NIST Technical Note 2130

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

National Institute of
Standards and Technology
U.S. Department of Commerce

NIST Technical Note 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

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

List of Tables

List of Figures

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.

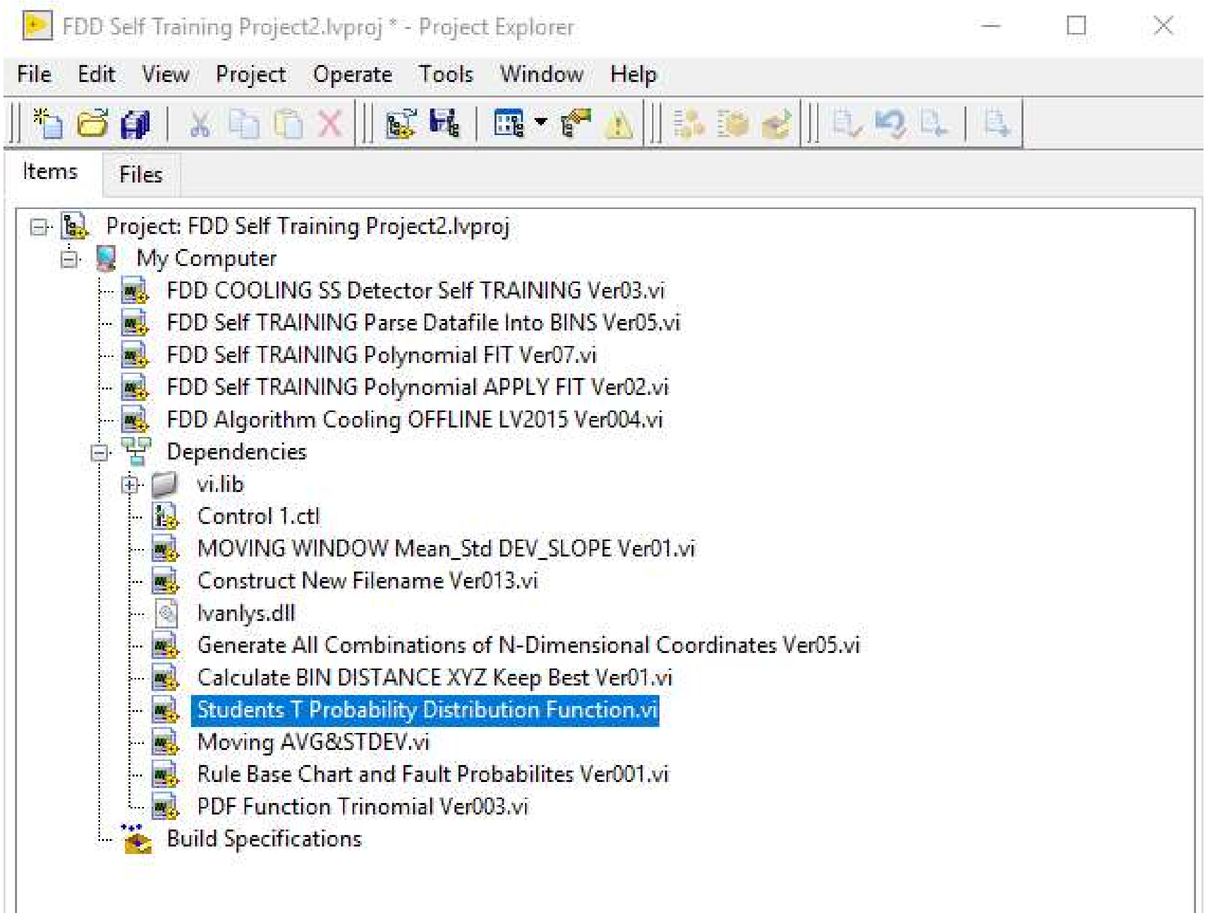


Figure 1: Project file listing as shown in LabView 2015

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 dry-bulb 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).

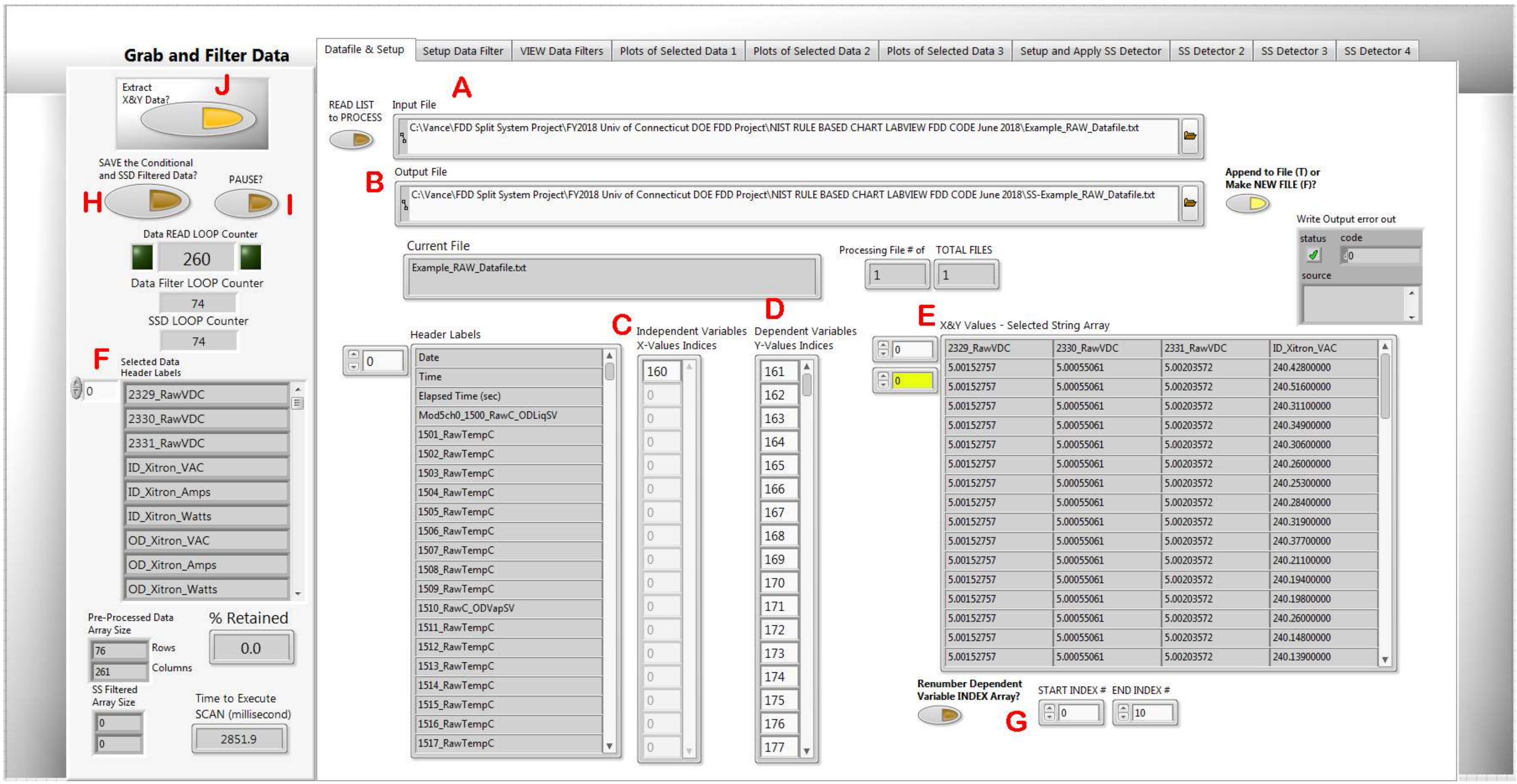


Figure 2: FDD SSD datafile and setup page

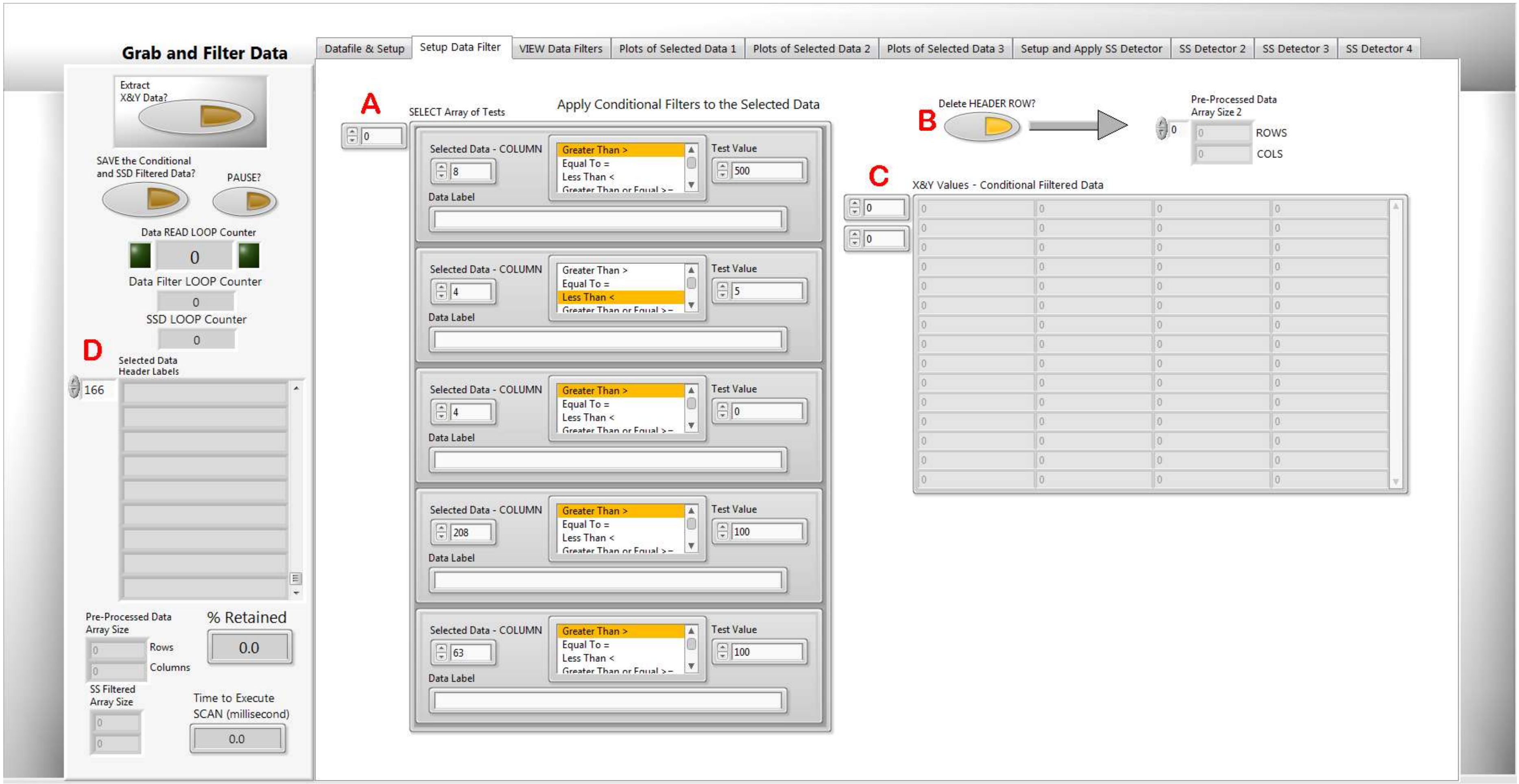


Figure 3: FDD SSD setup data filter tab

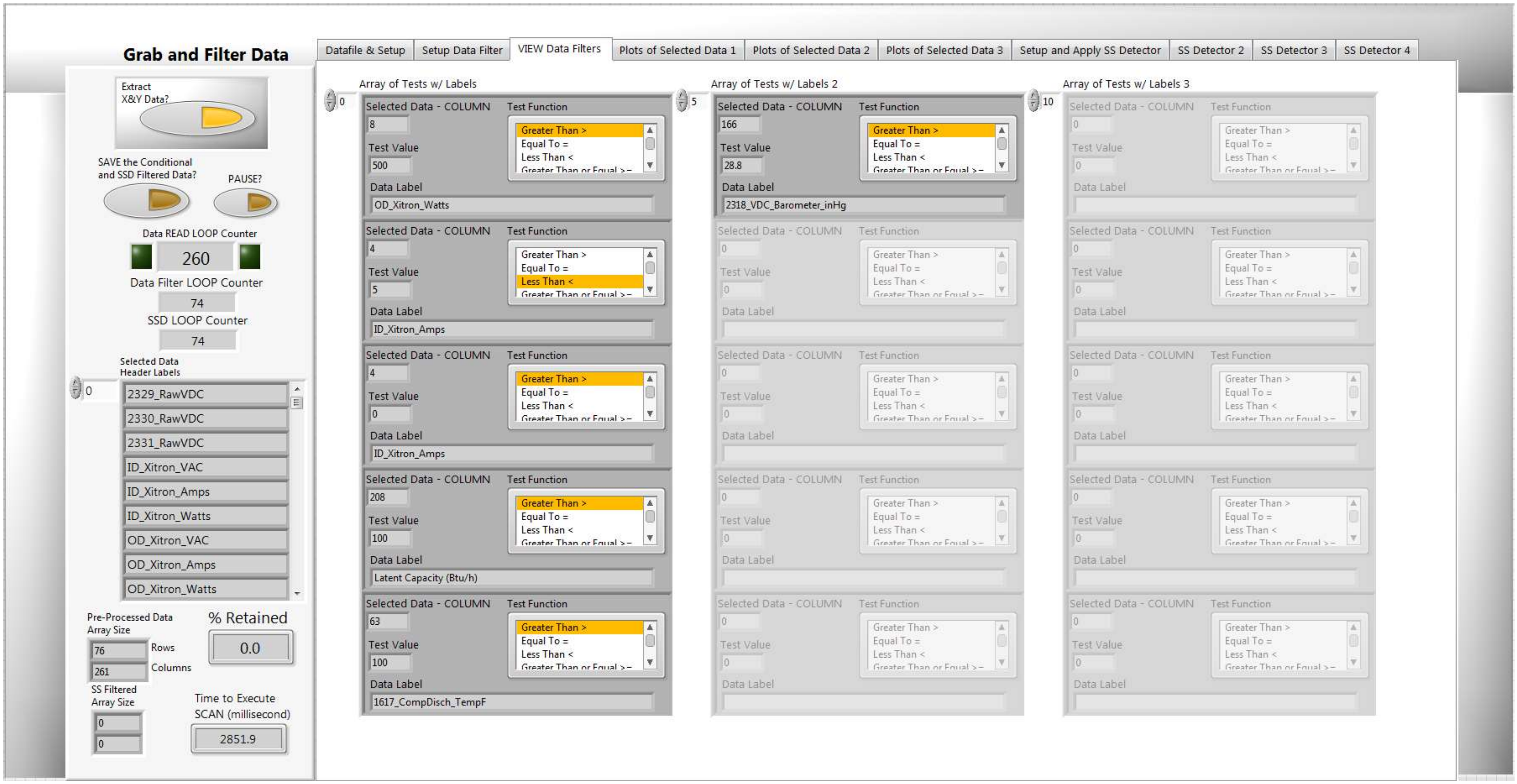


Figure 4: VIEW Data Filters tab

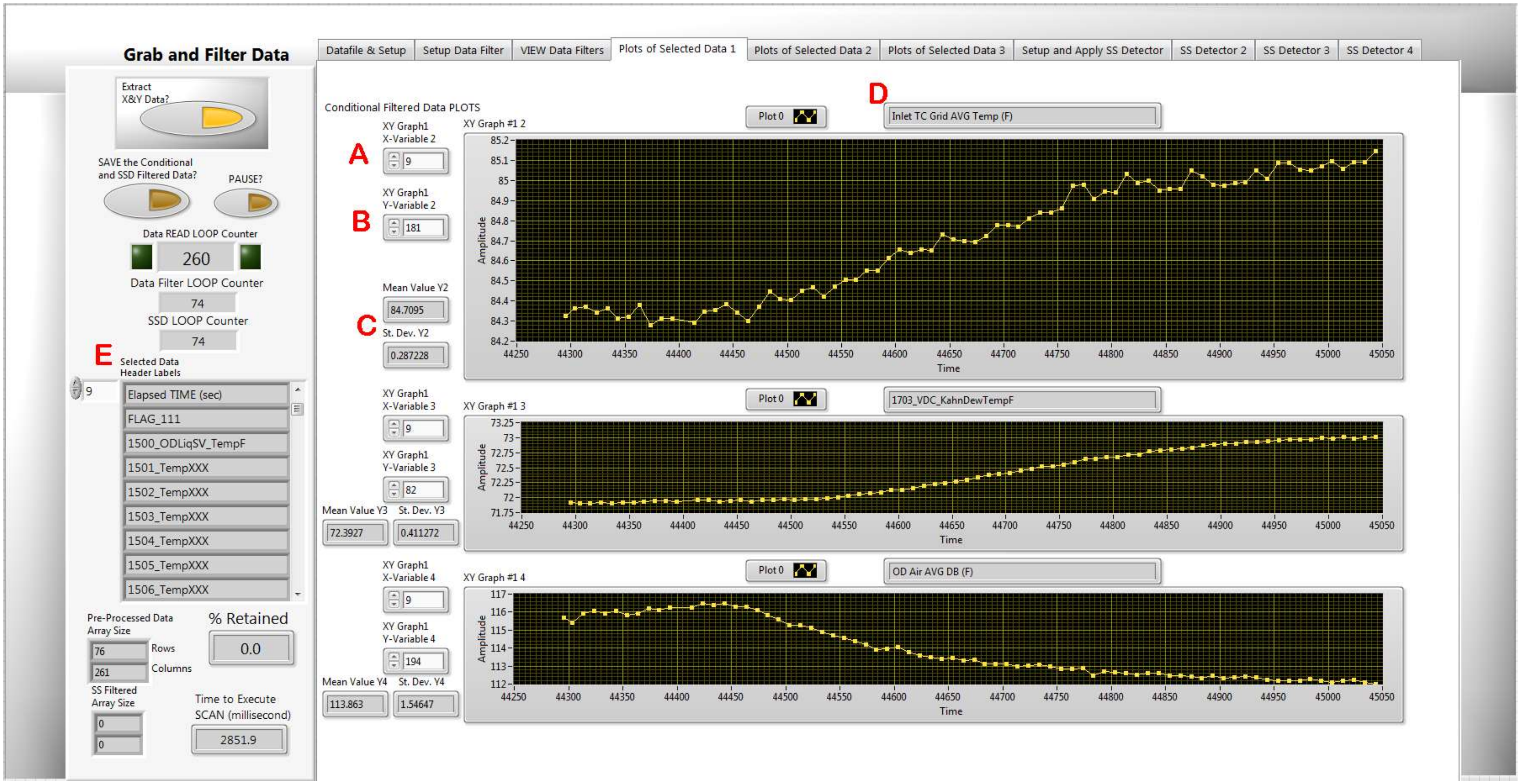


Figure 5: Plots of Selected Data 1 tab

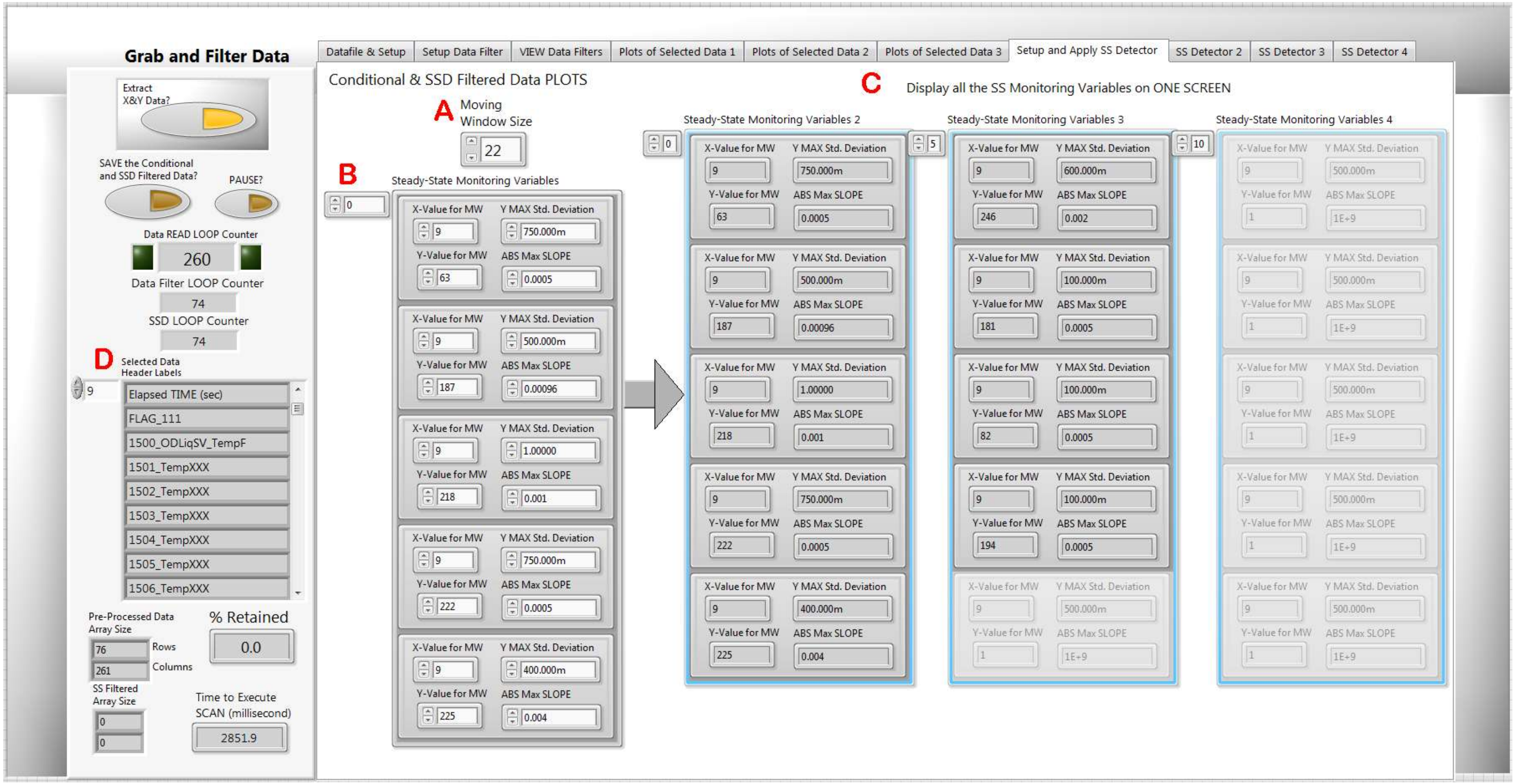


Figure 6: Setup and Apply SS Detector tab

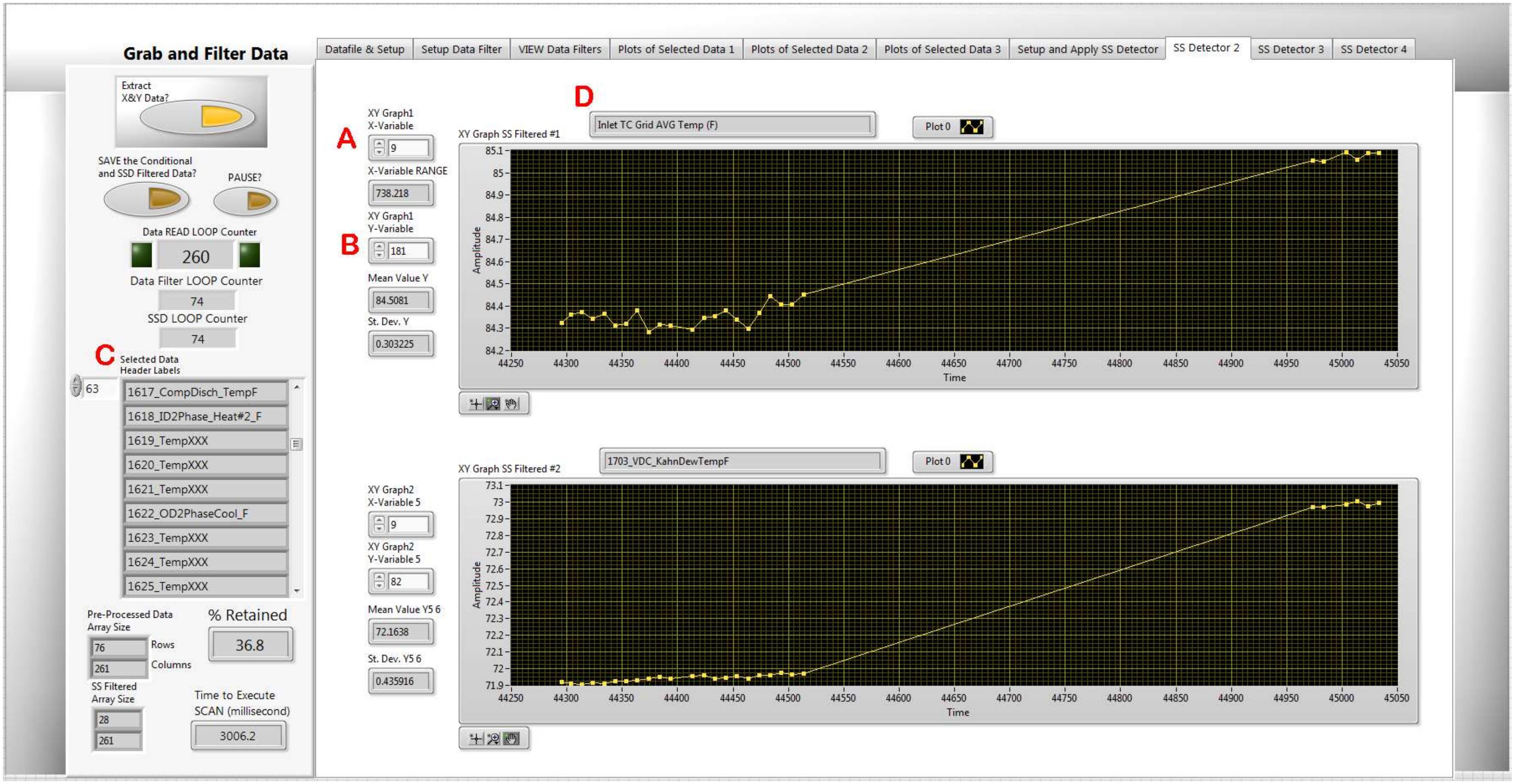


Figure 7: SS Detector 2, 3, and 4 tabs

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.

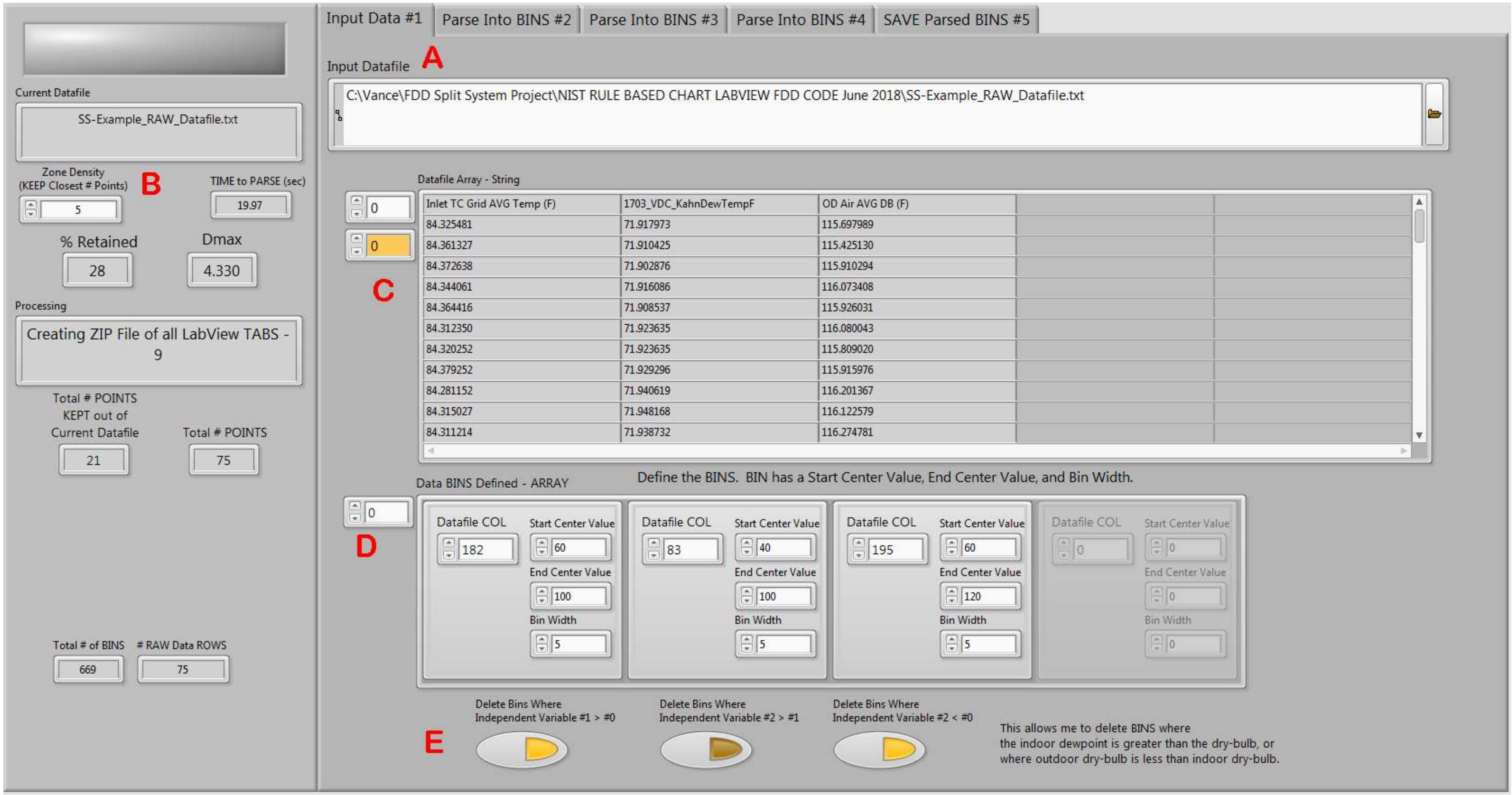


Figure 8: Input datafile tab

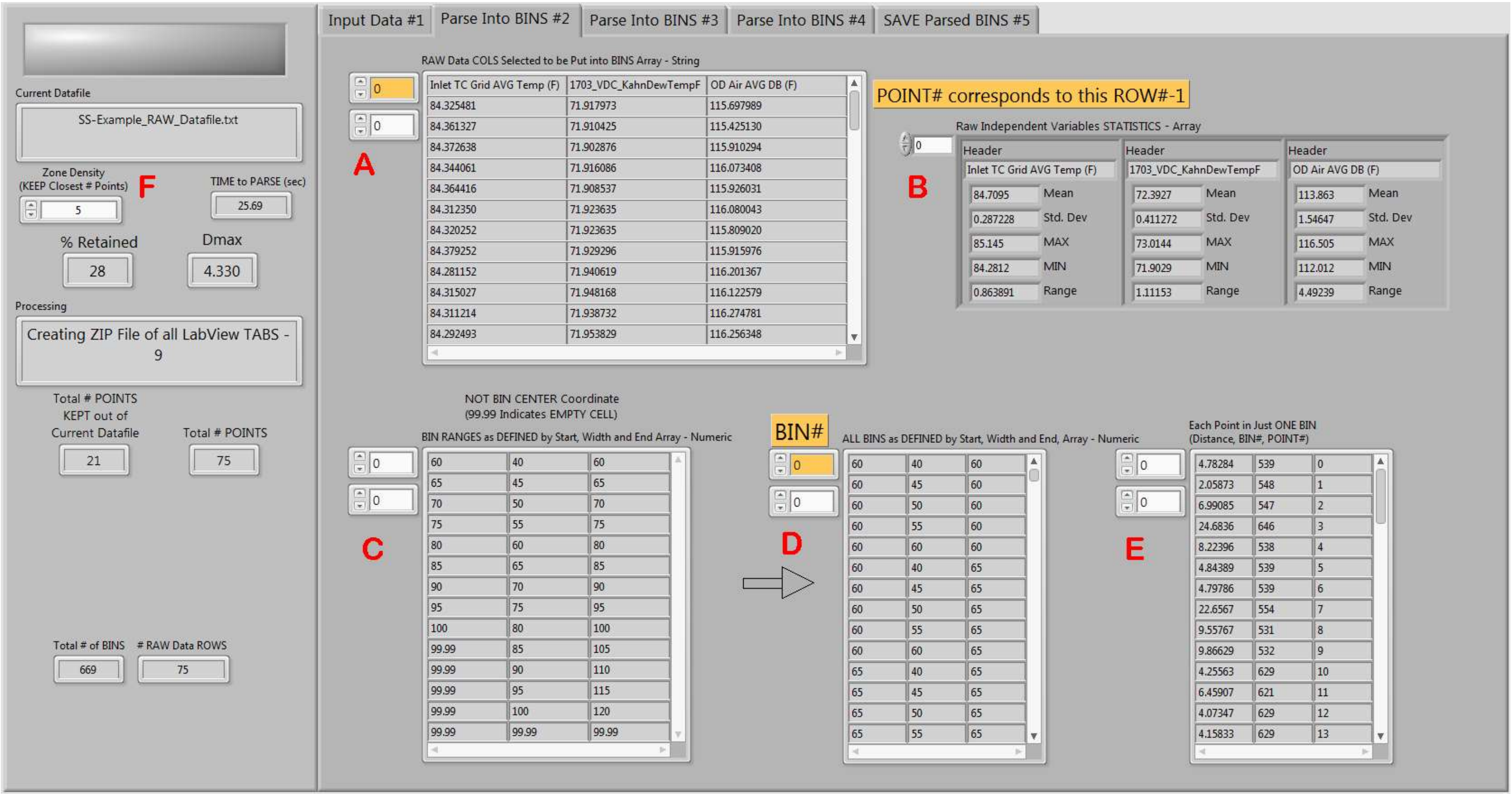
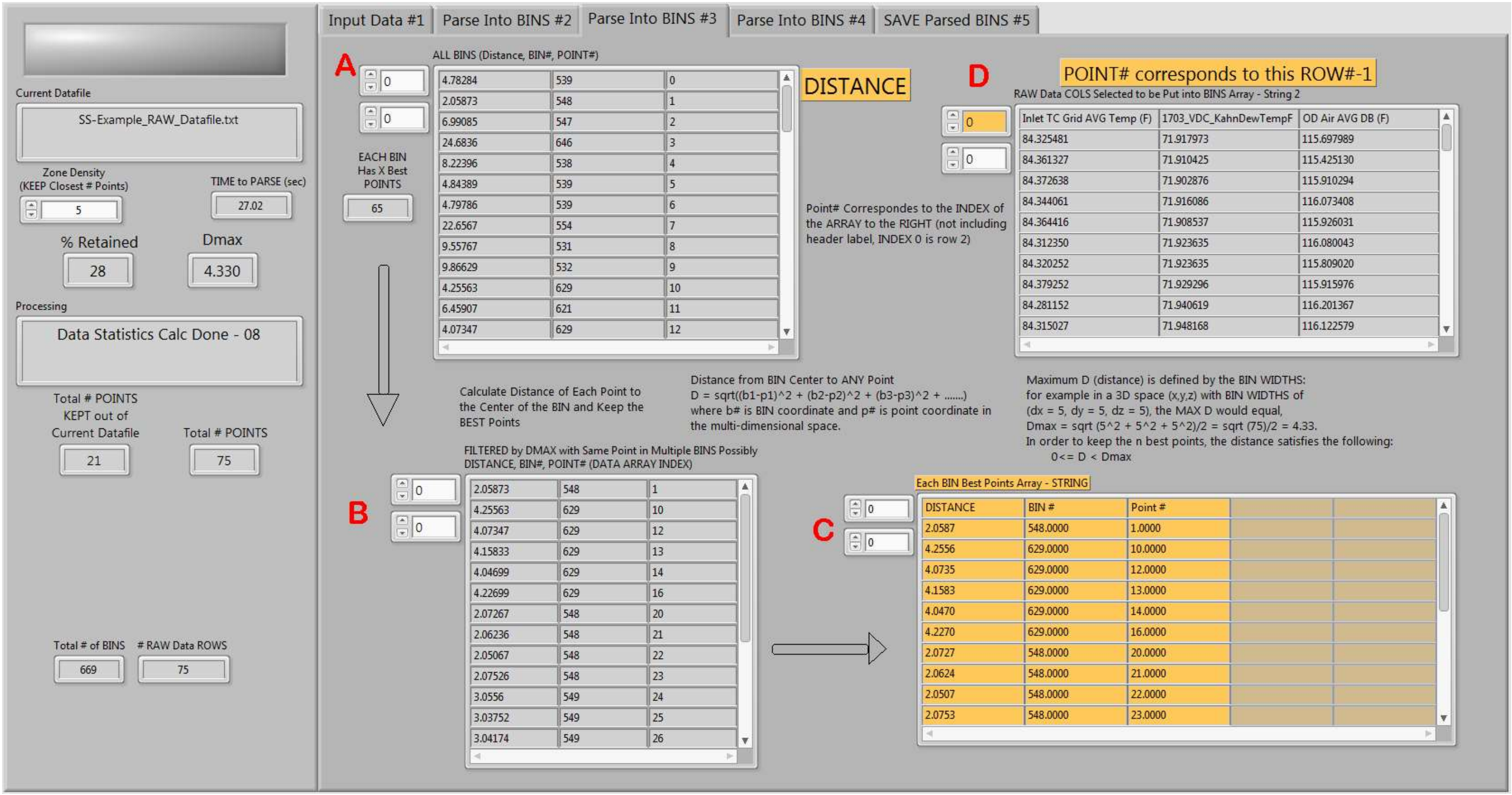


Figure 9: Parse into bins #2 tab



4.78284	539	0
2.05873	548	1
6.99085	547	2
24.6836	646	3
8.22396	538	4
4.84389	539	5
4.79786	539	6
22.6567	554	7
9.55767	531	8
9.86629	532	9
4.25563	629	10
6.45907	621	11
4.07347	629	12

Inlet TC Grid AVG Temp (F)	1703_VDC_KahnDewTempF	OD Air AVG DB (F)
84.325481	71.917973	115.697989
84.361327	71.910425	115.425130
84.372638	71.902876	115.910294
84.344061	71.916086	116.073408
84.364416	71.908537	115.926031
84.312350	71.923635	116.080043
84.320252	71.923635	115.809020
84.379252	71.929296	115.915976
84.281152	71.940619	116.201367
84.315027	71.948168	116.122579

2.05873	548	1
4.25563	629	10
4.07347	629	12
4.15833	629	13
4.04699	629	14
4.22699	629	16
2.07267	548	20
2.06236	548	21
2.05067	548	22
2.07526	548	23
3.0556	549	24
3.03752	549	25
3.04174	549	26

DISTANCE	BIN #	Point #
2.0587	548.0000	1.0000
4.2556	629.0000	10.0000
4.0735	629.0000	12.0000
4.1583	629.0000	13.0000
4.0470	629.0000	14.0000
4.2270	629.0000	16.0000
2.0727	548.0000	20.0000
2.0624	548.0000	21.0000
2.0507	548.0000	22.0000
2.0753	548.0000	23.0000

Figure 10: Parse into bins #3 tab

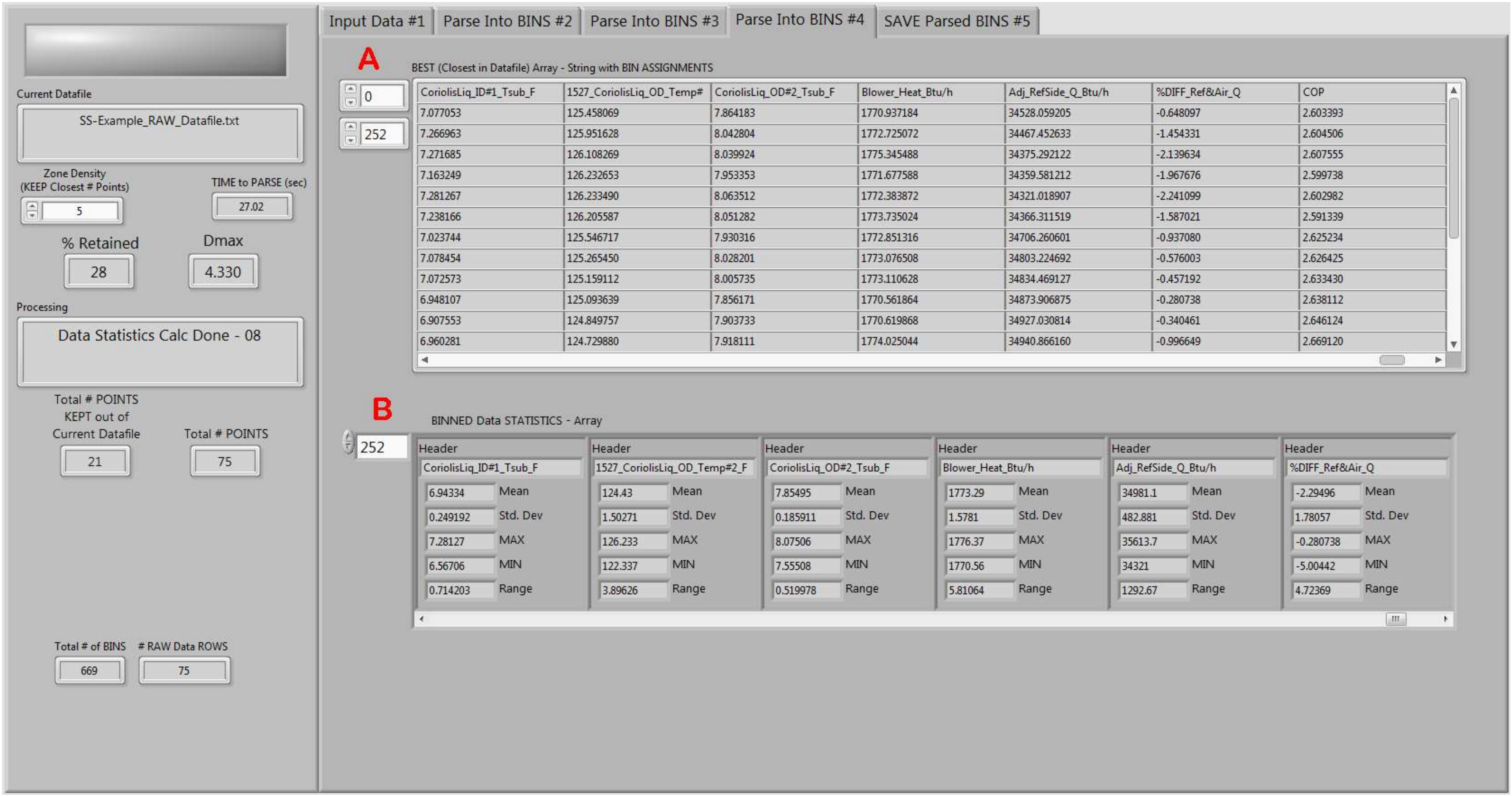


Figure 11: Parse into bins #4 tab

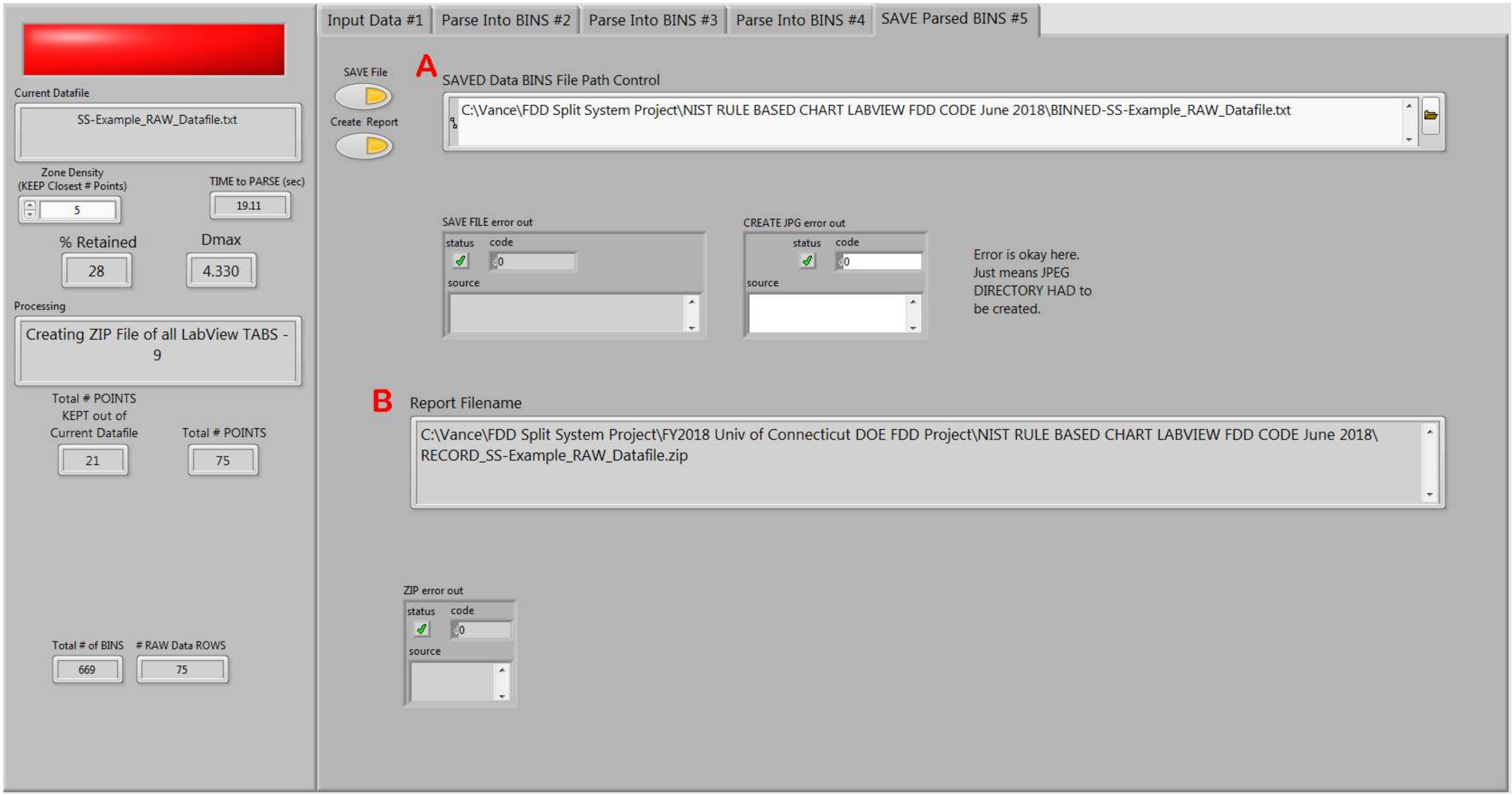


Figure 12: Save parsed bins #5 tab

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 25^{th} or 75^{th} 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). The plot 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 1^{st} 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.

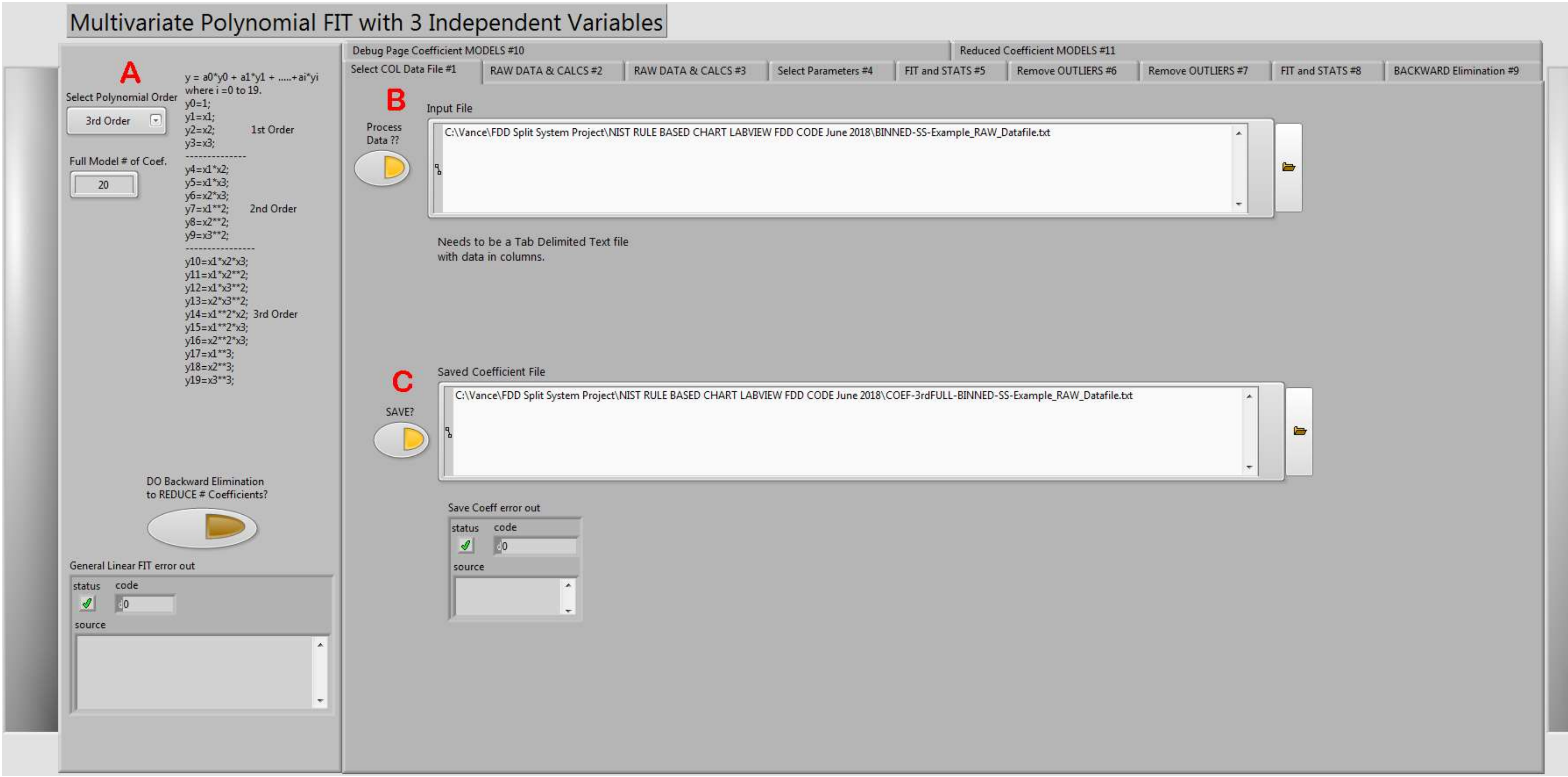


Figure 13: Select COL data file #1 tab

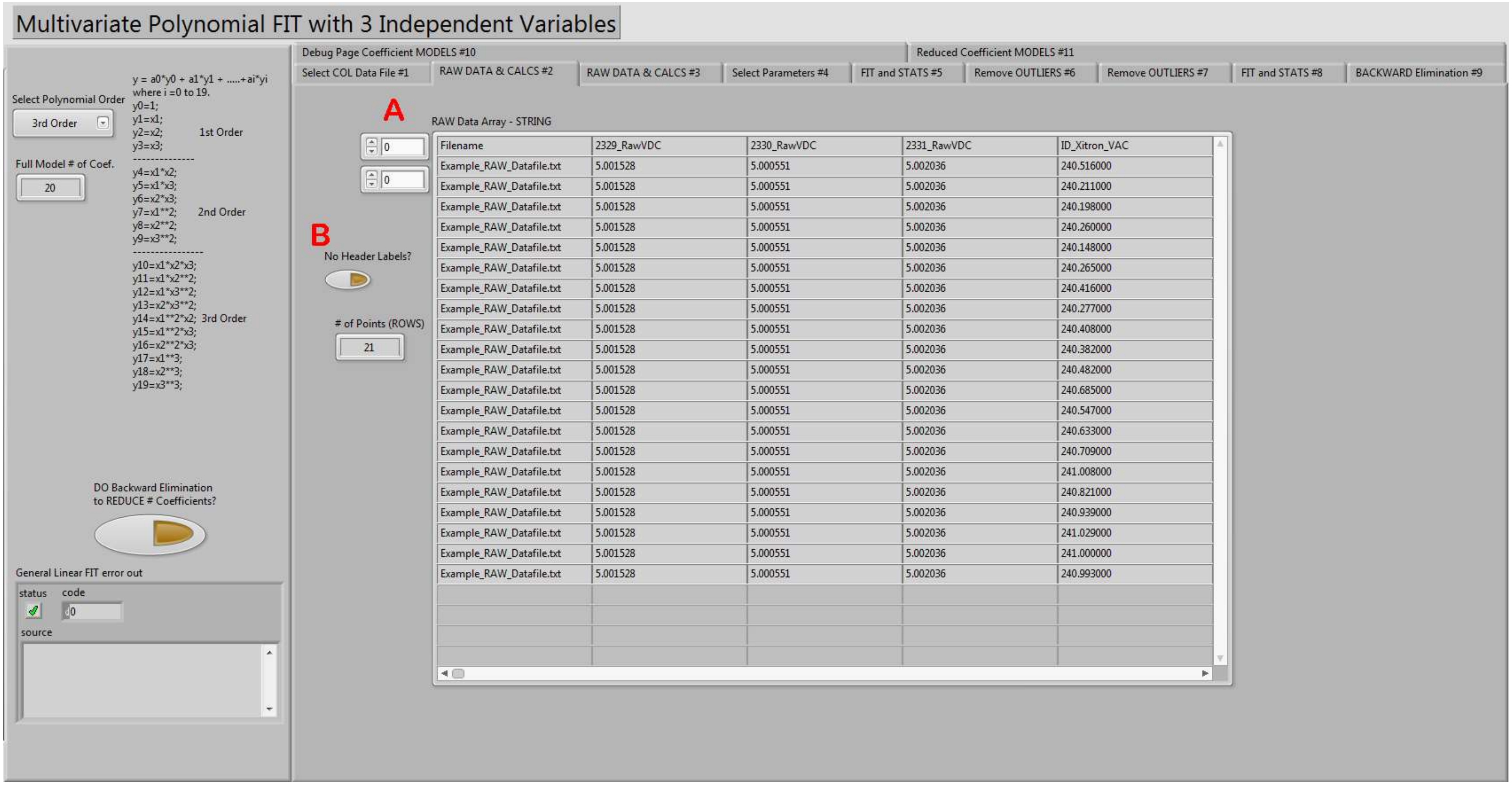


Figure 14: Raw data and calculations #2 tab

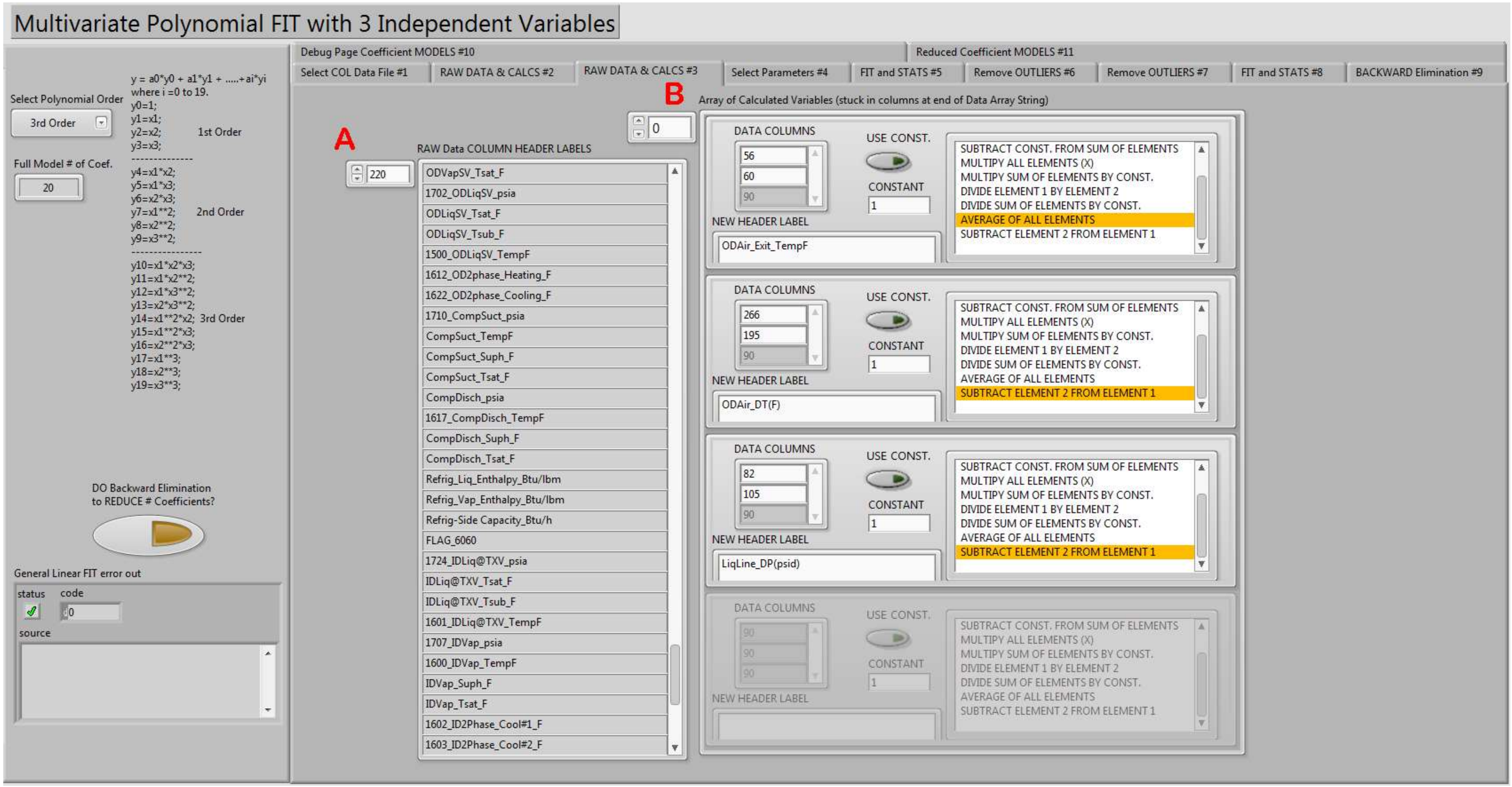


Figure 15: Raw data and calculations #3 tab

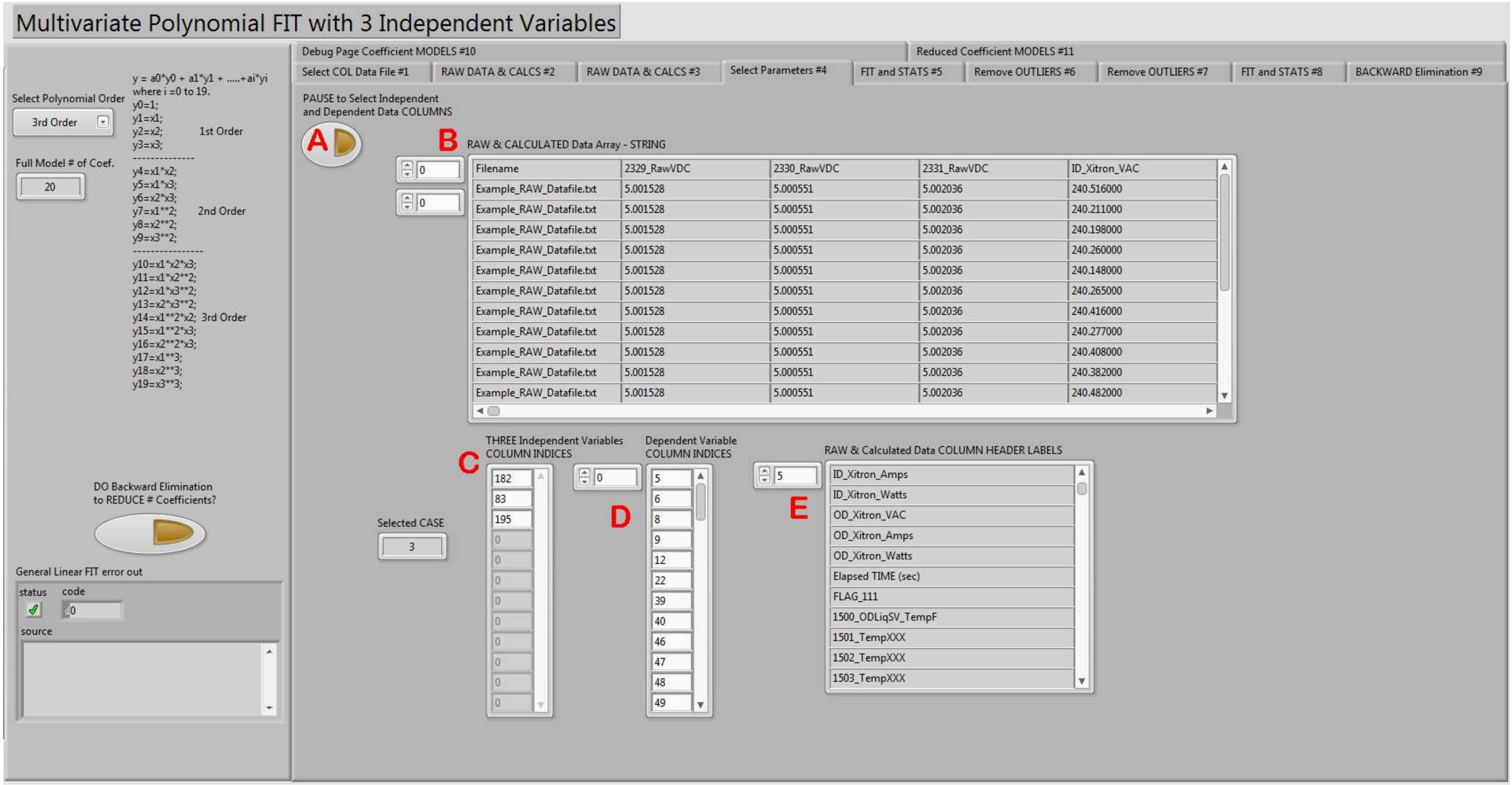


Figure 16: Select parameters #4 tab

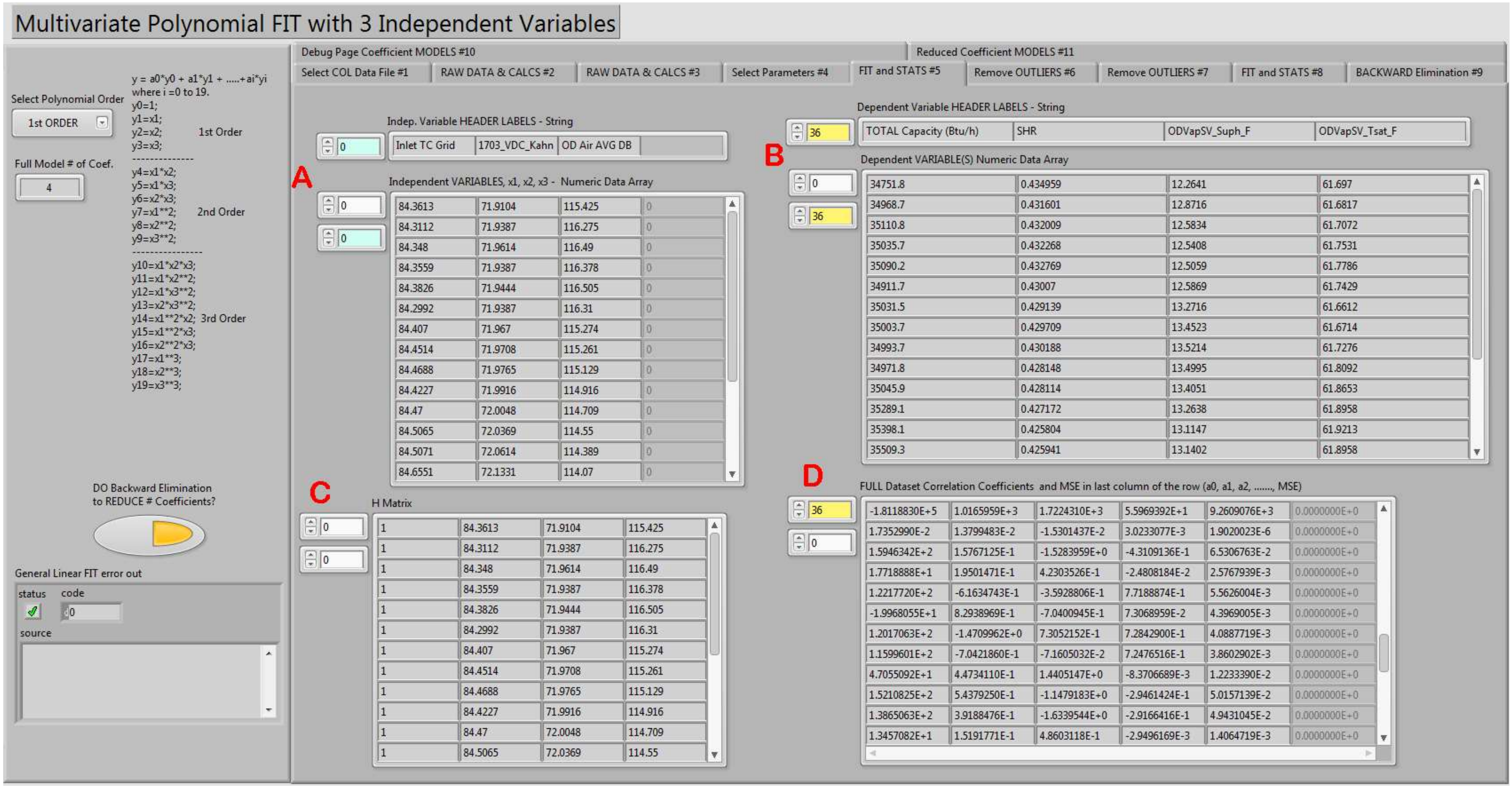


Figure 17: Fit and stats #5 tab

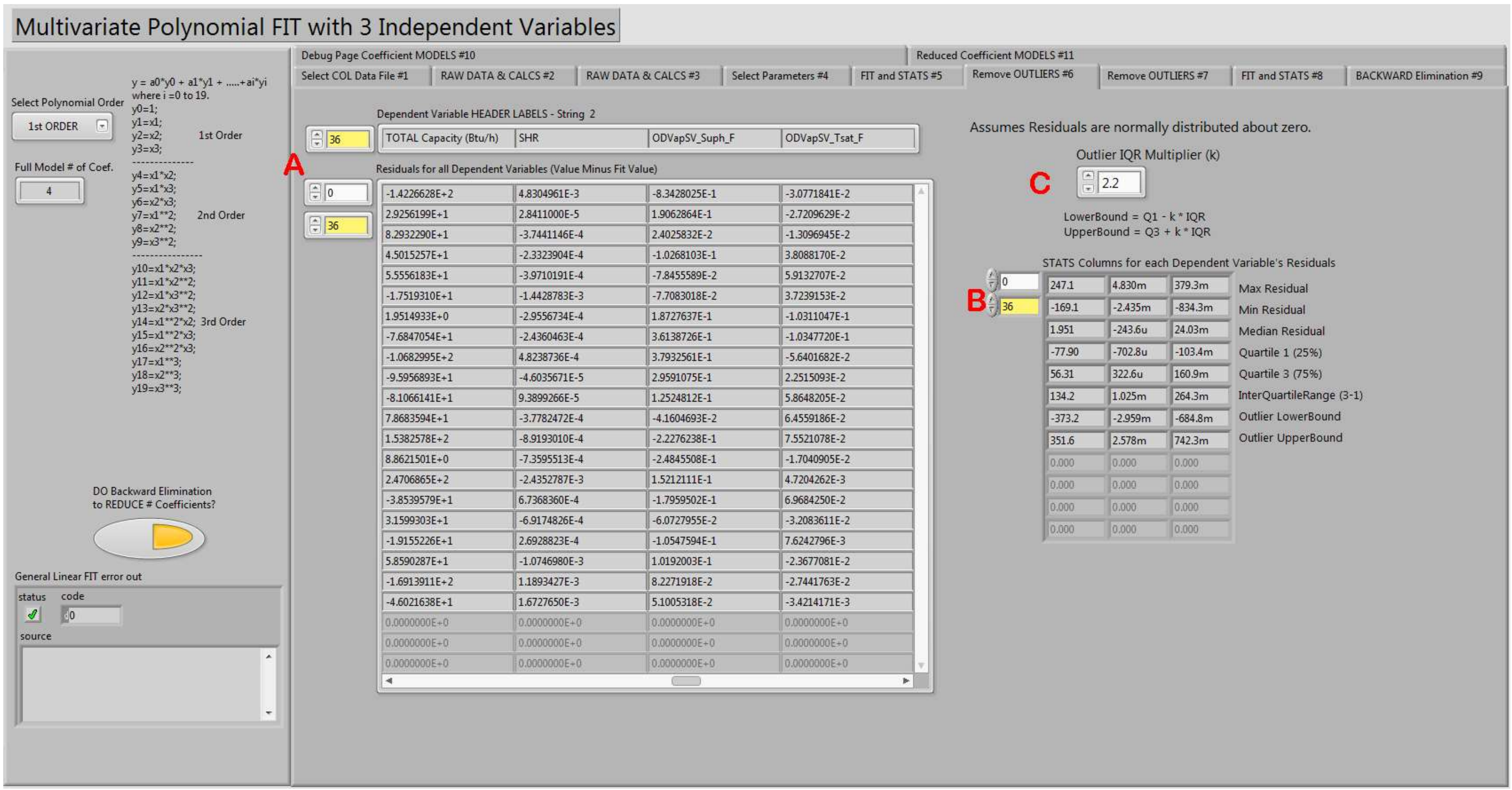


Figure 18: Remove outliers #6 tab

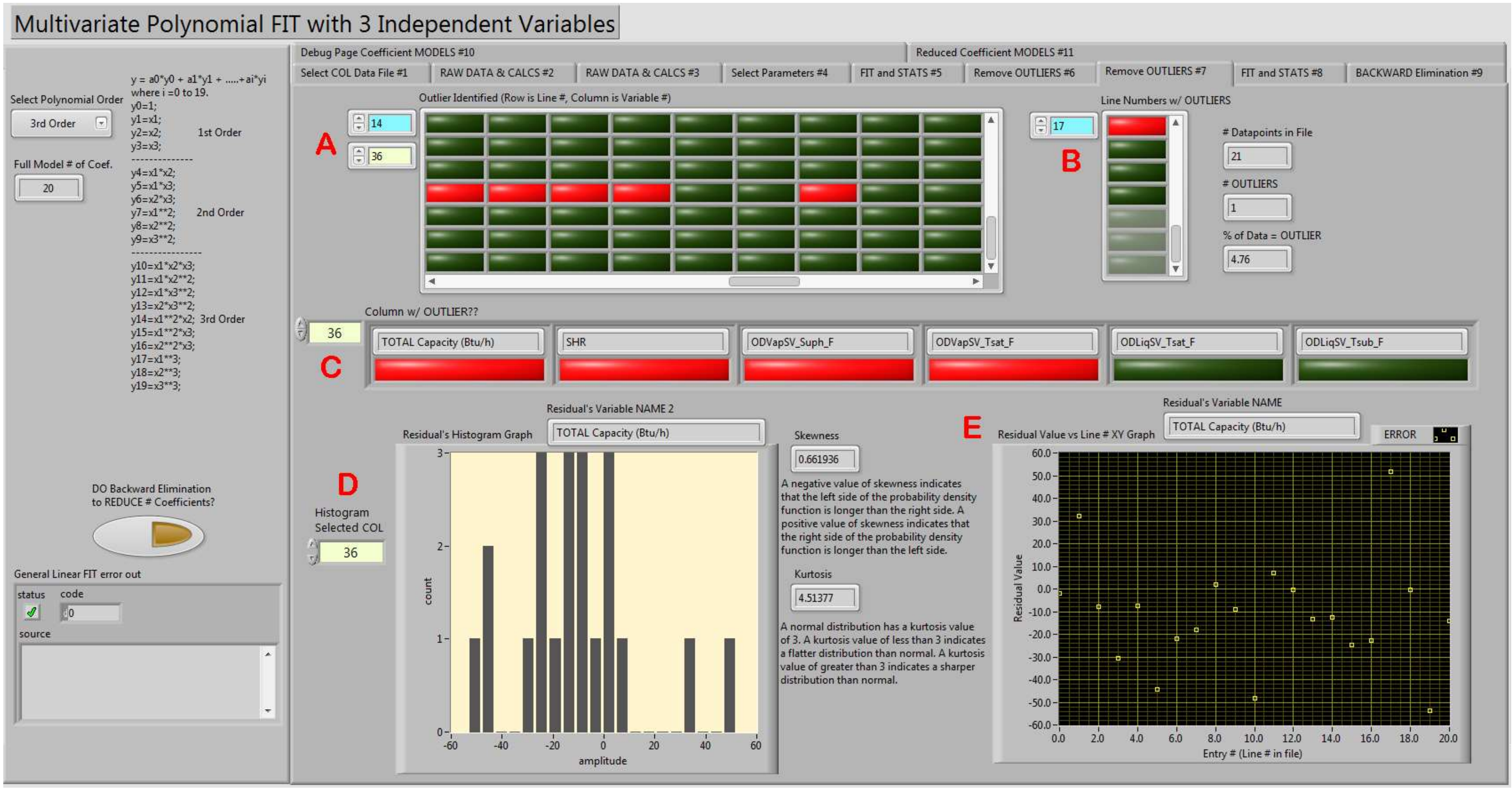


Figure 19: Remove outliers #7 tab

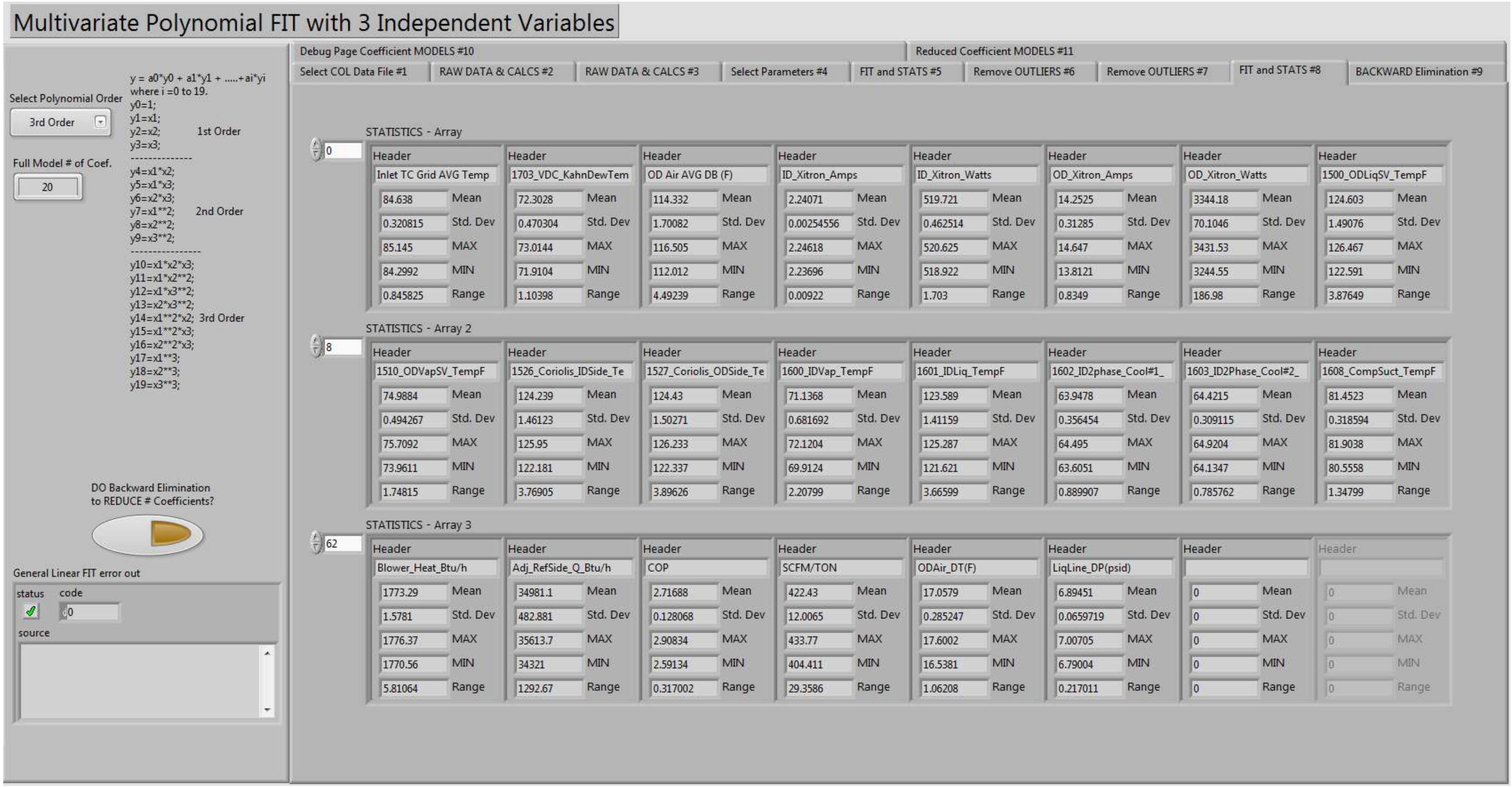


Figure 20: Fit and stats #8 tab

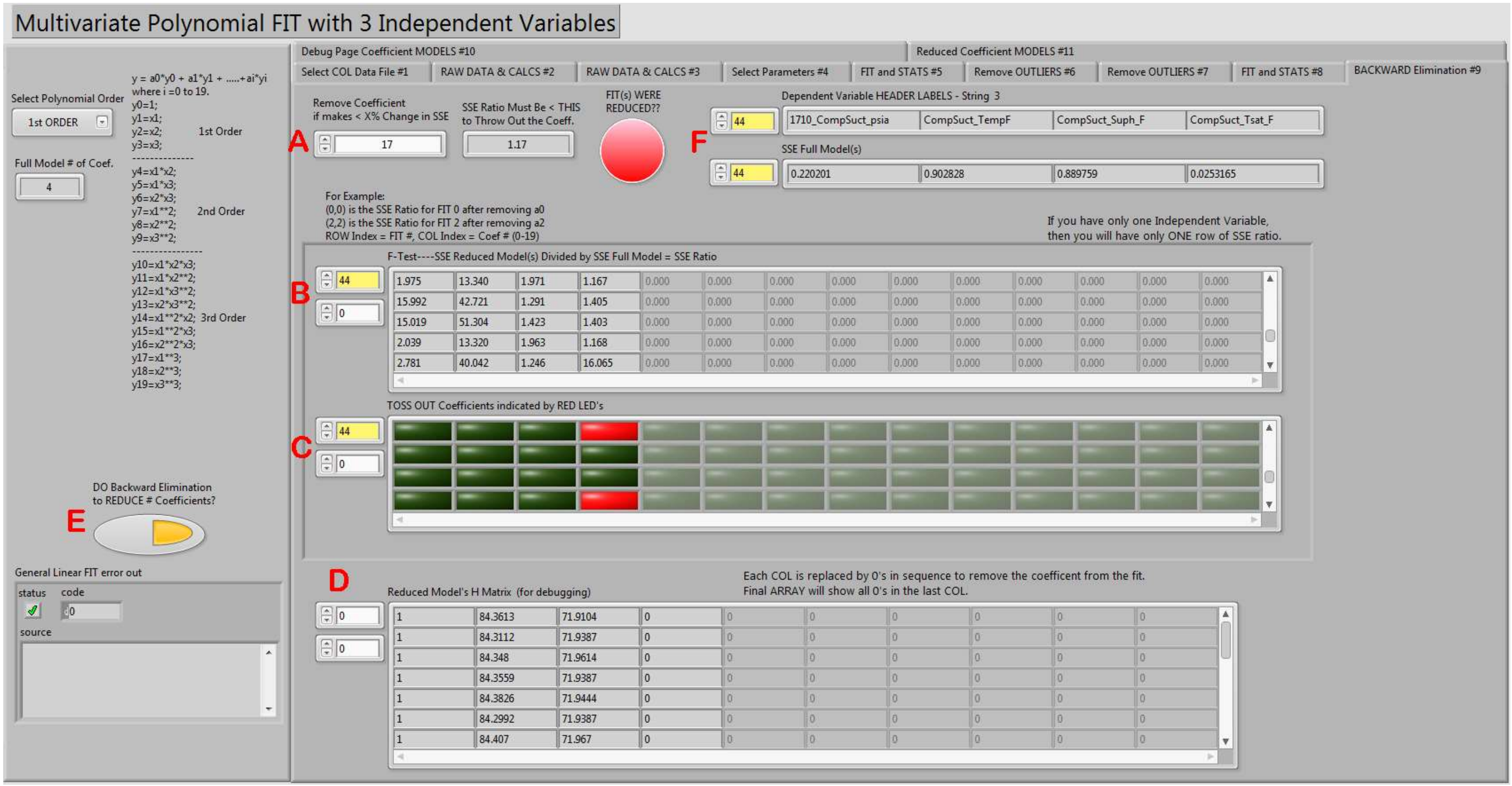


Figure 21: Backward elimination #9 tab

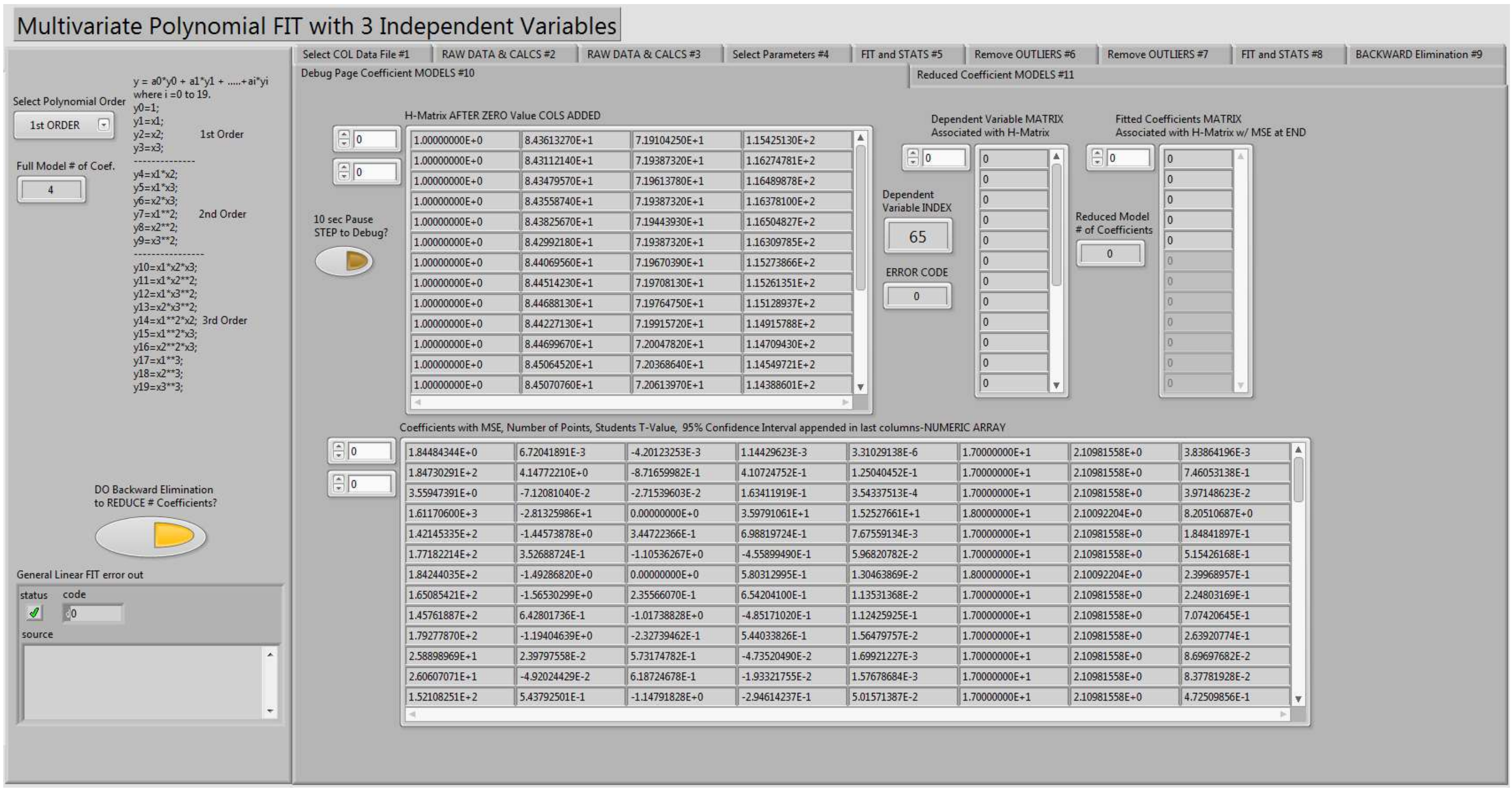
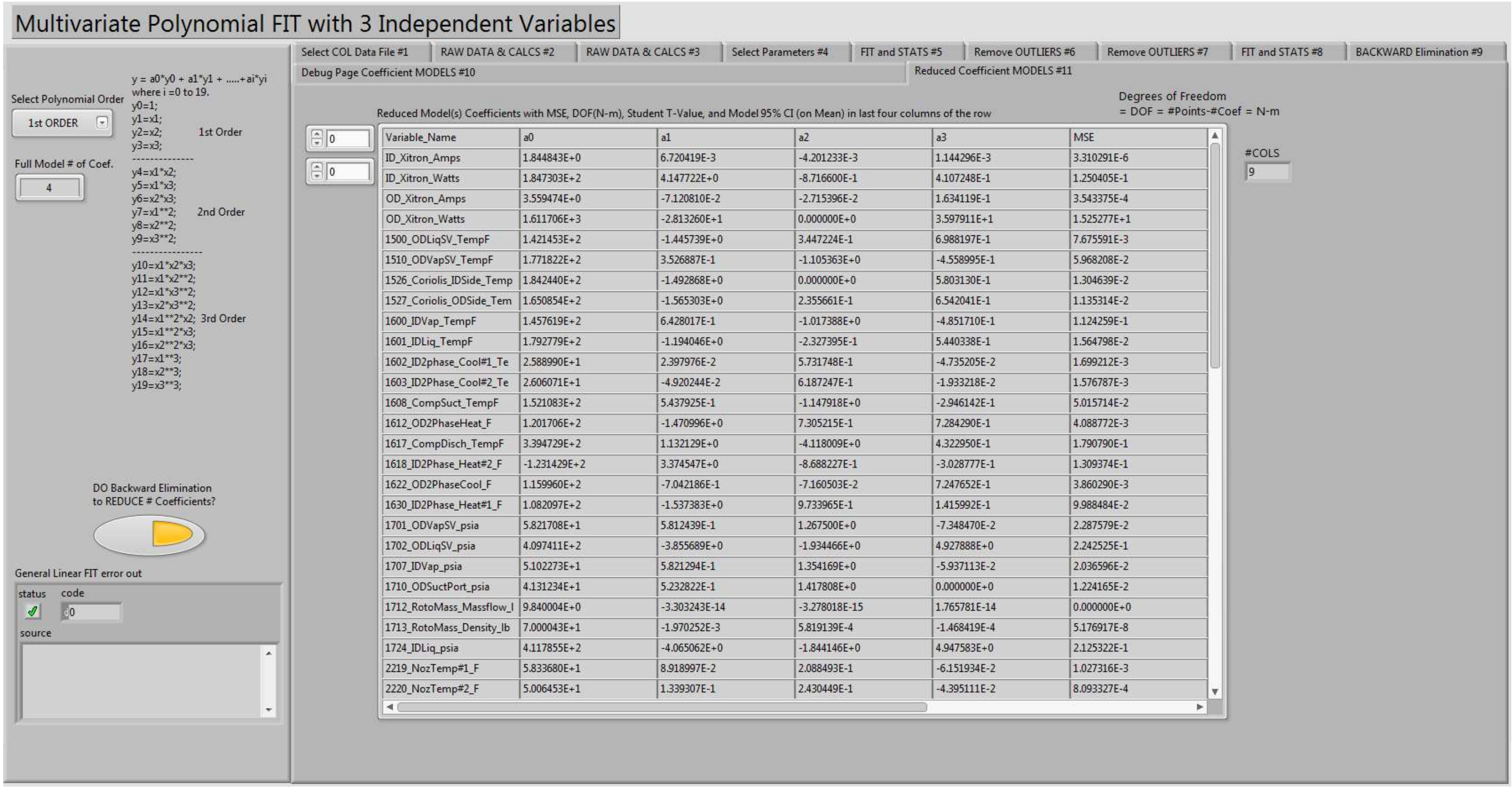


Figure 22: Debug page coefficient models #10 tab



Variable_Name	a0	a1	a2	a3	MSE
ID_Xitron_Amps	1.844843E+0	6.720419E-3	-4.201233E-3	1.144296E-3	3.310291E-6
ID_Xitron_Watts	1.847303E+2	4.147722E+0	-8.716600E-1	4.107248E-1	1.250405E-1
OD_Xitron_Amps	3.559474E+0	-7.120810E-2	-2.715396E-2	1.634119E-1	3.543375E-4
OD_Xitron_Watts	1.611706E+3	-2.813260E+1	0.000000E+0	3.597911E+1	1.525277E+1
1500_ODLiqSV_TempF	1.421453E+2	-1.445739E+0	3.447224E-1	6.988197E-1	7.675591E-3
1510_ODVapSV_TempF	1.771822E+2	3.526887E-1	-1.105363E+0	-4.558995E-1	5.968208E-2
1526_Coriolis_IDSide_Temp	1.842440E+2	-1.492868E+0	0.000000E+0	5.803130E-1	1.304639E-2
1527_Coriolis_ODSide_Tem	1.650854E+2	-1.565303E+0	2.355661E-1	6.542041E-1	1.135314E-2
1600_IDVap_TempF	1.457619E+2	6.428017E-1	-1.017388E+0	-4.851710E-1	1.124259E-1
1601_IDLiq_TempF	1.792779E+2	-1.194046E+0	-2.327395E-1	5.440338E-1	1.564798E-2
1602_ID2phase_Cool#1_Te	2.588990E+1	2.397976E-2	5.731748E-1	-4.735205E-2	1.699212E-3
1603_ID2Phase_Cool#2_Te	2.606071E+1	-4.920244E-2	6.187247E-1	-1.933218E-2	1.576787E-3
1608_CompSuct_TempF	1.521083E+2	5.437925E-1	-1.147918E+0	-2.946142E-1	5.015714E-2
1612_OD2PhaseHeat_F	1.201706E+2	-1.470996E+0	7.305215E-1	7.284290E-1	4.088772E-3
1617_CompDisch_TempF	3.394729E+2	1.132129E+0	-4.118009E+0	4.322950E-1	1.790790E-1
1618_ID2Phase_Heat#2_F	-1.231429E+2	3.374547E+0	-8.688227E-1	-3.028777E-1	1.309374E-1
1622_OD2PhaseCool_F	1.159960E+2	-7.042186E-1	-7.160503E-2	7.247652E-1	3.860290E-3
1630_ID2Phase_Heat#1_F	1.082097E+2	-1.537383E+0	9.733965E-1	1.415992E-1	9.988484E-2
1701_ODVapSV_psia	5.821708E+1	5.812439E-1	1.267500E+0	-7.348470E-2	2.287579E-2
1702_ODLiqSV_psia	4.097411E+2	-3.855689E+0	-1.934466E+0	4.927888E+0	2.242525E-1
1707_IDVap_psia	5.102273E+1	5.821294E-1	1.354169E+0	-5.937113E-2	2.036596E-2
1710_ODSuctPort_psia	4.131234E+1	5.232822E-1	1.417808E+0	0.000000E+0	1.224165E-2
1712_RotoMass_Massflow_l	9.840004E+0	-3.303243E-14	-3.278018E-15	1.765781E-14	0.000000E+0
1713_RotoMass_Density_lb	7.000043E+1	-1.970252E-3	5.819139E-4	-1.468419E-4	5.176917E-8
1724_IDLiq_psia	4.117855E+2	-4.065062E+0	-1.844146E+0	4.947583E+0	2.125322E-1
2219_NozTemp#1_F	5.833680E+1	8.918997E-2	2.088493E-1	-6.151934E-2	1.027316E-3
2220_NozTemp#2_F	5.006453E+1	1.339307E-1	2.430449E-1	-4.395111E-2	8.093327E-4

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 air-conditioners 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.

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

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

		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.

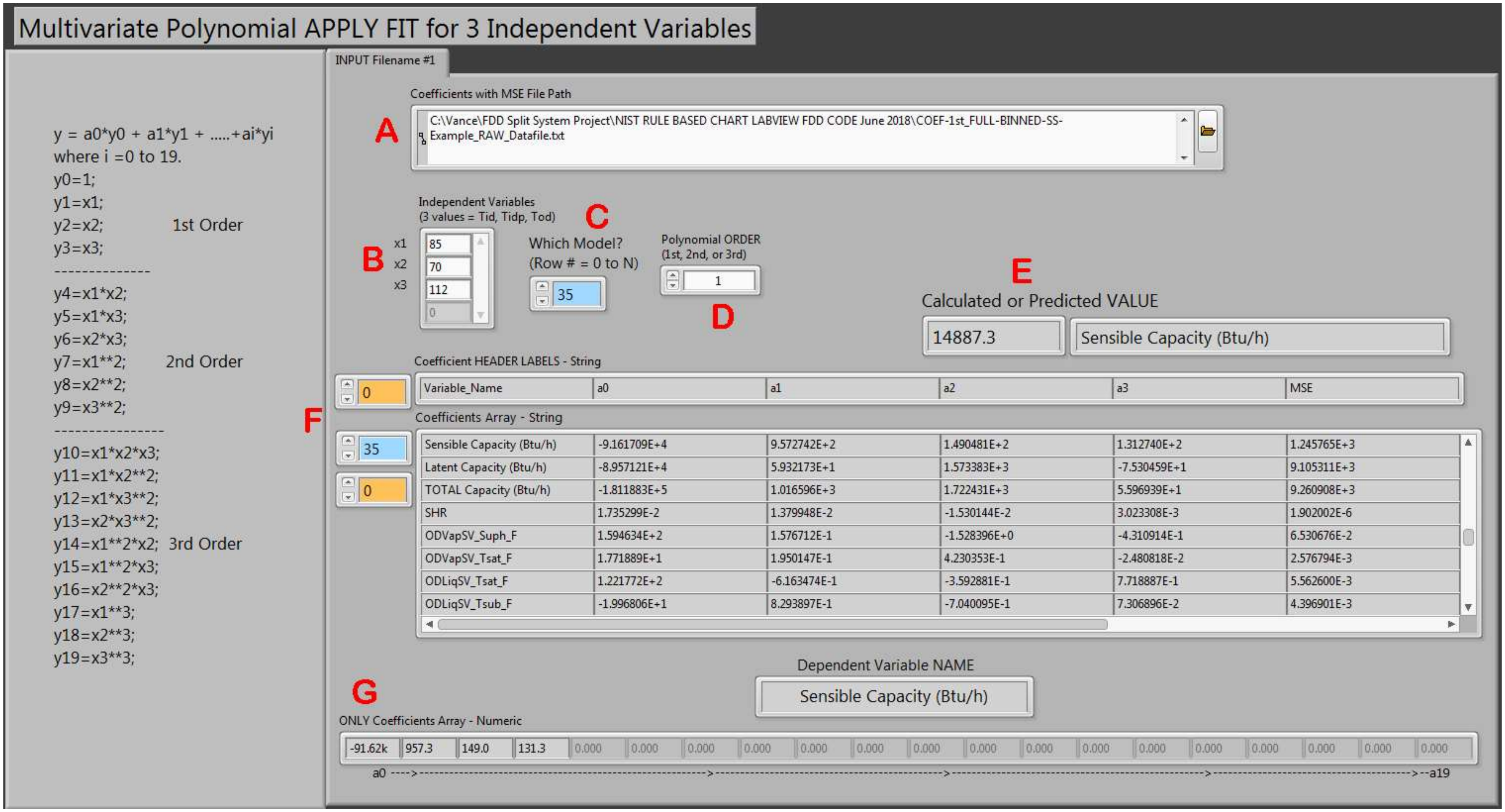


Figure 24: Input filename #1 tab

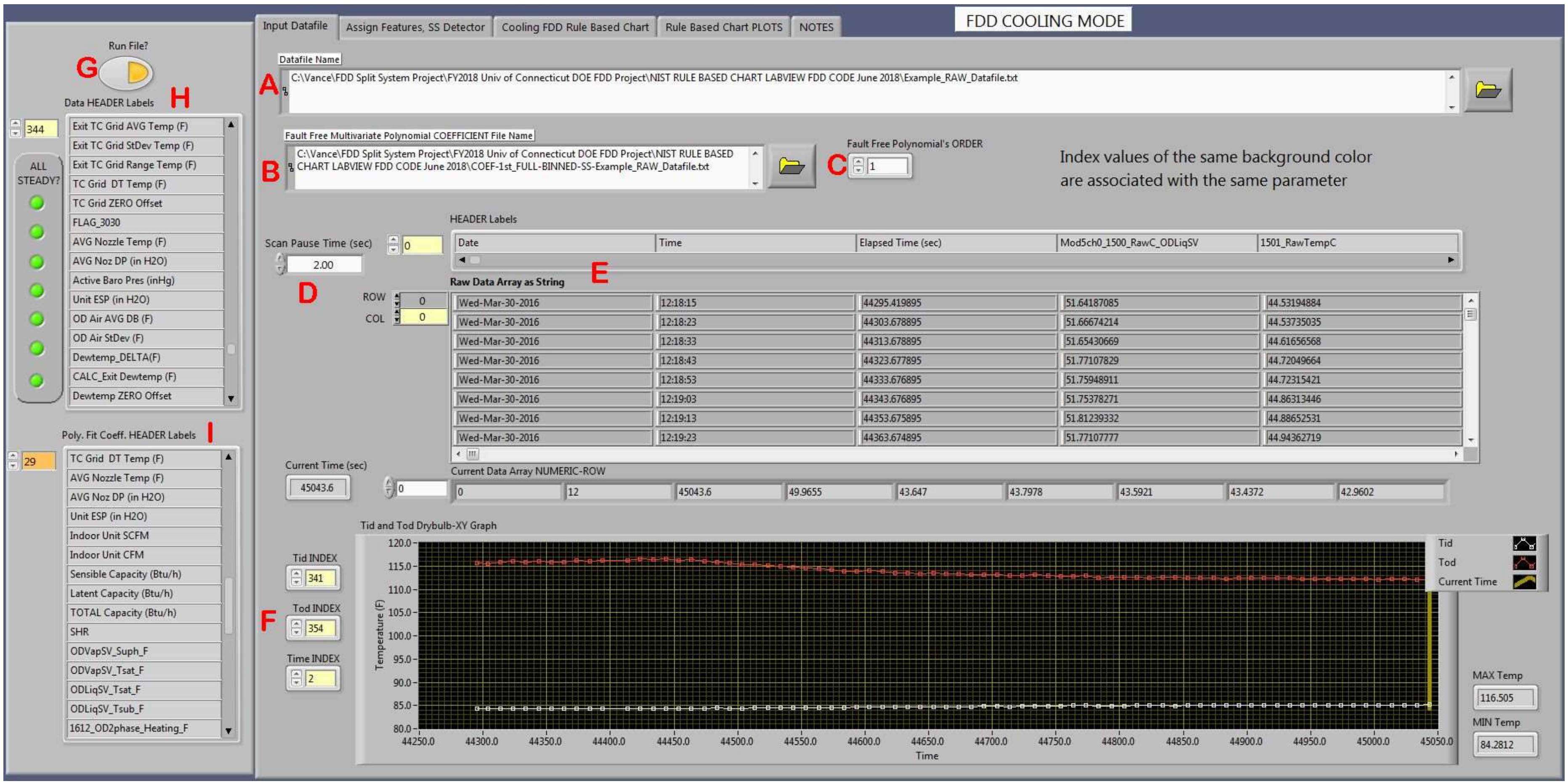


Figure 25: Input datafile tab

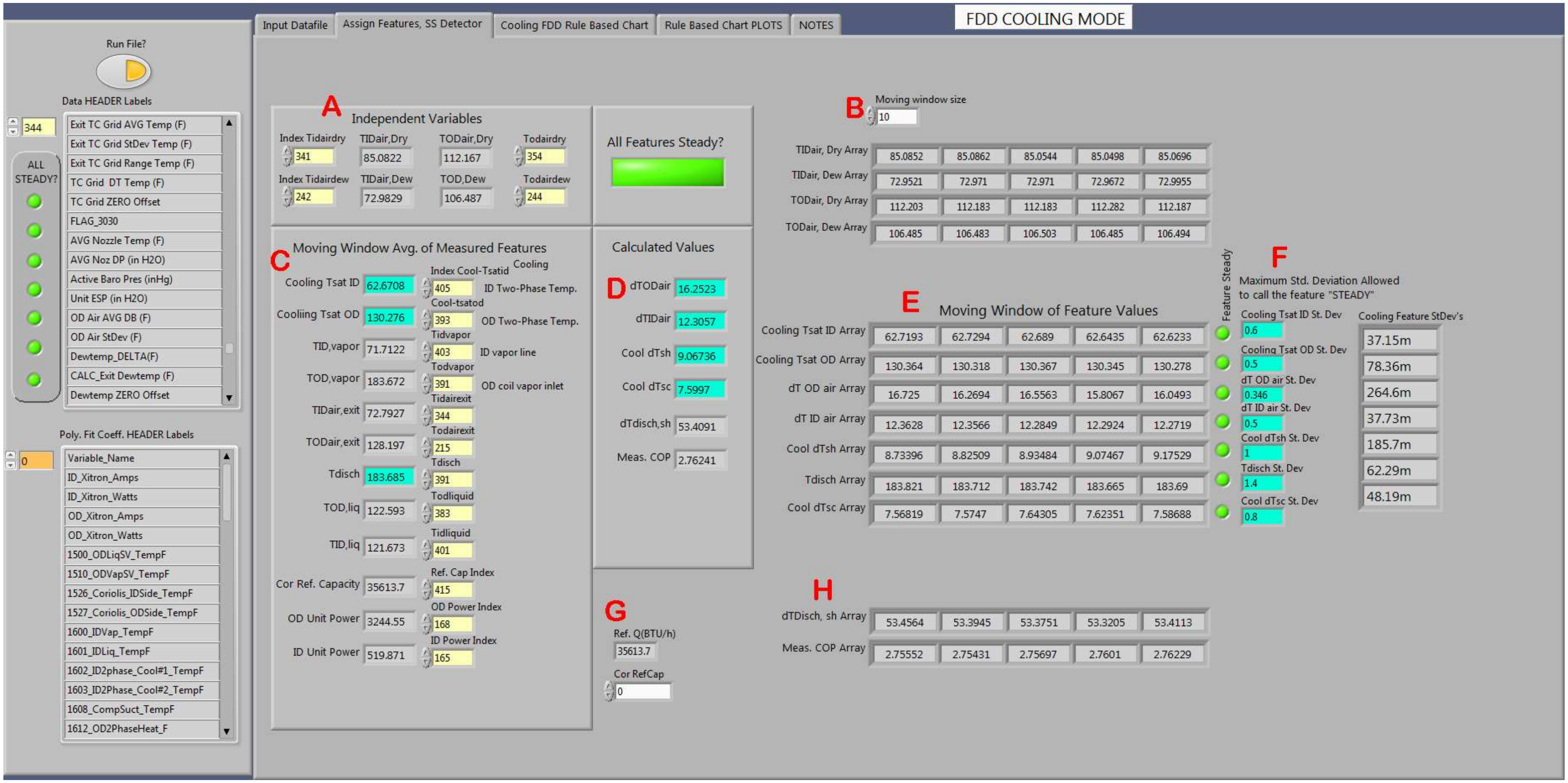


Figure 26: Assign features and steady-state detector tab

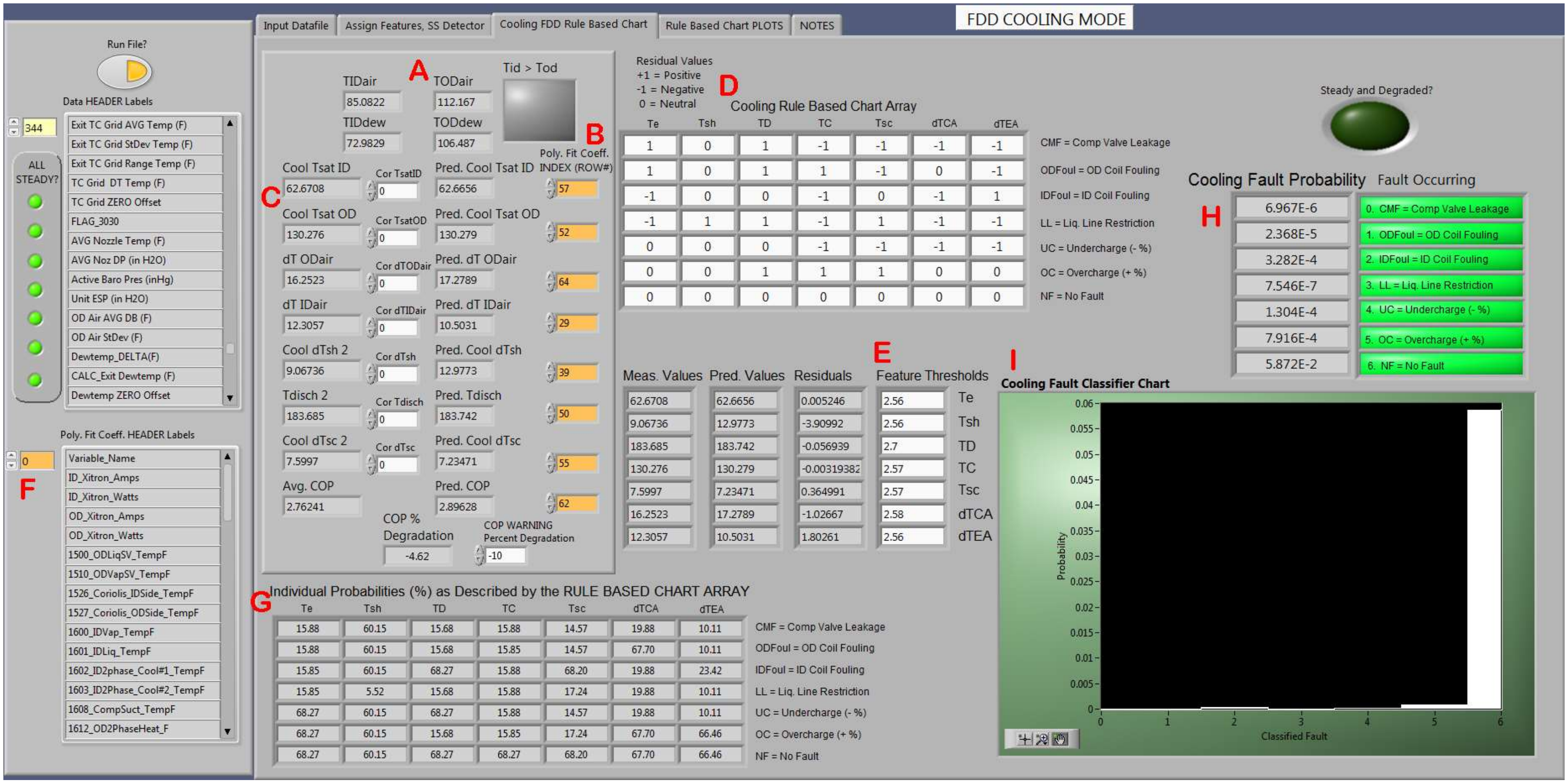


Figure 27: Cooling FDD rule-based chart tab

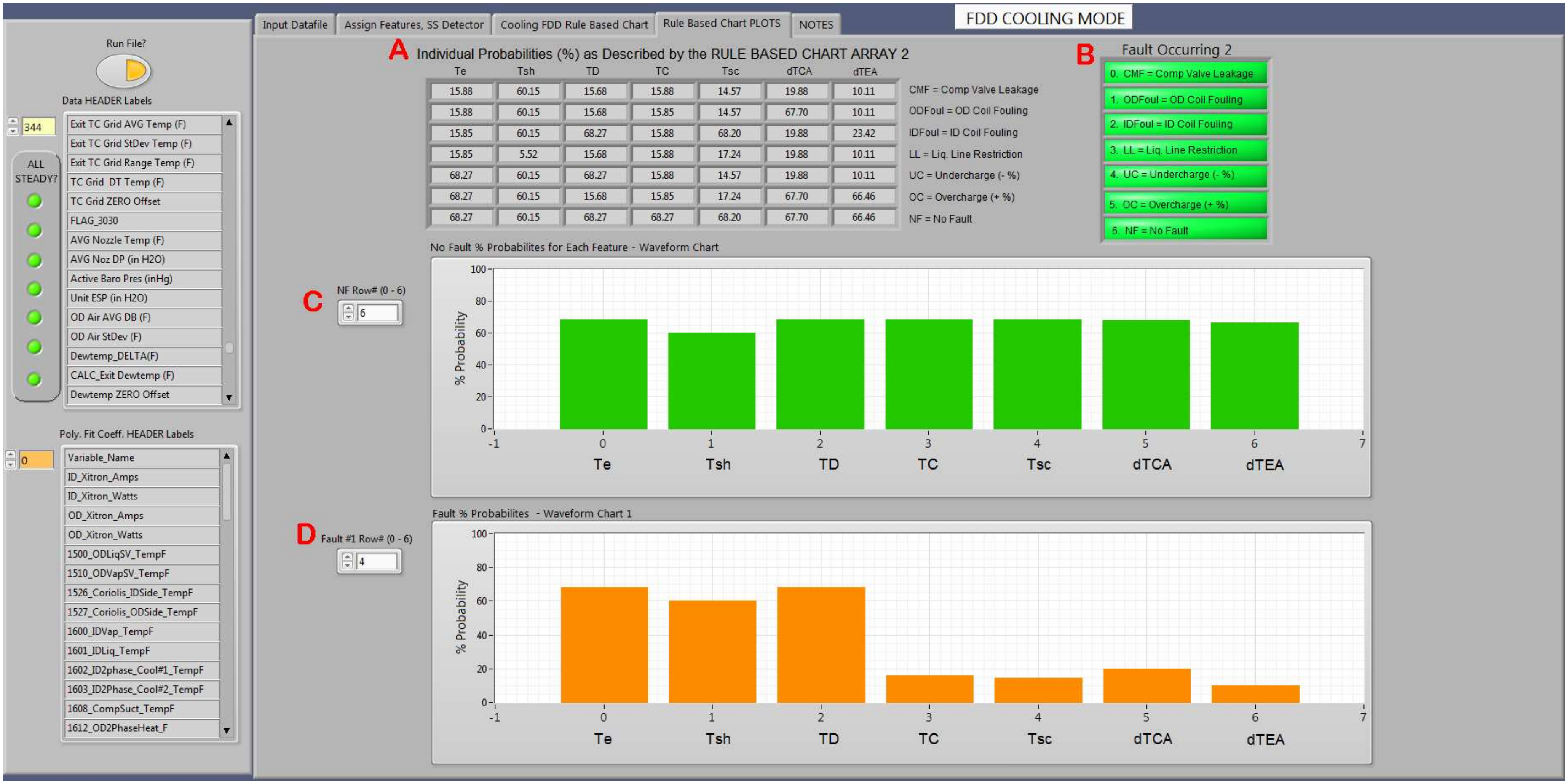


Figure 28: Rule-based chart plots tab

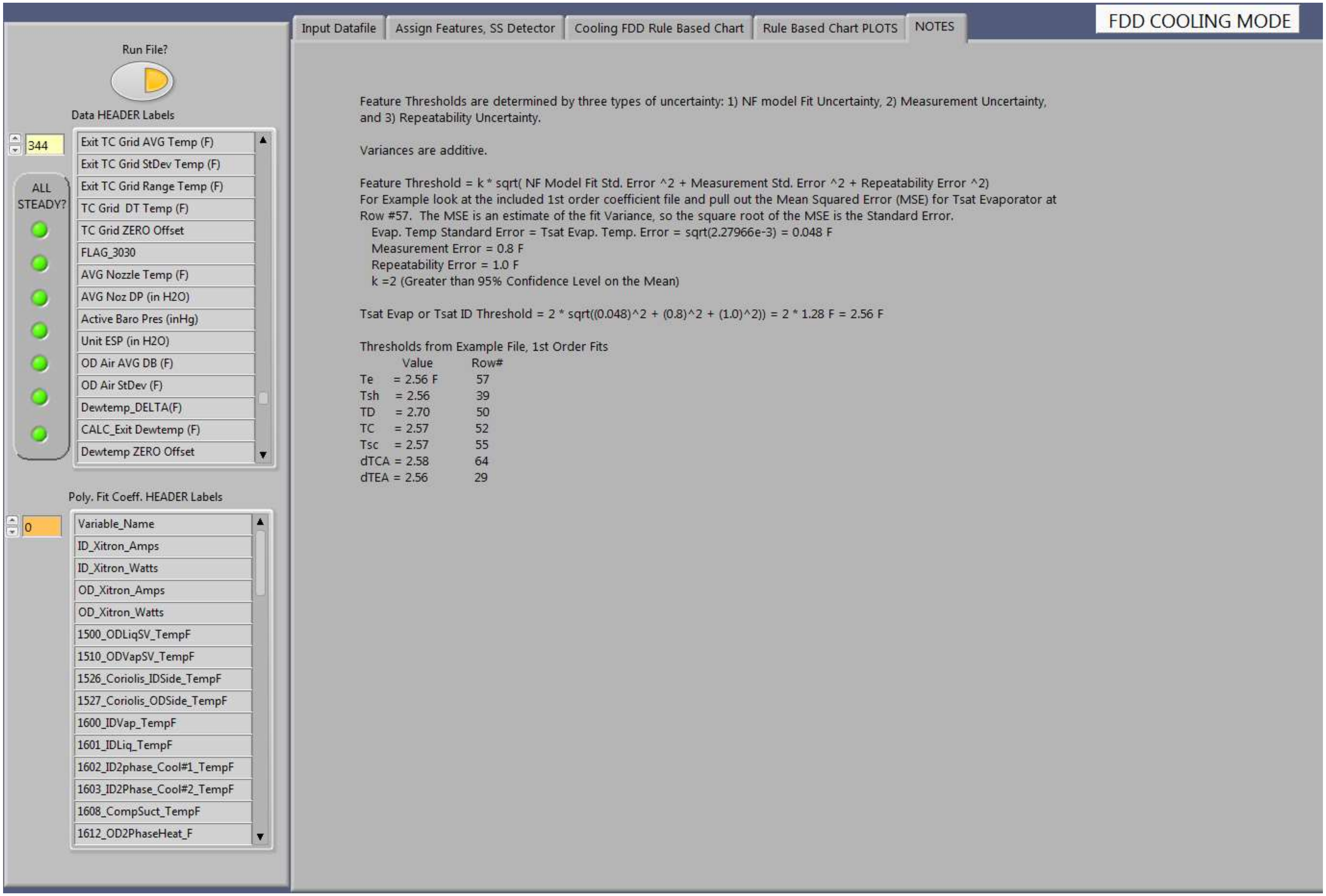


Figure 29: Notes tab

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/small-business-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.