DATA & CALCS #2	8
	5.3.	TAB: RAW DATA & CALCS #3	18
	5.4.	TAB: Select Parameters #4	18
	5.5.	TAB: FIT and STATS #5	19
	5.6.	TAB: Remove Outliers #6	19
	5.7.	TAB: Remove Outliers #7	19
	5.8.	TAB: FIT and STATS #8	20
	5.9.	TAB: BACKWARD Elimination #9	20
	5.10.	TAB: Debug Page Coefficient MODELS #10	20
	5.11.	TAB: Reduced Coefficient MODELS #11	20
6.	FD	D Self TRAINING Polynomial APPLY FIT Ver02.vi	32
7.	FD	D Algorithm Cooling OFFLINE LV2015 Ver004.vi	32
	7.1.	TAB: Input Datafile	32
	7.2.	TAB: Assign Features, SS Detector	33

Re	ferer	ıces		44
8.	SU	MMA	RY	44
,	7.5.	TAB:	NOTES	37
,	7.4.	TAB:	Rule-based Chart PLOTS	37
,	7.3.	TAB:	Cooling FDD Rule-based Chart	35

List of Tables

Table 1:	Parameters	used to	determine	moving	window	features and	values		4
----------	------------	---------	-----------	--------	--------	--------------	--------	--	---

List of Figures

Figure 1: Project file listing as shown in LabView 2015
Figure 2: FDD SSD datafile and setup page
Figure 3: FDD SSD setup data filter tab
Figure 4: VIEW Data Filters tab7
Figure 5: Plots of Selected Data 1 tab
Figure 6: Setup and Apply SS Detector tab
Figure 7: SS Detector 2, 3, and 4 tabs 10
Figure 8: Input datafile tab
Figure 9: Parse into bins #2 tab14
Figure 10: Parse into bins #3 tab15
Figure 11: Parse into bins #4 tab16
Figure 12: Save parsed bins #5 tab 17
Figure 13: Select COL data file #1 tab
Figure 14: Raw data and calculations #2 tab
Figure 15: Raw data and calculations #3 tab
Figure 16: Select parameters #4 tab
Figure 17: Fit and stats #5 tab
Figure 18: Remove outliers #6 tab
Figure 19: Remove outliers #7 tab
Figure 20: Fit and stats #8 tab
Figure 21: Backward elimination #9 tab
Figure 22: Debug page coefficient models #10 tab

Figure 23:	Reduced coefficient models #11 tab	31
Figure 24:	Input filename #1 tab	38
Figure 25:	Input datafile tab	39
Figure 26:	Assign features and steady-state detector tab	40
Figure 27:	Cooling FDD rule-based chart tab	41
Figure 28:	Rule-based chart plots tab	42
Figure 29:	Notes tab	43

1. Introduction

Fault detection and diagnosis for residential systems has received much attention over the last 20 years (Rogers, Guo, and Rasmussen 2019) (Katipamula and Brambley 2005). Many of the early notable works on fault detection and diagnosis were produced by university researchers (Rossi and Braun 1997) (J. Braun 2001) (J. E. Braun 2003) (Yuill and Braun 2013). There has been an almost exponential growth in the number of publications dealing with fault detection and diagnosis, especially with new tools that use machine learning techniques to analyze data sets with and without faults (Zogg, Shafai, and Geering 2001) (Han et al. 2010) (Isermann 2005) (Esen et al. 2008) (Du et al. 2014; Han et al. 2020). These new techniques have all shown the need for more operational data on installed systems to serve as training data for advanced neural networks and other algorithms.

Work on the problem of residential systems' fault detection and diagnosis began in the mid 2000's (Kim et al. 2006) (Kim et al. 2008). NIST's efforts focused on collecting data on typical residential heat pump systems operating in the cooling and heating modes with and without faults (Kim et al. 2009) (Yoon, Payne, and Domanski 2011). FDD techniques from the literature were applied to determine those techniques that would be suitable for deployment on low cost micro controllers and processors; this led to the adoption of a simple rule-based chart method of FDD originally proposed by Rossi (Rossi and Braun 1997). NIST continued with this work and sponsored a Small Business Innovative Research (SBIR) project to develop a low cost, easily programmable data logger that could be installed on residential AC and HP systems (Blemel, Kenneth (Management Sciences) 2014). Throughout this work, NIST researchers have developed computer codes to manipulate, correlate, and further analyze data from their experimental efforts; this report is a first attempt at documenting some of these codes and putting them into a form that would be useful for other FDD researchers, the HVAC industry, and software developers.

This report lists and explains the use of the NIST developed software for performing Fault Detection and Diagnostics (FDD) on vapor compression refrigeration equipment as applied to residential air-conditioners (ACs) and heat pumps (HPs) operating in the cooling mode. Figure 1 shows the LabView 2015 Project Explorer, which list the programs in the order as they will be presented in this manual. However, the programs are stand-alone and independent as written, and may be used in any sequence.

FDD Self Training Project2.lvproj * - Project Explorer	_	\times
File Edit View Project Operate Tools Window Help		
*≥ 😂 😭 🐰 🗈 © X 📽 🝕 🖽 • 🐔 📐 🛸 💓 🥪 🔍 🧐 🖻	14	
Items Files		
 Project: FDD Self Training Project2.lvproj My Computer FDD COOLING SS Detector Self TRAINING Ver03.vi FDD Self TRAINING Parse Datafile Into BINS Ver05.vi FDD Self TRAINING Polynomial FIT Ver07.vi FDD Self TRAINING Polynomial APPLY FIT Ver02.vi FDD Algorithm Cooling OFFLINE LV2015 Ver004.vi Dependencies vi.lib Control 1.ctl MOVING WINDOW Mean_Std DEV_SLOPE Ver01.vi Construct New Filename Ver013.vi Ivanlys.dll Generate All Combinations of N-Dimensional Coordinates Ver05.vi Calculate BIN DISTANCE XYZ Keep Best Ver01.vi Students T Probability Distribution Function.vi Moving AVG&STDEV.vi Rule Base Chart and Fault Probabilites Ver001.vi 		

Figure 1: Project file listing as shown in LabView 2015

The first five (5) files with names of FDD*.vi are explained in the following sections; the other files are called by these first five programs.

2. INSTALLATION

2.1. System Requirements

- Personal computer capable of running Microsoft Windows 7, 8, 10, or XP with Service Pack 3
- National Instruments LabView 2015 Professional Development System (National Instruments 2015)
- Hard disk with 25 megabytes of available space
- Screen resolution should be set to 800 x 600 or higher to view images in their entirety.

2.2. Installation Procedure

Place the ZIP file on any computer directory and unzip the archive. All the code and example files will be present in that directory. Initiate a LabView session and load the project file. Selectively run the codes in the order as they are being presented below to fully understand an example implementation.

3. FDD COOLING SS Detector Self TRAINING Ver03.vi

This program implements a data filter and Steady-State Detector (SSD) to keep only "steady" data in the raw data file. The raw datafile is a tab delimited text file. For clarity, an input file is included in the archive to provide an example of format and style. The raw tab delimited text file is a time sequence of scans that is arranged in columns with a header label at the top of each column. One of the columns includes the time in seconds. The size of the datafile is limited by the available computer memory. The software has been successfully tested with files as large as 20 MB. Larger files may be parsed into sub-files to prevent an out of memory error. The raw data should be collected at a rate suitable for the system being diagnosed. See NIST Technical Note 1087 (Kim et. al 2008) for a complete description of determining a sampling size. The Nyquist theorem (Wikipedia 2020b) still applies even for the sampling rate in FDD; sample at twice the frequency of the variation in the variable you are trying to measure. For most purposes, a sampling rate of between 10 sand 1 minute will allow detection and diagnosis in residential systems.

3.1. TAB: Datafile & Setup

Figure 2 shows the first tab page of the SSD. The *Input File* (red capital letter A in Fig. 2) is the raw data file. The *Output File* (B) contains only the column numbers selected in *Independent Variables X-Values Indices* (C) and *Dependent Variables Y-Values Indices* (D). The data that will be filtered and run through the SSD are shown in the X&Y Variables Selected String Array (E). The output file is the same format as the raw data file, but it only has the selected X and Y columns of data with all the non-steady rows of data removed once the raw file has been processed (the program executes). The independent variables can be selected from any variable being measured or calculated. For heat pump operations, the independent variables are the indoor and outdoor conditions which are best described by the indoor drybulb temperature, indoor dewpoint temperature, and outdoor dry-bulb temperature; therefore, these are the independent variables that drive the system and produce responses seen through the dependent variables.

Once the user has decided on a file to process, they select *Extract X&Y Data*, button (J). If they want to save the filtered and steady-state data to the output file, they select the *SAVE the Conditional and SSD Filtered Data*, button (H). The *PAUSE*? button (I) is used if the user had selected the input file as a list of different files by pressing the *READ LIST to PROCESS* button and wants to pause between the processing of different files to examine output for each file in the list. An example file list is included in the archive. This allows the user to process a single file or an entire list of files. If a list of files is selected, all of the steady data will be saved in the same output file.

3.2. TAB: Setup Data Filter

Figure 3 shows a snapshot of the setup data filter tab. This page contains an array of user defined conditional filters (A) that may be imposed on any data column; each column is defined by its index as seen in the array of header labels (D) that were selected in the previous tab. The program needs the raw data without the header row, so the delete header row button (B) is set to true by default. The conditional filters are executed on all the data columns selected and the resultant data can be seen in the *X*&*Y* Values Conditional Filtered Data, array (C).

3.3. TAB: VIEW Data Filters

Figure 4 shows a snapshot of the conditional filters that were defined in the previous tab. This allows the user to review the filters and to ensure that the appropriate data column is assigned to the intended conditional filter. No user input is required on this TAB.

3.4. TABs: Plots of Selected Data 1, 2, and 3

Figure 5 shows the *Plots of Selected Data 1* tab which is very similar to the *Plots of Selected Data 2* and *Plots of Selected Data 3* tabs. These three tabs show plots of selected data with indices that correspond to the data columns in the *Selected Data Header Labels* (E). The *XY Graph1 X-Variable 2* (A) is the x-axis variable and the *XY Graph1 Y-Variable 2* (B) is the y-axis variable. The mean and standard deviation are calculated for the y-variable and displayed at (C). The y-variable header label is also displayed above each graph similar to (D). The user familiar with LabView coding may customize the plots in any way they want to see the output.

3.5. TAB: Setup and Apply SS Detector

Figure 6 shows the setup of the moving window steady-state detector. The number of data points in the moving window is set at (A). Several different columns of data that were selected from the raw data and shown in the *Selected Data Header Labels* (D), may be included in the array of *Steady-State Monitoring Variables* (B). In the example data being processed and shown in Figure 6, the *Selected Data Header Labels* (D) index 9 is used as the X-Value for MW (Moving Window) and index 63 is the Y-Value for MW; the *Steady-State Monitoring Variables* (B) may include any number of elements from (D) as one of the variables monitored for steady-state determination. In Figure 6, nine variables are selected for monitoring as shown in the *Display all the SS Monitoring Variables on ONE SCREEN* (C) array. The standard deviation and slope of the values in the moving window are used to determine steady-state; the user specifies a maximum value of standard deviation and slope within the moving window. A detailed analysis of the moving window steady-state detector may be found in Kim et al. (2008).

3.6. TABs: SS Detector 2, 3, and 4

Figure 7 shows an example of the remaining three tabs in the SSD module; these last three tabs consist of plots of the steady-state data filtered per the restrictions specified in the previous tab. The x- and y-axes are specified in (A) and (B), respectively. The indices are selected per the list of header labels in (C). The Y-Variable name for each plot is also shown at (D).



Figure 2: FDD SSD datafile and setup page



Figure 3: FDD SSD setup data filter tab



Figure 4: VIEW Data Filters tab



Figure 5: Plots of Selected Data 1 tab



Figure 6: Setup and Apply SS Detector tab



Figure 7: SS Detector 2, 3, and 4 tabs

4. FDD Self TRAINING Parse Datafiles into BINS Ver06.vi

This program module applies a data clustering algorithm to the steady-state data to further reduce the number of data points required to perform a fault-free feature correlation. The details of this technique may be found in (Payne, Heo, and Domanski 2018). This program groups the data in temperature bins formed by the indoor air dry-bulb temperature, the indoor air dew-point temperature, and the outdoor air dry-bulb temperature. This program assumes that the data being binned has been through the steady-state detector, but it is not necessary to give this program steady data; it will perform the binning on any three variables that the user specifies.

4.1. TAB: Input Data #1

Figure 8 shows the screenshot of the Input Data #1 tab. This tab allows the user to input the datafile name (A) for the steady-state data that was gleaned from running the SSD program described in Sec. 2 and define the bins that will be used to cluster (filter) the steady-state data. Referring to label (D) in Fig. 8, the user must select the three independent variables that will be used to correlate all the other variables (other FDD parameters). For example, the SS-Example RAW Datafile.txt contains three data columns which are used as the independent variables; each column of the independent variables is filtered by a simple Euclidian distance technique that determines the distance from the current data point to the center of an XYZ data bin and keeps the five closet points (B) for each bin that is defined in the Data BINS Defined - Array (D). Array (D) describes Independent Variables #0, #1, and #2; in this example, #0 is Datafile Column 182, Inlet TC Grid Avg. Temp (F), #1 is Datafile Column 83, KahnDew TempF, and #2 is Datafile Column 195, OD Air Avg. DB (F). These three variables are indoor air dry-bulb temperature, indoor air dewpoint temperature, and outdoor air dry-bulb temperature. The BINS for Independent Variables have a Start Center Value, End Center Value, and Bin Width. The Bin Width determines the maximum distance a datapoint may be from the center of a bin to be included in that bin. Once the program executes, the user can see the selected independent variables in the *Datafile Array – String* (C).

4.2. TAB: Parse Into BINS #2

Figure 9 shows the *Parse Into BINS #2* tab. The *Raw Data COLS Array* (A) shows the three parameters being binned; in this example it is indoor air dry-bulb temperature, indoor air dewpoint temperature, and outdoor air dry-bulb temperature. The *Raw Independent Variables STATISTICS – Array* (B) gives a summary of the data being binned; it lists the mean, standard deviation, maximum, minimum and range for the three columns of data that are being binned. The *BIN RANGES* (C) shows the variation of the temperature bins that were defined on the previous tab. *ALL BINS* (D) shows all the combinations of the columns in (C) to form the bins; each row of the array (D) represents a possible bin. Figure 9E shows how each point in the raw data array (A) was assigned to a bin; it lists the calculated distance, bin number (as defined in (D)), and point number (row number minus 1 shown in (A)).

4.3. TAB: Parse Into BINS #3

Figure 10 shows the next tab, *Parse Into BINS #3*. The *ALL BINS* array (A) lists the bin assigned to each point; in this example each bin could have up to 5 points. In the *FILTERED by Dmax* array (B), the points in (A) which are greater than *Dmax* distance from their closest bin's center are removed. The final result is the list of points and their distance to the best

bin shown in (C). Figure 10D is just for reference and repeats the raw data array seen in previous tabs.

4.4. TAB: Parse Into BINS #4

Figure 11A shows the data points (rows of data in the raw datafile) for the points selected by the binning scheme. The *BINNED Data Statistics* – *Array* (B) provides a summary of all of the data columns in the raw datafile that were selected; the summary lists the mean, standard deviation, maximum, minimum and range for each column of the raw data that has been selected in the binning process.

4.5. TAB: SAVE Parsed BINS #5

Figure 12 shows the last tab in the program, *SAVE Parsed BINS #5* tab. The binned data is stored in the file indicated by the *SAVED Data BINS File Path Control* (A). The user should select the *SAVE File* button and *Create Report* button to write the binned data to a file and then create a snapshot of each tab saved in an archive file (B). The archive file, or report file, is a ZIP file saved in the indicated location.



Figure 8: Input datafile tab



Figure 9: Parse into bins #2 tab



Figure 10: Parse into bins #3 tab

stafile	0	CoriolisLiq_ID#	#1_Tsub_F	1527_CoriolisLiq_OD_T	[emp# CoriolisLi	q_OD#2_Tsub_F	Blower_Hea	it_Btu/h	Adj_RefSide_Q_Btu/h	%D	DIFF_Ref&Air_Q	COP	
SS-Evample RAW Datafile tyt		7.077053		125.458069	7.864183		1770.937184	l l	34528.059205	-0.6	548097	2.603393	
ss example_rorm_batanietxt	252	7.266963		125.951628	8.042804		1772.725072	2	34467.452633	-1.4	454331	2.604506	
	Construction of Construction	7.271685		126.108269	8.039924		1775.345488	3	34375.292122	-2.1	139634	2.607555	
Density TIME to PARSE (sec)		7.163249		126.232653	7.953353		1771.677588	3	34359.581212	-1.9	967676	2.599738	
est # Points)		7.281267		126.233490	8.063512		1772.383872	2	34321.018907	-2.2	241099	2.602982	
5		7.238166		126.205587	8.051282		1773.735024	ļ.	34366.311519	-1.5	587021	2.591339	
Retained Dmax		7.023744		125.546717	7.930316		1772.851316	j I	34706.260601	-0.9	937080	2.625234	
		7.078454		125.265450	8.028201		1773.076508	}	34803.224692	-0.3	576003	2.626425	
28 4.330		7.072573		125.159112	8.005735		1773.110628	5	34834.469127	-0.4	457192	2.633430	
		6.948107		125.093639	7.856171		1770.561864	ļ.	34873,906875	-0.1	280738	2.638112	
ta Statistica Cala Dana 00		6.907553		124.849757	7.903733		1770.619868	}	34927.030814	-0.3	340461	2.646124	
al # POINTS EPT out of rent Datafile Total # POINTS	В	BINNED Da	ata STATISTICS	124.729880 - Array	7.918111		1774.025044		34940.866160	-0.5	996649	2.669120	
al # POINTS EPT out of rent Datafile Total # POINTS	B ∂252	6.960281 ◀ BINNED Da	ata STATISTICS	- Array Header	7.918111	Header	1774.025044	Header	34940.866160	-0.5	996649	2.669120 Header	
al # POINTS EPT out of rent Datafile Total # POINTS 21 75	B	6.960281 ■ BINNED Da Header CoriolisLiq_ID	ata STATISTICS #1_Tsub_F 	- Array Header 1527_CoriolisLiq	7.918111 _OD_Temp#2_F	Header CoriolisLiq_OD	1774.025044 #2_Tsub_F	Header Blower_Hea	34940.866160 t_Btu/h	-0.3 Header Adj_RefSide	996649 e_Q_Btu/h	2.669120 Header %DIFF_Ref8./	Air_Q
al # POINTS EPT out of rent Datafile Total # POINTS 21 75	B	6.960281 ■ BINNED Da Header CoriolisLiq_ID4 6.94334	ata STATISTICS #1_Tsub_F Mean	- Array Header 1527_CoriolisLiq, 124.43	7.918111 _OD_Temp#2_F Mean	Header CoriolisLiq_OD 7.85495	1774.025044 #2_Tsub_F Mean	Header Blower_Hea 1773.29	34940.866160 t_Btu/h Mean	-0.3 Header Adj_RefSide 34981.1	996649 e_Q_Btu/h Mean	2.669120 Header %DIFF_Ref8.4 -2.29496	Air_Q Mean
al # POINTS EPT out of rent Datafile Total # POINTS 21 75	B € 252	6.960281 ■ BINNED Da Header CoriolisLiq_ID4 6.94334 0.249192	ata STATISTICS #1_Tsub_F Mean Std. Dev	- Array Header 1527_CoriolisLiq, 150271	7918111 _OD_Temp#2_F Mean Std. Dev	Header CoriolisLiq_OD 7.85495 0.185911	#2_Tsub_F Mean Std. Dev	Header Blower_Hea 1773.29 1.5781	34940.866160 t_Btu/h Mean Std. Dev	-0.3 Header Adj_RefSide 34981.1 482.881	2996649 E_Q_Btu/h Mean Std. Dev	2.669120 Header %DIFF_Ref8/4 -2.29496 1.78057	Air_Q Mean Std. Dev
al # POINTS EPT out of rent Datafile Total # POINTS 21 75	₿ 252	6.960281 BINNED Da Header Coriolistiq_ID 6.94334 0.249192 7.28127	ata STATISTICS #1_Tsub_F Mean Std. Dev MAX	- Array Header 1527_CoriolisLiq, 150271 5 126.233	_0D_Temp#2_F Mean Std. Dev MAX	Header CoriolisLiq_OD 7.85495 0.185911 8.07506	#2_Tsub_F Mean Std. Dev MAX	Header Blower Hea 1773.29 1.5781 1776.37	34940.866160 t_Btu/h Mean Std. Dev MAX	-0.3 Header Adj_RefSide 34981.1 482.881 35613.7	e_Q_Btu/h Mean Std. Dev MAX	2.669120 Header %DIFF_Ref8,4 -2.29496 1.78057 -0.280738	Air_Q Mean Std. Dev MAX
al # POINTS EPT out of rent Datafile Total # POINTS 21 75	₿ €] 252	6.960281 BINNED Da Header CoriolisLiq_ID4 6.94334 0.249192 7.28127 6.56706	#1_Tsub_F #1_Tsub_F Mean Std. Dev MAX MIN	- Array Header 1527_CoriolisLiq 124.43 1.50271 126.233	_0D_Temp#2_F Mean Std. Dev MAX MIN	Header CoriolisLiq_OD 7.85495 0.185911 8.07506 7.55508	#2_Tsub_F Mean Std. Dev MAX MIN	Header Blower Hea 1773.29 1.5781 1776.37 1770.56	34940.866160 t_Btu/h Mean Std. Dev MAX MIN	-0.3 Header Adj_RefSide 34981.1 482.881 35613.7 34321	s_Q_Btu/h Mean Std. Dev MAX MIN	2.669120 Header %DIFF_Ref8/4 -2.29496 1.78057 -0.280738 -5.00442	Air_Q Mean Std. Dev MAX MIN
al # POINTS EPT out of rent Datafile Total # POINTS 21 75	₿ €] 252	6.960281 BINNED Da Header CoriolisLiq_ID4 6.94334 0.249192 7.28127 6.56706 0.714203	#1_Tsub_F #1_Tsub_F Mean Std. Dev MAX MIN Range	- Array Header 1527_CoriolisLiq 124.43 1.50271 126.233 122.337 3.89626	7.918111 _OD_Temp#2_F Mean Std. Dev MAX MIN Range	Header CoriolisLiq_OD 7.85495 0.185911 8.07506 7.55508 0.519978	#2_Tsub_F Mean Std. Dev MAX MIN Range	Header Blower Hea 1773.29 1.5781 1776.37 1770.56 5.81064	34940.866160 t_Btu/h Mean Std. Dev MAX MIN Range	-0.3 Header Adj_RefSide 34981.1 482.881 35613.7 34321 1292.67	e_Q_Btu/h Mean Std. Dev MAX MIN Range	2.669120 Header %DIFF_Ref824 -2.29496 1.78057 -0.280738 -5.00442 4.72369	Air_Q Mean Std. Dev MAX MIN Range
al # POINTS EPT out of rent Datafile Total # POINTS 21 75	₿ € 252	6.960281 BINNED Da Header Coriolistiq_ID4 6.94334 0.249192 7.28127 6.56706 0.714203	#1_Tsub_F #1_Tsub_F Mean Std. Dev MAX MIN Range	- Array Header 1527_CoriolisLiq 124.43 1.50271 126.233 122.337 3.89626	7.918111 _OD_Temp#2_F Mean Std. Dev MAX MIN Range	Header CoriolisLiq_OD 7.85495 0.185911 8.07506 7.55508 0.519978	#2_Tsub_F Mean Std. Dev MAX MIN Range	Header Blower Hea 1773.29 1.5781 1776.37 1770.56 5.81064	34940.866160 t_Btu/h Mean Std. Dev MAX MIN Range	-0.3 Header Adj_RefSide 34981.1 482.881 35613.7 34321 1292.67	e_Q_Btu/h Mean Std. Dev MAX MIN Range	2.669120 Header %DIFF_Ref8/4 -2.29496 1.78057 -0.280738 -5.00442 4.72369	Air_Q Mean Std. Dev MAX MIN Range
tal # POINTS EPT out of rrent Datafile Total # POINTS 21 75	₿ 252	6.960281 BINNED Da Header CoriolisLiq_ID4 6.94334 0.249192 7.28127 6.56706 0.714203 <	#1_Tsub_F Mean Std. Dev MAX MIN Range	- Array Header 1527_CoriolisLiq 124.43 1.50271 126.233 122.337 3.89626	_OD_Temp#2_F Mean Std. Dev MAX MIN Range	Header CoriolisLiq_OD 7.85495 0.185911 8.07506 7.55508 0.519978	1774.025044 #2_Tsub_F Mean Std. Dev MAX MIN Range	Header Blower Hea 1773.29 1.5781 1776.37 1770.56 5.81064	34940.866160 t_Btu/h Mean Std. Dev MAX MIN Range	-0.5 Header Adj_RefSide 34981.1 482.881 35613.7 34321 1292.67	e_Q_Btu/h Mean Std. Dev MAX MIN Range	2.669120 Header %DIFF_Ref8,4 -2.29496 1.78057 -0.280738 -5.00442 4.72369	Air_Q Mean Std. D MAX MIN Range
al # POINTS PT out of rent Datafile Total # POINTS 21 75 I # of BINS # RAW Data ROWS	B	6.960281 BINNED Da Header Coriolistiq_ID3 6.94334 0.249192 7.28127 6.56706 0.714203 4	#1_Tsub_F Mean Std. Dev MAX MIN Range	- Array Header 1527_CoriolisLiq 124.43 1.50271 126.233 122.337 3.89626	7.918111 _OD_Temp#2_F Mean Std. Dev MAX MIN Range	Header CoriolisLiq_OD 7.85495 0.185911 8.07506 7.55508 0.519978	1774.025044 #2_Tsub_F Mean Std. Dev MAX MIN Range	Header Blower Hea 1773.29 1.5781 1776.37 1770.56 5.81064	34940.866160 t_Btu/h Mean Std. Dev MAX MIN Range	-0.5 Header Adj_RefSide 34981.1 482.881 35613.7 34321 1292.67	e_Q_Btu/h Mean Std. Dev MAX MIN Range	2.669120 Header %DIFF_Ref824 -2.29496 1.78057 -0.280738 -5.00442 4.72369	Air_Q Mean Std. Dev MAX MIN Range
al # POINTS EPT out of rent Datafile Total # POINTS 21 75 1 # of BINS # RAW Data ROWS 669 75	₿	6.960281 BINNED Da Header Coriolistiq_ID4 6.94334 0.249192 7.28127 6.56706 0.714203 4	ata STATISTICS #1_Tsub_F Mean Std. Dev MAX MIN Range	- Array Header 1527_CoriolisLiq 124.43 1.50271 126.233 122.337 3.89626	_OD_Temp#2_F Mean Std. Dev MAX MIN Range	Header CoriolisLiq_OD 7.85495 0.185911 8.07506 7.55508 0.519978	#2_Tsub_F Mean Std. Dev MAX MIN Range	Header Blower Hea 1773.29 1.5781 1776.37 1770.56 5.81064	34940.866160 t_Btu/h Mean Std. Dev MAX MIN Range	-0.5 Header Adj_RefSide 34981.1 482.881 35613.7 34321 1292.67	e_Q_Btu/h Mean Std. Dev MAX MIN Range	2.669120 Header %DIFF_Ref8/4 -2.29496 1.78057 -0.280738 -5.00442 4.72369	Air_Q Mean Std. Dev MAX MIN Range

Figure 11: Parse into bins #4 tab



Figure 12: Save parsed bins #5 tab

5. FDD Self Training POLYNOMIAL FIT Ver07.vi

This program will take any raw, tab delimited text file arranged in columns and fit selected columns (dependent data) to three columns designated as the independent data. The program will generate a 1st order, 2nd order, or 3rd order polynomial fit, perform outlier removal, and perform a backward elimination F-Test to reduce the number of coefficients. The resulting polynomial fit coefficients and some fit statistics are saved in a tab delimited text output file.

5.1. TAB: Select COL Data File #1

Figure 13 shows the first tab for the polynomial fit program. The *Input File* (B) is a tab delimited text file of columnar data. The *Saved Coefficient File* (C) is another tab delimited text file for the output of the fit coefficients; the order of the fit is selected on the left at (A); a 1st order, 2nd order, or 3rd order polynomial fit may be performed. The form of the polynomial fit is also indicated at (A). In this example, the binned datafile of fault free data is used; this file contains the three independent variables: 1) x1-indoor air dry-bulb temperature, 2) x2-indoor air dewpoint temperature, and 3) x3-outdoor air dry-bulb temperature. All of the other columns of data in the file will be fit as dependent variables.

5.2. TAB: RAW DATA & CALCS #2

Figure 14 shows the *RAW DATA & CALCS #2* tab; here the raw datafile is read into a string array for the user to inspect. The user also selects whether or not the file contains a header label row at (B). This tab is intended to allow the user to abort the running of the program if the data array (A) does not look as expected.

5.3. TAB: RAW DATA & CALCS #3

Figure 15 shows a snapshot of the next tab that allows the user to perform some calculations to create a new dependent variable from a mathematical combination of the existing variables in the datafile. The array at (A) shows the header labels associated with each column of data. An array of calculated variables is created at (B). In the Array of Calculated Variables (B), the two (or more) columns indicated by the DATA COLUMNS array may have any of the operations performed as indicated in the selection list; the user selects the two data columns for the mathematical operation from the numeric index of header labels at (A) and selects an operation from the list. The user may also enter a NEW HEADER LABEL for the newly defined variable. Multiple, new variables may be defined in this way. The newly defined variables may also be used to define another new variable if the entry of the mathematical operation defining the first new dependent variable is performed before the subsequent dependent variable uses it in its definition. Newly created dependent variables are appended to the columnar input datafile and their numeric index will equal the last index indicated at (A) plus their position in the definition sequence indicated at (B). For example, if there were 20 columns (0 to 19) indicated at (A), then any newly defined dependent variable created in the (B) array would be appended starting at index 20 (or column 21). This newly defined dependent variable could then be referenced in array (B) by the number 20 in subsequent definitions in array (B).

5.4. TAB: Select Parameters #4

Figure 16 shows the *Select Parameters* #4 tab. Here the user can see the original datafile with the newly defined, and appended, columns at (B). With the raw data input in tab #1, reviewed

in tab #2, and new dependent variables defined in tab #3, the user now selects the three independent variables and dependent variables from the data in *RAW & CALCULATED Data Array* – *String*, array (B). The array (C) holds the independent variables' column indices that are selected from the columns in array (B). Array (D) holds the column indices of all of the dependent variables selected from array (B). The column indices input into arrays (C) and (D) are also found in array (E); array (E) is just a copy of all of the column header labels from array (B). Once all of the variables have been specified for the fits, the user may proceed to perform the fits by running the program.

5.5. TAB: FIT and STATS #5

Figure 17 shows the *FIT and STATS* #5 tab. This snapshot shows *Independent VARIABLES*, x1, x2, x3 – *Numeric Data Array*, array (A); the independent variables that come from the raw data array with appended variables in the previous tab. Array (B), *Dependent VARIABLE(S) Numeric Data Array*, is the dependent variables from the same aforementioned array on the previous tab. Array (C) is the statistical fit *H*-*Matrix* which lists y0 through y19 as calculated from the dependent variables in array (A). Each row in array (D) represents the fit coefficients (a0 to a19) beginning with row index 0 to 19 for each dependent variable column shown in array (B). The Mean Squared Error of the fit for each of the dependent variables is appended as the last column for all of the dependent variables of array(B).

5.6. TAB: Remove Outliers #6

Figure 18 shows a snapshot of the tab used to setup the outlying data removal. The array indices with the same color background are associated with the same dependent variable. This tab shows the fit residual (value minus fit-value) for each dependent variable shown in Figure 17, array (B). Array (B), *STATS Columns for each Dependent Variable's Residuals*, shows the statistics for the residuals listed in array (A); each column of array (B) summarizes the fit statistics, listed to the right of array(B), for each column of dependent variables (array (B) of Figure 17). The constant value shown at (C) is the interquartile range multiplier (*k*) (Wikipedia 2020a); the interquartile range for each dependent variable is listed in row index 5 of array (B). Any residual that lies outside of the range of the Lower Bound and Upper Bound (in this example 2.2 times the interquartile range) will be flagged as an outlier. A value is flagged as an outlier if it is *k* times below or *k* times above the 25th or 75th percentile, respectively. All this assumes a normal distribution of dependent variable residuals.

5.7. TAB: Remove Outliers #7

Figure 19 shows the *Remove OUTLIERS* #7 tab. Here the user can get a color-coded visual indication of the dependent data that were identified as outliers. Array indices with the same background color represent the same or related variables. The red LED in array (A) indicates that a dependent variable point was removed as an outlier because its fit residual was more than *k* times the interquartile range. Array (B) indicates the line number of the raw data with appended variables that had an outlier, while array (C) shows you the column numbers and header labels of any dependent variable with outliers. Arrays (B) and (C) indicate if any row (line) or column in Array (A) had a red LED; if any LED was red in the line (B) or column (C) they will show it. Constant (D) allows you to select a particular dependent variable's residual histogram plot (which should be normally distributed about zero). The skewness and kurtosis are calculated to the right of the histogram, and a plot of the residual is shown at (E).

at (E) allows the user to see the distribution of residuals and verify the line numbers that had outliers.

5.8. TAB: FIT and STATS #8

Figure 20 shows three arrays that hold summary statistics for all of the dependent variables with the outliers removed. The mean, standard deviation, maximum, minimum, and range are show for all the dependent variables.

5.9. TAB: BACKWARD Elimination #9

Figure 21 shows the tab page that does the setup for a backward elimination technique to reduce the number of coefficients used in the dependent variable fits. If the user wishes to perform a backward elimination, the button at (E) is selected. This will run the code associated with removing those coefficients from the full-fit that, when removed, cause less than a certain percentage increase (A) in the sum of squared errors, SSE. In this example 17 % is selected as the threshold for removing a coefficient from the full model. The sum of squared error ratio equals the SSE of the reduced model divided by the SSE of the full model. Array (B) shows the SSE ratio for each removal of a coefficient; the row index (44 in this example) is the number of the fit equation, and the column index indicates the coefficient that was removed (0-3 in case of a 1st order fit). Array (B) in this example, looking at row index 44, column index 3, the value of the SSE ratio was 1.167; this means that the SSE of the reduced model was that many times greater than the SSE of the full model with coefficient zero (a3 removed). Coefficient (a3) should be removed because its SSE ratio was less than 1.17 set at (A). Array (C) is a visual indicator of those coefficients that could be removed; green means they stay, red means they can be removed. In this example, if removal of the coefficient causes less than a 17 % increase in the SSE, it is thrown out. All the array index indicators with the same color background (yellow here) are associated with the same dependent variable; see (F), (B), and (C).

5.10. TAB: Debug Page Coefficient MODELS #10

Figure 22 is a snapshot of a tab used to debug and step through the program while the backward elimination and correlation fits are done.

5.11. TAB: Reduced Coefficient MODELS #11

Figure 23 shows the final polynomial fits for each of the dependent variables that were selected. The array also includes the mean squared error (MSE), the fit degrees of freedom (DOF), the Student T-Value, and the 95 % confidence interval on a mean value calculated using the fit. The saved datafile input on the *Select COL Data File #1* tab includes this exact information in the form of a tab delimited text file.



Figure 13: Select COL data file #1 tab

					-				
	Debug Page Coefficient M	ODELS #10	1	x.	Reduced	Coefficient MODELS #11			X
y = a0*y0 + a1*y1 ++ai*yi	Select COL Data File #1	RAW DATA & CALCS #2	RAW DATA & CALCS #3	Select Parameters #4	FIT and STATS #5	Remove OUTLIERS #6	Remove OUTLIERS #7	FIT and STATS #8	BACKWARD Elimination #
Select Polynomial Order where i =0 to 19.	A								
3rd Order 🕤 y1=x1;	A	RAW Data Array - STRING							
yz=x2; 1st Order y3=x3;	E 0	Filename	2329_RawVDC	2330_RawVDC	2331_RawV	DC ID_Xit	ron_VAC	ন	
Full Model # of Coef.		Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.51	.6000		
20 y5=x1*x3;	0	Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.21	1000		
y6=x2*x3; v7=x1**2; 2nd Order		Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.19	8000		
y8=x2**2;	D	Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.26	50000		
y9=x3**2;	D Na Usadas Labela?	Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.14	8000		
y10=x1*x2*x3;	No Header Labers:	Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.26	5000		
y11=x1*x2**2; y12=x1*x3**2;		Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.41	.6000		
y13=x2*x3**2;		Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.27	7000		
y14=x1 *2 x2; 5rd Order y15=x1**2*x3;	# of Points (ROWS)	Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.40	8000		
y16=x2**2*x3;	21	Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.38	32000		
y18=x2**3;		Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.48	32000		
y19=x3**3;		Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.68	35000		
		Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.54	7000		
		Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.63	3000		
		Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.70	9000		
		Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	241.00	08000		
DO Backward Elimination to REDUCE # Coefficients?		Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.82	21000		
to hebbee # coefficients		Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.93	9000		
		Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	241.02	29000		
9		Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	241.00	00000		
General Linear FIT error out		Example_RAW_Datafile.txt	5.001528	5.000551	5.002036	240.99	3000		
status code									
2 0									
source									
* ·								v	
		I					Þ		
J									

Notes Internet and 90 G 934

Figure 14: Raw data and calculations #2 tab



Figure 15: Raw data and calculations #3 tab



Figure 16: Select parameters #4 tab

Multivariate Polynomial I	FIT with 3	3 Indep	endent V	ariable	s									
	Debug Page C	Coefficient MODE	LS #10					Reduc	ed Coefficient N	AODELS #11				
y = a0*y0 + a1*y1 ++ai*yi	Select COL Da	ita File #1	RAW DATA & CALC	S #2 RAW	DATA & CALCS #3	Se	lect Parameters #4	FIT and STATS #5	Remove 0	OUTLIERS #6 R	emove OUTLIER	S #7 FIT and	STATS #8	BACKWARD Elimination #9
Select Polynomial Order where i =0 to 19. y0=1;								Dependent Variable	HEADER LABEI	LS - String				
1st ORDER y1=x1, y2=x2; 1st Order		Indep. Varia	DIE HEADER LABELS	- String			36	TOTAL Capacity	(Btu/h)	SHR	ODV	apSV_Suph_F	ODVap	oSV_Tsat_F
y3=x3;	0	Inlet TC G	rid 1703_VDC_K	ahn OD Air AV	G DB		B	Dependent VARIA	BLE(S) Numeric	Data Array				
y4=x1*x2; 4 y5=x1*x3;	A	Independer	t VARIABLES, x1, x2,	x3 - Numeric D)ata Array		0	34751.8		0.434959	12.2	641	61.69	7
y6=x2*x3; y7=y1**2; 2nd Order	()	84.3613	71.9104	115.425	0	*		34968.7	1	0.431601	12.8	716	61.68	17
y8=x2**2;		84.3112	71.9387	116.275	0		30	35110.8		0.432009	12.5	834	61.70	72
y9=x3**2;	I	84.348	71.9614	116.49	0			35035.7		0.432268	12.5	408	61.75	31
y10=x1*x2*x3;		84.3559	71.9387	116.378	0			35090.2		0.432769	12.5	059	61.778	36
y11=x1 x2 2; y12=x1*x3**2;		84.3826	71.9444	116.505	0			34911.7		0.43007	12.5	869	61.74	29
y13=x2*x3**2;		84.2992	71.9387	116.31	0			35031.5		0.429139	13.2	716	61.66	12
y15=x1**2*x3;		84.407	71.967	115.274	0			35003.7		0.429709	13.4	523	61.67	14
y16=x2**2*x3; v17=v1**3;		84.4514	71.9708	115.261	0			34993.7		0.430188	13.5	214	61.72	76
y18=x2**3;		84,4688	71.9765	115.129	0			34971.8		0.428148	13.4	995	61.809	92
y19=x3**3;		84,4227	71.9916	114.916	0			35045.9		0.428114	13.4	051	61.86	53
		84.47	72.0048	114.709	0			35289.1		0.427172	13.2	638	61.89	58
		84.5065	72.0369	114.55	0			35398.1		0.425804	13.1	147	61.92	13
		84.5071	72.0614	114.389	0	_		35509.3		0.425941	13.1	402	61.89	58 💌
DO Redeved Discipution		84.6551	72.1331	114.07	0	v	D	FUILI Dataset Corre	lation Coefficie	onte and MSE in last	column of the ro	www.(a0, a1, a2	MSEL	
to REDUCE # Coefficients?	C	H Matrix					A 16	1.91199205.5	1 01650505 . 2	1 72242105 . 2	5 50602025 - 1	0.26000765.2		
	Ê0	1	84.3613	71.9104	115.425			1 73520005-2	1 3700483E-2	-1 52014275-2	3.0232077E-3	1.9020023E-6	0.0000000	
		1	84.3112	71.9387	116.275		0	1 59463425+2	1.5767125E-1	-1.53014576-2	-4 2100126E-1	6 52067625-0	0.00000000	-0
General Linear EIT error out	0	1	84.348	71.9614	116.49	i II		1 7718888E+1	1 9501471E-1	4 2303526F-1	-2.4808184F-2	2 5767939E-3	0.0000000	-0
status code	1	1	84.3559	71.9387	116.378	1		1.2217720E+2	-6.1634743E-1	-3.5928806F-1	7.7188874F-1	5.5626004E-3	0.00000000	+0
1 0		1	84.3826	71.9444	116.505			-1.9968055E+1	8,2938969E-1	-7.0400945E-1	7.3068959E-2	4.3969005E-3	0.0000000E-	+0
source		1	84.2992	71.9387	116.31			1.2017063E+2	-1.4709962E+(0 7.3052152E-1	7.2842900E-1	4.0887719E-3	0.0000000E-	+0
		1	84.407	71.967	115.274			1.1599601E+2	-7.0421860E-1	-7.1605032E-2	7.2476516E-1	3.8602902E-3	0.0000000E-	+0
		1	84.4514	71.9708	115.261			4.7055092E+1	4.4734110E-1	1.4405147E+0	-8.3706689E-3	1.2233390E-2	0.0000000E-	+0
		1	84.4688	71.9765	115.129			1.5210825E+2	5.4379250E-1	-1.1479183E+0	-2.9461424E-1	5.0157139E-2	0.0000000E-	+0
•		1	84.4227	71.9916	114.916			1.3865063E+2	3.9188476E-1	-1.6339544E+0	-2.9166416E-1	4.9431045E-2	0.0000000E-	+0
		1	84.47	72.0048	114.709			1.3457082E+1	1.5191771E-1	4.8603118E-1	-2.9496169E-3	1.4064719E-3	0.0000000E-	+0 🔻
		1	84.5065	72.0369	114.55	v		4	×		×	*		
		het.				1								

Figure 17: Fit and stats #5 tab

Multivariate Polynomial	FIT with	3 Independen	t Variables									
	Debug Page	Coefficient MODELS #10					Reduce	d Coefficient MODELS #11				
v − a0%0 + a1%0 + +ai%0	Select COL I	Data File #1 RAW DATA 8	CALCS #2 RAW DA	ATA & CALCS #3	Select Paramete	ers #4 FT	T and STATS #5	Remove OUTLIERS #6	Remove OU	TLIERS #7	FIT and STATS #8	BACKWARD Elimination #9
Select Polynomial Order where i =0 to 19.					-							
y0=1;		Dependent Variable HEADE	R LABELS - String 2					A				
y2=x2; 1st Order	36	TOTAL Capacity (Btu/h)	SHR	ODVapSV_Sup	oh_F ODV	VapSV_Tsat_F		Assumes Residual	s are normali	y distribut	ed about zero.	
Full Model # of Coef.	A	Residuals for all Dependent	Variables (Value Minus Fi	t Value)					Outlier IQR Mu	iltiplier (k)		
4 y5=x1*x3;		-1.4226628E+2	4.8304961E-3	-8.3428025E-1	-3.0	771841E-2		G	* 2.2			
y6=x2*x3; y7=x1**2; 2nd Order		2.9256199E+1	2.8411000E-5	1.9062864E-1	-2.7	209629E-2		Lov	erBound = 01	- k * IOR		
y8=x2**2;	36	8.2932290E+1	-3.7441146E-4	2.4025832E-2	-1.3	096945E-2		Up	perBound = Q3	+ k * IQR		
y9=x3**2;		4.5015257E+1	-2.3323904E-4	-1.0268103E-1	3.80	088170E-2						
y10=x1*x2*x3;		5.5556183E+1	-3.9710191E-4	-7.8455589E-2	5.91	132707E-2		A DIALES	olumns for eac	n Depender	T Variable's Residuals	
y11=x1 x2 2; y12=x1*x3**2;		-1.7519310E+1	-1.4428783E-3	-7.7083018E-2	3.72	239153E-2		247.1	4.830m	379.3m	Max Residual	
y13=x2*x3**2; x14=x1**2*x2: 3rd Order		1.9514933E+0	-2.9556734E-4	1.8727637E-1	-1.0	311047E-1		D - 36 - 169.1	-2.435m	-834.3m	Min Residual	
y15=x1**2*x3;		-7.6847054E+1	-2.4360463E-4	3.6138726E-1	-1.0	1347720E-1		1.951	-243.6u	24.03m	Median Residual	
y16=x2**2*x3; y17=y1**3;		-1.0682995E+2	4.8238736E-4	3.7932561E-1	-5.6	401682E-2		-77.90	-702.8u	-103.4m	Quartile 1 (25%)	
y18=x2**3;		-9.5956893E+1	-4.6035671E-5	2.9591075E-1	2.25	515093E-2		56.31	322.6u	160.9m	Quartile 3 (75%)	
y19=x3**3;		-8.1066141E+1	9.3899266E-5	1.2524812E-1	5.86	548205E-2		134.2	1.025m	264.3m	InterQuartileRange	(3-1)
		7.8683594E+1	-3.7782472E-4	-4.1604693E-2	6.45	559186E-2		-373.2	-2.959m	-684.8m	Outlier LowerBound	l .
		1.5382578E+2	-8.9193010E-4	-2.2276238E-1	7.55	521078E-2		351.6	2.578m	742.3m	Outlier UpperBound	i .
		8.8621501E+0	-7.3595513E-4	-2.4845508E-1	-1.7	040905E-2		0.000	0.000	0.000	-	
		2.4706865E+2	-2.4352787E-3	1.5212111E-1	4.72	204262E-3		0.000	0.000	0.000	-	
DO Backward Elimination to REDUCE # Coefficients?		-3.8539579E+1	6.7368360E-4	-1.7959502E-1	6.96	584250E-2		0.000	0.000	0.000	4	
		3.1599303E+1	-6.9174826E-4	-6.0727955E-2	-3.2	083611E-2		0.000	0.000	0.000	4	
		-1.9155226E+1	2.6928823E-4	-1.0547594E-1	7.62	242796E-3		10.000	0.000	10.000		
		5.8590287E+1	-1.0746980E-3	1.0192003E-1	-2.3	677081E-2						
General Linear FIT error out		-1.6913911E+2	1.1893427E-3	8.2271918E-2	-2.74	441763E-2						
status code		-4.6021638E+1	1.6727650E-3	5.1005318E-2	-3.4	214171E-3						
<u>✓</u> <u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u></u>		0.0000000E+0	0.0000000E+0	0.0000000E+0	0.00	000000E+0						
source		0.000000E+0	0.0000000E+0	0.0000000E+0	0.00	000000E+0						
^		0.000000E+0	0.0000000E+0	0.000000E+0	0.00	000000E+0						
		4										
. Д	1											

Figure 18: Remove outliers #6 tab



Figure 19: Remove outliers #7 tab

	Debug Page	Coefficient MOD	ELS #10							Reduced Co	efficient MODE	LS #11					
y = a0*y0 + a1*y1 ++ai*yi	Select COL I	Data File #1	RAW DATA 8	k CALCS #2	RAW DATA	& CALCS #3	Select Pa	rameters #4	FIT and ST	ATS #5	Remove OUTL	ERS #6 F	Remove OUTLI	ERS #7	FIT and STATS #	8 BACH	KWARD Eliminat
Polynomial Order where i =0 to 19. y0=1;																	
d Order y1=x1; y2=x2; 1st Order	(1)	STATISTICS -	Array														
yd=xd; Andel # of Conf	÷)o	Header		Header		Header		Header		Header		Header		Header		Header	
y4=x1*x2;		Inlet TC Grid	AVG Temp	1703_VDC_K	ahnDewTem	OD Air AVG	DB (F)	ID_Xitron_Am	ps	ID_Xitron_W	/atts	OD_Xitron_A	Amps	OD_Xitron_	Watts	1500_ODLiq	SV_TempF
y6=x2*x3;		84.638	Mean	72.3028	Mean	114.332	Mean	2.24071	Mean	519.721	Mean	14.2525	Mean	3344.18	Mean	124.603	Mean
y/=x1^2; 2nd Order y8=x2**2;		0.320815	Std. Dev	0.470304	Std. Dev	1.70082	Std. Dev	0.00254556	Std. Dev	0.462514	Std. Dev	0.31285	Std. Dev	70.1046	Std. Dev	1.49076	Std. Dev
y9=x3**2;		85.145	MAX	73.0144	MAX	116.505	MAX	2.24618	MAX	520.625	MAX	14.647	MAX	3431.53	MAX	126.467	MAX
y10=x1*x2*x3; y11=x1*x2**2		84.2992	MIN	71.9104	MIN	112.012	MIN	2.23696	MIN	518.922	MIN	13.8121	MIN	3244.55	MIN	122.591	MIN
y12=x1*x3**2; y13=y2*y3**2;		0.845825	Range	1.10398	Range	4.49239	Range	0.00922	Range	1.703	Range	0.8349	Range	186.98	Range	3.87649	Range
y14=x1**2*x2; 3rd Order y15=x1**2*x3;		STATISTICS - A	Array 2														
y16=x2**2*x3;	18	Header		Header		Header		Header		Header		Header		Header		Header	
y17=X1 3, y18=x2**3;	100	1510_ODVap	SV_TempF	1526_Corioli	s_IDSide_Te	1527_Corioli	s_ODSide_Te	1600_IDVap_T	empF	1601_IDLiq_	TempF	1602_ID2pha	se_Cool#1_	1603_ID2Ph	ase_Cool#2_	1608_Comp	Suct_TempF
y19=x3**3;		74.9884	Mean	124.239	Mean	124.43	Mean	71.1368	Mean	123.589	Mean	63.9478	Mean	64.4215	Mean	81.4523	Mean
		0.494267	Std. Dev	1.46123	Std. Dev	1.50271	Std. Dev	0.681692	Std. Dev	1.41159	Std. Dev	0.356454	Std. Dev	0.309115	Std. Dev	0.318594	Std. Dev
		75.7092	MAX	125.95	MAX	126.233	MAX	72.1204	MAX	125.287	MAX	64.495	MAX	64.9204	MAX	81.9038	MAX
		73.9611	MIN	122.181	MIN	122.337	MIN	69.9124	MIN	121.621	MIN	63.6051	MIN	64.1347	MIN	80.5558	MIN
DO Backward Elimination to REDUCE # Coefficients?		1.74815	Range	3.76905	Range	3.89626	Range	2.20799	Range	3.66599	Range	0.889907	Range	0.785762	Range	1.34799	Range
		STATISTICS - /	Array 3														
	62	Header		Header		Header		Header		Header		Header		Header		Header	
I Linear FIT error out		Blower_Heat	_Btu/h	Adj_RefSide_	Q_Btu/h	COP		SCFM/TON		ODAir_DT(F)	LiqLine_DP(p	psid)				
code	T	1773.29	Mean	34981.1	Mean	2.71688	Mean	422.43	Mean	17.0579	Mean	6.89451	Mean	0	Mean	0	Mean
0		1.5781	Std. Dev	482.881	Std. Dev	0.128068	Std. Dev	12.0065	Std. Dev	0.285247	Std. Dev	0.0659719	Std. Dev	0	Std. Dev	0	Std. Dev
		1776.37	MAX	35613.7	MAX	2.90834	MAX	433.77	MAX	17.6002	MAX	7.00705	MAX	0	MAX	0	MAX
		1770.56	MIN	34321	MIN	2.59134	MIN	404.411	MIN	16.5381	MIN	6.79004	MIN	0	MIN		MIN
		5.81064	Range	1292.67	Range	0.317002	Range	29.3586	Range	1.06208	Range	0.217011	Range	0	Range	0	Range

Multivariate Polynomial FIT with 3 Independent Variable

Figure 20: Fit and stats #8 tab



Figure 21: Backward elimination #9 tab

	Select COL Data File	#1 RAW DATA	& CALCS #2 RAW	DATA & CALCS #3	Select Parameters #4	FIT	and STATS #5	Remove OUTLIERS	#6 Remove OU	TLIERS #7	FIT and STATS #8	BACKWARD Eliminat
$v = a0^{*}v0 + a1^{*}v1 ++ai^{*}vi$	Debug Page Coefficie	ent MODELS #10					Reduced	d Coefficient MODELS #	#11			
olynomial Order where i =0 to 19.							1					
oppen vi=xi:		H-Matrix AFTER ZER	O Value COLS ADDED				Depe	ndent Variable MATRIX	Fitted Coe	efficients MATI	RIX	
y2=x2; 1st Order		1.0000000E+0	8.43613270E+1	7.19104250E+1	1.15425130E+2		Asso	ciated with H-Matrix	Associated	d with H-Matri	x w/ MSE at END	
y3=x3;		1.0000000E+0	8.43112140E+1	7.19387320E+1	1.16274781E+2	-101	0	0		0		
y4=x1*x2;	0	1.0000000E+0	8.43479570E+1	7.19613780E+1	1.16489878E+2	-111		0		0	i	
y5=x1 x5; y6=x2*x3;		1.0000000E+0	8.43558740E+1	7.19387320E+1	1.16378100E+2	-111	Dependent	0		0		
y7=x1**2; 2nd Order	10 sec Pause	1.0000000E+0	8.43825670E+1	7.19443930E+1	1.16504827E+2		Variable INDEX	0	Reduced Model	0		
y9=x3**2;	STEP to Debug?	1.0000000E+0	8.42992180E+1	7.19387320E+1	1.16309785E+2		65	0	# of Coefficients	0		
v10-x1*v2*x3·		1.0000000E+0	8.44069560E+1	7.19670390E+1	1.15273866E+2			0		0	1	
y11=x1*x2**2;		1.0000000E+0	8.44514230E+1	7.19708130E+1	1.15261351E+2		ERROR CODE	0		0	1	
y12=x1*x3**2; y13=x2*x3**2:		1.0000000E+0	8.44688130E+1	7.19764750E+1	1.15128937E+2		0	0		0	1	
y14=x1**2*x2; 3rd Order		1.0000000E+0	8.44227130E+1	7.19915720E+1	1.14915788E+2			0		0		
y15=x1**2*x3; y16=x2**2*x3;		1.0000000E+0	8.44699670E+1	7.20047820E+1	1.14709430E+2			0		0		
y17=x1**3;		1.0000000E+0	8.45064520E+1	7.20368640E+1	1.14549721E+2			0		0		
y10=x2 5; y19=x3**3;		1.0000000E+0	8.45070760E+1	7.20613970E+1	1.14388601E+2	v		0 🗸		0	v	
		4				2		C	_	9		
		Coefficients with MSE	, Number of Points, Stu	dents T-Value, 95% Co	nfidence Interval appende	ed in last	columns-NUME	RIC ARRAY				
	0	1.84484344E+0	6.72041891E-3	-4.20123253E-3	1.14429623E-3	3.310	29138E-6	1.7000000E+1	2.10981558E+0	3.8386419	6E-3	
		1.84730291E+2	4.14772210E+0	-8.71659982E-1	4.10724752E-1	1.250	40452E-1	1.7000000E+1	2.10981558E+0	7.4605313	8E-1	
DO Backward Elimination		3.55947391E+0	-7.12081040E-2	-2.71539603E-2	1.63411919E-1	3.543	37513E-4	1.7000000E+1	2.10981558E+0	3.9714862	3E-2	
to REDUCE # Coefficients?		1.61170600E+3	-2.81325986E+1	0.0000000E+0	3.59791061E+1	1.525	27661E+1	1.8000000E+1	2.10092204E+0	8.2051068	7E+0	
		1.42145335E+2	-1.44573878E+0	3.44722366E-1	6.98819724E-1	7.675	59134E-3	1.7000000E+1	2.10981558E+0	1.8484189	7E-1	
		1.77182214E+2	3.52688724E-1	-1.10536267E+0	-4.55899490E-1	5.968	20782E-2	1.7000000E+1	2.10981558E+0	5.1542616	8E-1	
		1.84244035E+2	-1.49286820E+0	0.0000000E+0	5.80312995E-1	1.304	63869E-2	1.8000000E+1	2.10092204E+0	2.3996895	7E-1	
Linear FIT error out		and a first state of the local s	ESTRATION PROPERTY AND INCOME.	0.0000000000	C 543044005 4	1 1 25	31368F-2	1 70000005 . 1	2.10981558E+0	2.2480316	9E-1	
Linear FIT error out	r I	1.65085421E+2	-1.56530299E+0	2.35566070E-1	0.54204100E-1	11.155		1.70000002+1	1			
Linear FIT error out code 0		1.65085421E+2 1.45761887E+2	-1.56530299E+0 6.42801736E-1	-1.01738828E+0	-4.85171020E-1	1.124	25925E-1	1.70000000E+1	2.10981558E+0	7.0742064	5E-1	
Linear FIT error out code 0		1.65085421E+2 1.45761887E+2 1.79277870E+2	-1.56530299E+0 6.42801736E-1 -1.19404639E+0	-2.32739462E-1	-4.85171020E-1 5.44033826E-1	1.135	25925E-1 79757E-2	1.7000000E+1 1.7000000E+1 1.7000000E+1	2.10981558E+0 2.10981558E+0	7.0742064	5E-1 4E-1	
Linear FIT error out code 0		1.65085421E+2 1.45761887E+2 1.79277870E+2 2.58898969E+1	-1.56530299E+0 6.42801736E-1 -1.19404639E+0 2.39797558E-2	2.35566070E-1 -1.01738828E+0 -2.32739462E-1 5.73174782E-1	6.54204100E-1 -4.85171020E-1 5.44033826E-1 -4.73520490E-2	1.135 1.124 1.564 1.699	25925E-1 79757E-2 21227E-3	1.7000000E+1 1.7000000E+1 1.7000000E+1 1.7000000E+1	2.10981558E+0 2.10981558E+0 2.10981558E+0	7.0742064 2.6392077 8.6969768	5E-1 4E-1 2E-2	
Linear FIT error out code 		1.65085421E+2 1.45761887E+2 1.79277870E+2 2.58898969E+1 2.60607071E+1	-1.56530299E+0 6.42801736E-1 -1.19404639E+0 2.39797558E-2 -4.92024429E-2	2.35566070E-1 -1.01738828E+0 -2.32739462E-1 5.73174782E-1 6.18724678E-1	6.54204100E-1 -4.85171020E-1 5.44033826E-1 -4.73520490E-2 -1.93321755E-2	1.124 1.564 1.699 1.576	25925E-1 79757E-2 21227E-3 78684E-3	1.7000000E+1 1.7000000E+1 1.7000000E+1 1.7000000E+1 1.7000000E+1	2.10981558E+0 2.10981558E+0 2.10981558E+0 2.10981558E+0 2.10981558E+0	7.0742064 2.6392077 8.6969768 8.3778192	5E-1 4E-1 2E-2 8E-2	
Linear FIT error out code 0		1.65085421E+2 1.45761887E+2 1.79277870E+2 2.58898969E+1 2.60607071E+1 1.52108251E+2	-1.56530299E+0 6.42801736E-1 -1.19404639E+0 2.39797558E-2 -4.92024429E-2 5.43792501E-1	2.3556070E-1 -1.01738828E+0 -2.32739462E-1 5.73174782E-1 6.18724678E-1 -1.14791828E+0	6.54204100E-1 -4.85171020E-1 5.44033826E-1 -4.73520490E-2 -1.93321755E-2 -2.94614237E-1	1.124 1.564 1.699 1.576 5.015	25925E-1 79757E-2 21227E-3 78684E-3 71387E-2	1.7000000E+1 1.7000000E+1 1.7000000E+1 1.7000000E+1 1.7000000E+1 1.7000000E+1	2.10981558E+0 2.10981558E+0 2.10981558E+0 2.10981558E+0 2.10981558E+0 2.10981558E+0	7.0742064 2.6392077 8.6969768 8.3778192 4.7250985	5E-1 4E-1 2E-2 8E-2 6E-1	

Figure 22: Debug page coefficient models #10 tab

$\begin{array}{c} y = a0^{*}y0 + a1^{*}y1 + + ai^{*}yi \\ \text{Velynomial Order} & y0 = 1; \\ \text{ORDER} & y1 = x1; \\ y2 = x2; & 1 \text{st Order} \end{array}$	Debug Page C	oefficient MODELS #10									
Polynomial Order where i =0 to 19. y0=1; y1=x1; y2=x2; 1st Order						Reduced	d Coefficient MOE	0ELS #11			
ORDER y1=x1; y2=x2; 1st Order											
yz=xz; 1st order	(m)	reaucea model(s) Coefficients with MSE, bUP(N-m), Student 1-Value, and Model 95% CL (on Mean) in last four columns of the row = DUP = #POIIILS*#C))			
y2=x2; ist other y3=x3; ull Model # of Coef. y5=x1*x3; y5=x1*x3; y6=x2*x3; y6=x2*x3; y6=x2*x3;	0	Variable_Name	a0	al	a2	a3	a3 MS 1.144296E-3 3.3 4.107248E-1 1.2	MSE	^	#COLS	
	0	ID_Xitron_Amps	1.844843E+0	6.720419E-3	-4.201233E-	E-3 1.144 E-1 4.107		3.310291E-6			
		ID_Xitron_Watts	1.847303E+2	4.147722E+0	-8.716600E-			1.250405E-1			
		OD_Xitron_Amps	3.559474E+0	-7.120810E-2	-2.715396E-	-2 1.63	34119E-1	3.543375E-4			
y/=x1*2; 2nd Order y8=x2**2;		OD_Xitron_Watts	1.611706E+3	-2.813260E+1	E+1 0.00000E+	-0 3.597	97911E+1 1.52	1.525277E+1			
y9=x3**2;		1500_ODLiqSV_TempF	1.421453E+2	-1.445739E+0	3.447224E-1	L 6.98	88197E-1	7.675591E-3			
y10=x1*x2*x3;		1510_ODVapSV_TempF	1.771822E+2	3.526887E-1	-1.105363E-	+0 -4.5	58995E-1	5.968208E-2			
y11=x1*x2**2;		1526_Coriolis_IDSide_Temp	1.842440E+2	-1.492868E+0	0.00000E+	0 5.80)3130E-1	1.304639E-2			
y12=x1*x3**2; y13=x2*x3**2;		1527_Coriolis_ODSide_Tem	1.650854E+2	-1.565303E+0	2.355661E-1	L 6.54	42041E-1	1.135314E-2			
y14=x1**2*x2; 3rd Order		1600_IDVap_TempF	1.457619E+2	6.428017E-1	-1.017388E-	+0 -4.8	51710E-1	1.124259E-1			
y15=x1~2~x3; y16=x2**2*x3;		1601_IDLiq_TempF	1.792779E+2	-1.194046E+0	-2.327395E-	1 5.44	40338E-1	1.564798E-2			
y17=x1**3;		1602_ID2phase_Cool#1_Te	2.588990E+1	2.397976E-2	5.731748E-1	L -4.7	35205E-2	1.699212E-3			
y10=x2 5; y19=x3**3;		1603_ID2Phase_Cool#2_Te	2.606071E+1	-4.920244E-2	6.187247E-1	l -1.9	33218E-2	1.576787E-3			
		1608_CompSuct_TempF	1.521083E+2	5.437925E-1	-1.147918E-	+0 -2.9	46142E-1	5.015714E-2			
		1612_OD2PhaseHeat_F	1.201706E+2	-1.470996E+0	7.305215E-1	1 7.28	34290E-1	4.088772E-3			
		1617_CompDisch_TempF	3.394729E+2	1.132129E+0	-4.118009E	+0 4.32	22950E-1	1.790790E-1			
		1618_ID2Phase_Heat#2_F	-1.231429E+2	3.374547E+0	-8.688227E-	1 -3.0	28777E-1	1.309374E-1			
DO Backward Elimination		1622_OD2PhaseCool_F	1.159960E+2	-7.042186E-1	-7.160503E-	-2 7.24	47652E-1	3.860290E-3			
to REDUCE # Coefficients?		1630_ID2Phase_Heat#1_F	1.082097E+2	-1.537383E+0	9.733965E-1	1.41	L5992E-1	9.988484E-2			
		1701_ODVapSV_psia	5.821708E+1	5.812439E-1	1.267500E+	0 -7.3	48470E-2	2.287579E-2			
		1702_ODLiqSV_psia	4.097411E+2	-3.855689E+0	-1.934466E	+0 4.92	27888E+0	2.242525E-1			
		1707_IDVap_psia	5.102273E+1	5.821294E-1	1.354169E+	0 -5.9	37113E-2	2.036596E-2			
a Linear Fil error out		1710 ODSuctPort psia	4.131234E+1	5.232822E-1	1.417808E+	0.00	0000E+0	1.224165E-2			
code	1	1712 RotoMass Massflow I	9.840004E+0	-3.303243E-14	-3.278018E	15 1.70	55781E-14	0.00000E+0			
source		1713 RotoMass Density Ib	7.000043E+1	-1.970252E-3	5.819139E-4	-1.4	68419E-4	5.176917E-8			
		1724 IDLig psia	4.117855E+2	-4.065062E+0	-1.844146E-	+0 4.94	17583E+0	2.125322E-1			
		2219 NozTemp#1 F	5.833680E+1	8.918997E-2	2.088493E-1	-6.1	51934E-2	1.027316E-3			
		2220 NozTemp#2 F	5.006453E+1	1.339307E-1	2.430449E-1	-4.3	95111E-2	8.093327E-4			
		4		1	1.000			1			

Figure 23: Reduced coefficient models #11 tab

6. FDD Self TRAINING Polynomial APPLY FIT Ver02.vi

This program shown in Figure 24 applies the fit generated by the previous LabView program *FDD Self Training POLYNOMIAL FIT Ver07.vi*. This program is meant to be called as a sub-VI from other LabView programs. This program consists of one page that requires an input filename (A), the three independent variables in an array (B), the row index for the dependent variable's polynomial fit coefficients (C), and the order of the polynomial fit (D). The output of the program is the predicted dependent variable's (E) numeric value, the dependent variable's name string (E), and the dependent variable's fit coefficients (G) floating point values array.

For the example shown in Figure 24, the independent variables input at (B) are 29 °C (85 °F) indoor dry-bulb temperature, 21 °C (70 °F) indoor dewpoint temperature, and 44 °C (112 °F) outdoor dry-bulb temperature. The file with the appropriate coefficients is selected at (A), and the order of the polynomial fit is selected at (D). For this example, the predicted sensible capacity needs to be determined, so row 35 (C) is selected, which corresponds to row 35 in the file shown at (F). Because this is a first order polynomial fit, four coefficients are shown at (G), in LabView's SI format, along with the dependent variable's name taken from the first column of the input file.

7. FDD Algorithm Cooling OFFLINE LV2015 Ver004.vi

The previous 6 sections describe the collection of raw data from the HP under test all the way through the development of the fault-free feature polynomial fits. Now these fits will be used with subsequent raw data to perform a rule-based chart type of fault detection and diagnosis. Figure 25 shows the first tab page of the implementation of a Rule-Based Chart FDD algorithm. The details of the algorithm may be found in NIST Technical Note 1087 (Kim et. al 2008). This code uses the previously developed fault-free polynomial fits to determine the predicted feature values for the current temperature conditions, then it calculates residuals which are the predicted minus the actual values. The rule-based chart method of fault detection and diagnosis requires knowledge of the variation of system features at steady-state, so a steady-state detector is implemented within the code. More details will be provided below for each tab of the program.

7.1. TAB: Input Datafile

Figure 25 shows the *Input Datafile* tab of the FDD algorithm program. The *Datafile Name* (A) is a time series file of the measurements being collected from the heat pump (or air-conditioner) under test. One column in the input file should contain the time in seconds past midnight; each day runs from 0 s to 86399 s. If the input file is a collection of different data files taken at random times, then the user should include a column of pseudo-time values, in seconds, with the appropriate time interval for the scans.

The user must input the location of the coefficient file for the fault-free polynomial fits of the important system features at (B) and input the polynomial order at (C). The fault-free polynomial fits are used by other parts of the program. The program processes one row at a time, in sequential order, so to prevent the program from executing so quickly that inputs and outputs cannot be observed, the scan pause time at (D) slows the input process down. The user may input the time column index and two other column indices at (F); in the example file, the time index, the outdoor dry-bulb, and the indoor dry-bulb are shown in the plot; as the program

steps through the data input at the selected pause time (D), the yellow bar in the plot at (F) will move from left to right to give the user some indication of how far the program has executed through the current input raw data file. Data header labels (H) and the polynomial fits coefficient header labels (I) are always shown to aid the user in selecting the appropriate indices as required for inputs on this tab and other tabs. The button at (G), if not pressed (OFF), allows the user to read-in the raw data file and select appropriate indices before executing the FDD program; this is helpful when setting up the initial run of the program.

7.2. TAB: Assign Features, SS Detector

The next tab defines the important system features (independent and dependent variables) and sets the number of scans in the steady-state detector's moving window averaging and standard deviation calculations (Figure 26). The users must input the appropriate raw data file column indices at (A) and (C) to define the independent variables and FDD features, respectively. In this example shown in Figure 26, the independent variables are the indoor dry-bulb temperature, index 341 in the raw data file; indoor dewpoint temperature, index 242; outdoor air dry-bulb temperature, index 354; and outdoor dewpoint temperature, index 244. The important system features that must be defined are listed at (E). The indices to define these features are input at (C), and some calculated features are shown at (D).

The indices listed at (C) are those features that would be measured on the vapor compression system being monitored; only the first nine temperatures are necessary. The remaining three features, corrected refrigerant-side capacity (Cor Ref. Capacity), outdoor unit total power (OD Unit Power), and indoor unit (or air handler) total power (ID Unit Power) are not required to perform the FDD. All the indices with the same background color are related; for instance, the yellow background indices are all selected from the raw *Data HEADER Labels* array in the far upper left. The blue backgrounds are all values related to the moving window features.

The moving window features are all listed at (E) with their associated maximum standard deviations for steady-state at (F). The steady-state, maximum standard deviations are determined for each system installation by measurements taken while the system is operating over a "long" time interval. Continuous operation of the unit over a long time interval will give a good sample of the calculated features which will allow for determination of steady-state standard deviations. The maximum standard deviations shown in Figure 26 at (F) were determined from laboratory system measurements using carefully instrumented airconditioners and heat pumps. Field measurements, with lower cost data acquisition equipment, may have larger values of the maximum standard deviations. All seven of the moving window feature value's standard deviations must be less than the maximum before the FDD rule-based chart method can be applied. A more thorough description of the steady-state detector and an analysis of selecting appropriate moving window size may be found in NIST Technical Note 1087 (Kim et. al 2008). The features defined at (C) and (D) are described in Table 1.

Name as shown at (C) and (D) in Figure 26	Description	NOTE				
Cooling Tsat ID	Indoor evaporator two-phase refrigerant temperature	Measured on refrigerant tube surface of the expansion valve distributor or an evaporator tube return bend				
Cooling Tsat OD	Outdoor coil/condenser refrigerant saturation temperature	Measured on a return bend of the condenser coil; verify that two-phase is occurring using a pyrometer or other temperature measuring device				
TID, vapor	Indoor coil/evaporator exit refrigerant vapor temperature	Measured on the vapor line/suction line near the indoor air handler by a surface mounted sensor				
TOD, vapor	Outdoor coil/condenser cooling mode hot refrigerant gas temperature entering the coil	Measured on the surface of the refrigerant vapor line near the inlet of the condenser (could be the same as the compressor discharge temperature for an AC)				
TIDair, exit	Indoor air handler supply air dry-bulb temperature (cold air temperature exiting the AC)	Preferably measured in the airstream at the center of the ductwork about 2 duct diameters after the air handler (a single sensor)				
TODair, exit	Outdoor coil/condenser air exit dry-bulb temperature	Measured at the exit of the condenser, after the fan				
Tdisch	Compressor exit/ discharge refrigerant temperature	Measured on the surface of the compressor refrigerant discharge line				
TOD, liq	Outdoor coil/condenser refrigerant liquid temperature	Measured on the surface of the refrigerant liquid line near the service valve				
TID, liq	Indoor coil/evaporator refrigerant liquid temperature	Measured on the surface of the refrigerant liquid line near the indoor coil/evaporator before the expansion valve				
Cor Ref. Capacity	Corrected refrigerant- side cooling capacity	The user may have to modify the program to calculate the corrected refrigerant-side capacity from the measurements being made and a compressor map mass flowrate. In this example, the value was found from actual				

Table 1: Parameters used to determine moving window features and values

		mass flow measurements with temperatures and pressures at the inlet and exit to the evaporator. The corrected refrigerant-side capacity has the indoor blower estimated, or measured, power demand subtracted to reflect the actual air-side capacity.
OD Unit Power	Outdoor/Condensing unit total power	Calculated from a current transducer installed on the input power wire (AC Amps) and previously measured input voltage (VAC) and an assumed/measured power factor. The voltage and power factor are assumed to be stable $P = V*I*PF$
ID Unit Power	Indoor/Evaporator air handler total power	Calculated the same as the outdoor total power using a current transducer installed on the input power wire
dTODair	Outdoor coil/condenser air dry-bulb temperature change	Calculated from the outdoor air dry-bulb (TODair,Dry at (A) in Figure 26) and condenser exit air dry-bulb temperature difference
dTIDair	Indoor coil/evaporator air dry-bulb temperature change	Calculated from the indoor return air dry-bulb (TIDair,Dry at (A) in Figure 26)and supply air dry-bulb temperature difference
Cool dTsh	Indoor coil/evaporator refrigerant exit vapor superheat	Calculated as the difference in TID,vapor and Cooling Tsat ID
Cool dTsc	Outdoor/condenser refrigerant liquid line subcooling	Calculated as the difference in TOD, liq and Cooling Tsat OD
dTdisch,sh	Compressor discharge refrigerant superheat	Calculated as the difference in TOD, vapor and Cooling Tsat OD
Meas. COP	Measured Coefficient of Performance (COP)	Calculated from the Cor Ref. Capacity and sum of OD Unit Power and ID Unit Power

The moving window feature values, maximum standard deviations, and actual standard deviations are shown at (E) and (F) of Fig. 26. Compressor discharge superheat and measured COP are also placed in the moving window calculation at (H). The refrigerant-side capacity is shown at (G) with a user adjustable additive correction input location. The correction is added to the refrigerant-side capacity being calculated and then displayed at (C).

7.3. TAB: Cooling FDD Rule-based Chart

The tab shown at Figure 27 is where the rule-based chart is defined, and the appropriate coefficient file indices are selected for calculating the fault-free values of the important system

features. The temperature and dewpoint value indices were already input on the previous tab, but are shown again here at (A). The fault-free feature values are listed at (C) and the appropriate indices are input at (B) from the header labels array at (F). The calculated COP serves as a warning value to indicate something is causing the system to deviate from its fault-free performance; a maximum percent degradation is input at the bottom of the array at (C). In this example, when the measured/calculated COP drops by 10 % or more, the *Steady and Degraded?* indicator light near (H) will turn red and the background color of the most likely fault, in the list numbered 0 to 6 at (H), will also turn red. The background color of the bar chart at (I) will also turn red as the chart shows the probabilities of the 7 possible fault conditions listed at (H).

The Cooling Rule-Based Chart Array is defined at (D) and consists of values of 0, +1, or -1. Each column of the chart is associated with a particular system feature; Te – evaporator refrigerant saturation temperature, Tsh - evaporator exit refrigerant vapor superheat, TD compressor refrigerant discharge temperature, TC - condenser refrigerant saturation temperature, Tsc – condenser liquid refrigerant subcooling, dTCA – condenser inlet and exit air temperature change, and dTEA – evaporator air temperature change. The fault-free values of all these system features are calculated from the polynomial fits selected at (B) from the file at (F). The rows of (D), from top to bottom, are CMF (compressor valve leakage or hot gas bypassing the compressor and going directly to the suction side), ODFoul (condenser air flow area blockage), IDFoul (evaporator air flow change), LL (refrigerant liquid line restriction causing excessive pressure drop), UC (refrigerant under charge or loss of refrigerant), OC (refrigerant over charge or excessive mass of refrigerant in the system), and NF (no-fault or fault-free operation). The intersection of a row and column indicates the tendency of the feature residual (measured value minus predicted value) to be neutral or fault-free (0), higher or larger than expected (positive one, +1), or lower than expected (negative one, -1). The neutral case (or no-fault case) is defined by the variation of the moving window average value from its predicted value; the acceptable variation of a feature's value while still remaining fault-free is described by the Feature Thresholds (E). Defining the feature thresholds at a particular statistical confidence level is described in NIST Technical Note 1087 (Kim et. al 2008).

The *Individual Probabilities (%) as Described by the RULE-BASED CHART ARRAY* indicates the individual probability that a particular feature has a neutral, positive, or negative residual (a 0, +1, or -1). For example, refer to Figure 27 at (D), the *Cooling Rule-based Chart Array*, and observe that row 0 column 0 (*Te* with a CMF fault at index (0,0)) equals positive one (+1). This means that the evaporator saturation temperature will be higher than normal for a compressor valve leakage fault; the individual probabilities array at row 0 column 0, index(0,0), shows 15.88 % as this probability (0 % to 100 %). For this example file, the value of 15.88 % is the probability of the evaporator saturation being higher than the neutral case (as defined by the Feature Thresholds at (E) for a 95 % confidence level). Referencing the rule-based chart at (D) at row 2 column 0, index(2,0), *Te* has a rule of negative one (-1) for an IDFoul fault at an individual probability of 15.85 % in the probability chart at (G) also at index(2,0). The *Te* is neutral, value 0, in the rule-based chart at (D), index(4,0), for an under charged refrigerant fault (UC fault). The corresponding individual probability of the neutral case is found at (G), index(4,0), as 68.27 %. The sum of the individual probabilities for being neutral, positive, or negative is always one (100 %). For example, the sum of the individual

probabilities for Te (being neutral -index(4,0), positive -index(0,0), or negative -index(2,0)) is 100 % (60.87 %+15.88 %+15.85 %=100 %). If you multiply all the probabilities across a row of the array at (G), that equals the combined probability that a given fault is occurring; in this example the multiplication of all the elements in the last row of the array at (G) equals the value shown in the last row of the array at (H) (*Cooling Fault Probability*) which corresponds to the no-fault case (fault-free case). Fault-free or no-fault is also the largest bar on the chart at (I).

7.4. TAB: Rule-based Chart PLOTS

Figure 28 shows some bar charts that are meant to help the user to visualize how the overall probability of the different faults develop as the raw data file is processed. The individual probabilities chart at (A) and the most likely fault indicator at (B) is copied here from the previous tab just for the user's reference. The bar charts at (C) and (D) are meant to graphically show the individual probabilities for all the important system features listed across the top of the array at (A). The bar chart at (C) defaults to showing the individual probabilities for the fault-free row of the array at (A); row 6 is the fault-free (neutral probability) for all of the important system features. The bar chart at (D) defaults to show the individual probabilities for the fault on row 4; the liquid line restriction fault. Using these two bar charts at (C) and (D), the user may compare different rows of the array at (A).

7.5. TAB: NOTES

Figure 29 contains a text indicator box that may be used by the user to save information that is a useful reference from run-to-run of the main program. The example shown in Figure 29 explains some of the considerations that were implemented for calculating the overall values of the neutral or no-fault (fault-free) thresholds. If the user wishes to edit and save changes to the text box, they would select the box in the edit mode (Ctrl-M toggles the edit mode) and then save the entire VI.

Multivariate Polyno	INPUT Filena	I TOP 3 INDEPER	ndent Variab	les				
	Coefficients with MSE File Path							
y = a0*y0 + a1*y1 ++a where i =0 to 19. y0=1; y1=x1; y2=x2; 1st Order y3=x3; 	i*yi A B xi xi	C:\Vance\FDD Split System & Example_RAW_Datafile.bt Independent Variables (3 values = Tid, Tidp, Tod) 1 85 Which 70 (Row # 112 35	Project\NIST RULE BASED C Model? = 0 to N) Polynomia (1st, 2nd, o 1 1 1 1 1 1 1 1 1 1 1 1 1	HART LABVIEW FDD CODE June 20 ORDER r 3rd)	118\COEF-1st_FULL-BINNED-SS-	cted VALUE		
y5=x1*x3; v6=x2*x3;		0		2	14887.3	Sensible Capacity (Btu	u/h)	
y7=x1**2; 2nd Order		Coefficient HEADER LABELS - S	itring		(<u>1 - 1</u>)	1 2 1	· ·	
y8=x2**2;		Variable_Name	a0	al	a2	a3	MSE	
y9=x3**2;		Coefficients Array - String						
		Sensible Canacity (Btu/h)	-9 161709E+4	9 572742F+2	1 490481E+2	1 312740E+2	1.245765E+3	
y10=x1*x2*x3;	50	Latent Capacity (Btu/h)	-8.957121E+4	5.932173E+1	1.573383E+3	-7.530459E+1	9.105311E+3	
y11=x1*x2**2;		TOTAL Capacity (Btu/h)	-1.811883E+5	1.016596E+3	1.722431E+3	5.596939E+1	9.260908E+3	
y12=x1^x3^2;		SHR	1.735299E-2	1.379948E-2	-1.530144E-2	3.023308E-3	1.902002E-6	
y13=X2^X3^^Z;		ODVapSV_Suph_F	1.594634E+2	1.576712E-1	-1.528396E+0	-4.310914E-1	6.530676E-2	
y14=x1^2x2; 3rd Order		ODVapSV_Tsat_F	1.771889E+1	1.950147E-1	4.230353E-1	-2.480818E-2	2.576794E-3	
y15=x1***2*x3;		ODLiqSV_Tsat_F	1.221772E+2	-6.163474E-1	-3.592881E-1	7.718887E-1	5.562600E-3	
y10=X2^2X3;		ODLiqSV_Tsub_F	-1.996806E+1	8.293897E-1	-7.040095E-1	7.306896E-2	4.396901E-3	
y1/=x15,		4 C						
y18=x2^^; y19=x3**3;	G ONLY Coeffi [-91.62k] a0	Dependent Variable NAME G Sensible Capacity (Btu/h) ONLY Coefficients Array - Numeric 91.62k 957.3 149.0 131.3 0.000						

Figure 24: Input filename #1 tab

1



Figure 25: Input datafile tab



Figure 26: Assign features and steady-state detector tab



Figure 27: Cooling FDD rule-based chart tab



Figure 28: Rule-based chart plots tab



8. SUMMARY

The codes presented in this report can be found here:

https://github.com/FDeeDee/NIST-FDD-for-Residential-Air-Conditioners-and-Heat-Pumps

These codes allow a user to 1) process a continuous stream of incoming data and save those points which are at steady-state, 2) keep a portion of the steady-state data that covers the widest possible landscape described by the independent variables, 3) produce polynomial correlations of fault-free system parameters, 4) use the polynomial fits to determine system parameters given the current independent variables, and 5) use the fault-free fits in a rule-based chart method to detect and diagnose faults in the cooling mode.

Acknowledgments

The work of many NIST researchers and foreign guest researchers has gone into producing and refining the FDD codes illustrated in this guide. The author thanks Drs. Minsung Kim, Seok Ho Yoon, Christian J.L. Hermes, Jin Min Cho, Jaehyeok Heo, and Piotr A. Domanski. Technician support to gather FDD data was provided by Glenn Glaeser, David Wilmering, Arthur Ellison, and John Wamsley.

References

- Blemel, Kenneth (Management Sciences, Inc.). 2014. "Cognitive Residential Heat Pump Fault Detection and Diagnostic Datalogger." 2014. https://www.nist.gov/tpo/smallbusiness-innovation-research-program/sbir-past-awards/fy2014.
- Braun, James. 2001. "Description of Fdd Modeling Approach for Normal Performance Expectation" 1077 (December): 1–60.
- Braun, James E. 2003. "Automated Fault Detection and Diagnostics for Vapor Compression Cooling Equipment." *Journal of Solar Energy Engineering, Transactions of the ASME* 125 (3): 266–74. https://doi.org/10.1115/1.1591001.
- Du, Zhimin, Bo Fan, Xinqiao Jin, and Jinlei Chi. 2014. "Fault Detection and Diagnosis for Buildings and HVAC Systems Using Combined Neural Networks and Subtractive Clustering Analysis." *Building and Environment* 73: 1–11. https://doi.org/10.1016/j.buildenv.2013.11.021.
- Esen, Hikmet, Mustafa Inalli, Abdulkadir Sengur, and Mehmet Esen. 2008. "Modelling a Ground-Coupled Heat Pump System Using Adaptive Neuro-Fuzzy Inference Systems." *International Journal of Refrigeration* 31 (1): 65–74. https://doi.org/10.1016/j.ijrefrig.2007.06.007.
- Han, Hua, Zhikun Cao, Bo Gu, and Neng Ren. 2010. "Pca-Svm-Based Automated Fault Detection and Diagnosis (Afdd) for Vapor-Compression Refrigeration Systems." *HVAC* and R Research 16 (3): 295–313. https://doi.org/10.1080/10789669.2010.10390906.
- Han, Hua, Ling Xu, Xiaoyu Cui, and Yuqiang Fan. 2020. "Novel Chiller Fault Diagnosis Using Deep Neural Network (DNN) with Simulated Annealing (SA)." *International Journal of Refrigeration*. https://doi.org/10.1016/j.ijrefrig.2020.10.023.

- Isermann, Rolf. 2005. "Model-Based Fault-Detection and Diagnosis Status and Applications." *Annual Reviews in Control* 29 (1): 71–85. https://doi.org/10.1016/j.arcontrol.2004.12.002.
- Katipamula, Srinivas, and Michael R. Brambley. 2005. "Review Article: Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems—a Review, Part II." *HVAC and R Research* 11 (2): 169–87. https://doi.org/10.1080/10789669.2005.10391133.
- Kim, Minsung, W. Vance Payne, Piotr A. Domanski, Seok Ho Yoon, and Christian J.L. Hermes. 2009. "Performance of a Residential Heat Pump Operating in the Cooling Mode with Single Faults Imposed." *Applied Thermal Engineering* 29 (4): 770–78. https://doi.org/10.1016/j.applthermaleng.2008.04.009.
- Kim, Minsung, W Vance Payne, Piotr A Domanski, and Christian J.L. Hermes. 2006. "Operating in the Cooling Mode With Single Faults Imposed." Gaithersburg, MD USA.
- Kim, Minsung, Seok Ho Yoon, Piotr A. Domanski, and W. Vance Payne. 2008. "Design of a Steady-State Detector for Fault Detection and Diagnosis of a Residential Air Conditioner." *International Journal of Refrigeration* 31 (5): 790–99. https://doi.org/10.1016/j.ijrefrig.2007.11.008.
- National Instruments. 2015. "LabView Professional Development System 2015." 11500 N Mopac Expwy, Austin, TX 78759-3504: National Instruments Corp. https://www.ni.com/en-us/about-ni.html.
- Payne, W. Vance, Jaehyeok Heo, and Piotr A. Domanski. 2018. "A Data-Clustering Technique for Fault Detection and Diagnostics in Field-Assembled Air Conditioners." *International Journal of Air-Conditioning and Refrigeration* 26 (2): 1–17. https://doi.org/10.1142/S2010132518500153.
- Rogers, A. P., F. Guo, and B. P. Rasmussen. 2019. "A Review of Fault Detection and Diagnosis Methods for Residential Air Conditioning Systems." *Building and Environment* 161 (June): 106236. https://doi.org/10.1016/j.buildenv.2019.106236.
- Rossi, Todd M, and James E Braun. 1997. "A Statistical, Rule-Based Fault Detection and Diagnostic Method for Vapor Compression Air Conditioners." *HVAC&R Research* 3 (1): 19–37. https://doi.org/10.1080/10789669.1997.10391359.
- Wikipedia. 2020a. "Interquartile Range." 2020. https://en.wikipedia.org/wiki/Interquartile_range.
 - ----. 2020b. "Nyquist-Shannon Sampling Theorem." 2020. https://en.wikipedia.org/wiki/Nyquist-Shannon_sampling_theorem.
- Yoon, Seok Ho, W. Vance Payne, and Piotr A. Domanski. 2011. "Residential Heat Pump Heating Performance with Single Faults Imposed." *Applied Thermal Engineering* 31 (5): 765–71. https://doi.org/10.1016/j.applthermaleng.2010.10.023.

Yuill, David P., and James E. Braun. 2013. "Evaluating the Performance of Fault Detection

and Diagnostics Protocols Applied to Air-Cooled Unitary Air-Conditioning Equipment." *HVAC and R Research* 19 (7): 882–91. https://doi.org/10.1080/10789669.2013.808135.

Zogg, David, Esfandiar Shafai, and Hans P. Geering. 2001. "A Fault Diagnosis System for Heat Pumps." In 2001 IEEE International Conf. on Control Applications, 70–76. Mexico City.